Limiting the Number of Trees in Random Forests

Patrice Latinne[1], Olivier Debeir[2], and Christine Decaestecker[3]

[1] IRIDIA Laboratory, Université Libre de Bruxelles,
50, avenue Franklin Roosevelt cp 196/06
B-1050 Brussels, Belgium
platinne@ulb.ac.be
http://www.ulb.ac.be/polytech/march

[2] Information and Decision Systems, Université Libre de Bruxelles
50, avenue Franklin Roosevelt cp 165/57
B-1050 Brussels, Belgium
odebeir@ulb.ac.be

[3] Laboratory of Histopathology , Université Libre de Bruxelles,
808, route de Lennik cp 620
B-1070 Brussels, Belgium
cdecaes@ulb.ac.be

Presented to the *2nd International Workshop on Multiple Classifier Systems - MCS'2001, Cambridge, UK*

Abstract. The aim of this paper is to propose a simple procedure that *a priori* determines a minimum number of classifiers to combine in order to obtain a prediction accuracy level similar to the one obtained with the combination of larger ensembles. The procedure is based on the McNemar non-parametric test of significance. Knowing a priori the minimum size of the classifier ensemble giving the best prediction accuracy, constitutes a gain for time and memory costs especially for huge data bases and real-time applications. Here we applied this procedure to four multiple classifier systems with C4.5 decision tree (Breiman's Bagging, Ho's Random subspaces, their combination we labeled 'Bagfs', and Breiman's Random forests) and five large benchmark data bases. It is worth noticing that the proposed procedure may easily be extended to other base learning algorithms than a decision tree as well. The experimental results showed that it is possible to limit significantly the number of trees. We also showed that the minimum number of trees required for obtaining the best prediction accuracy may vary from one classifier combination method to another.

1 Introduction

Many methods have been proposed for combining multiple decision trees to improve prediction accuracy [4, 6, 8, 10, 12, 13, 22]). These classifiers are *weakened* to commit errors in a different way so that their combination can correct the mistakes an individual makes [1, 11, 19, 21]. The main experimental studies quoted above applied *systematic* methods to combine hundreds of classifiers and then did not limit *a priori* the number of trees to combine.

As far as we know, optimizing the number of classifiers to combine is an open question in the literature about the improvements of MCSs' design. This number has to be large enough to create diversity among the predictions but it may exist a number beyond which the prediction accuracy remains the same or even decreases with respect to a given criterion. Giacento and Roli [9] proposed to select among a large set of classifiers an optimal subset of both diverse and accurate classifiers of different types (neural and statistical classifiers) .

This approach combines both a systematic design and an 'overproduce-and-choose' strategy which is a problem simpler than generating accurate and diverse classifiers 'directly'. Here we propose a simple procedure based on a direct non-parametric test of comparison, the McNemar test. The procedure systematically determines a minimum number of weakened classifiers to combine for a given data base. It does not require the overproduction of classifiers and does not select better classifiers than others with respect to a given criterion such as proposed by Giacento and Roli's approach. We mean that, once the procedure has been applied, it may be possible to improve the MCS design again with other post-treatments based on the selection of 'good' classifiers for instance.

Nevertheless, to assess the performance of the proposed procedure, we built a large number of weakened decision trees to show that it may not be required to grow random forests to significantly improve prediction accuracy. We applied the procedure to four multiple classifier systems based on C4.5 decision tree: Breiman's Bagging [4], Ho's Random Subspaces [10], their combination in a same model labeled 'Bagfs' [13] and Breiman's Random forests [6]. We assessed the procedure's performances on five large benchmark databases. Indeed, the proposed procedure based on the McNemar test is practically useful for huge data bases or real-time applications for which it has already been successfully applied. It actually allows to reduce memory and time requirements which may be strong criteria for the real-world application of MCSs. The experimental results showed that the use of the McNemar test enables to limit the number of trees for each method significantly. We also observed that the minimal number of trees required for maximum accuracy may vary so that a good trade-off between prediction accuracy and tree requirements of an MCS may be found.

The paper is organized as follows. The random forests are described in Section 2. Then the McNemar test of significance and the procedure for limiting the numbers of classifiers are explained in Section 3. The data bases to which the multiple classifier systems are applied are detailed in Section 4 and the experimental framework in Section 5. We discuss the results in Section 6 before the conclusion (Section 7) and the references.

2 Random Forests

To illustrate our idea of limiting the number of classifiers, we selected four ways of building weakened decision forests: (1) bootstrap aggregating ('Bagging',[4, 16]) (2) Random subspace method (or 'MFS' for Multiple Feature Subsets,[10, 2]) (3) the combination of Bagging with Random subspace ('Bagfs', [14]) and (4) Random Forest ('Bagrf', [6]).

Bagging consists of building B bootstrap replicates of an original data set and of using these to run a learning algorithm. Ross Quinlan [16] has validated the Bagging method with C4.5 decision tree inducer.

The Random subspace method consists of training a given number of classifiers (B), with each having as its input a given proportion of features (k) picked *randomly* from the original set of f features with or without replacement. Ho [10] proposed this approach for decision trees. Bay [2] applied a very similar approach, labeled 'MFS', to nearest neighbors. This method was performed here by using the original feature set only (i.e. without expanding the feature vector with combination functions of features) and by selecting randomly a proportion of features *without* replacement. In the rest of the paper, we will refer to this weakening method by the label 'MFS'.

We showed on benchmark data bases in [13] that combining Bagging and MFS in the same architecture ('Bagfs') could improve prediction accuracy. In [13], the Bagfs' architecture had two levels of decision (A 'nested' level for each bootstrap between all its MFS and a 'final' level between all bootstraps). Here, we applied a simpler architecture with only one level of decision (See also [14]). We generated B bootstrap replicates of the learning set (The same ones used to apply the bagging method). In each replicate we independently sampled a subset of f' features, randomly selected from amongst the f initial ones without replacement (the same ones used to apply 'MFS'). We denoted $k = f'/f$ as the proportion of features in these B subsets. The proposed architecture has thus two parameters, B and k, to be set.

The proportion of features in each subspace, denoted k_{opt} in Table 1, of MFS and Bagfs was optimized by performing a nested stratified 10-fold cross-validation (as more detailed in [13]). It's worth noticing that we obtained the same f_{opt} for both MFS and Bagfs.

Breiman's Random forest method (we labeled 'Bagrf', [6]) consists of creating B bootstrap replicates of the learning set. For each replicate, a feature subset to split on is randomly selected (without replacement) at each node of the tree. According to Breiman's method, we fixed the size of these random subsets, denoted F in Table 1, to be the first integer less than $\log_2(f) + 1$, where f is the number of features.

A common feature of all methods is that they combine predictions by means of the plurality vote. Moreover, Bagging, MFS and Bagfs can be applied to any learning algorithms that are *unstable* for training modifications (e.g. decision trees, artificial neural networks) and feature set modification (e.g. decision trees, nearest neighbours) while Bagrf is specific to decision trees.

We tested each method with respect to Ross Quinlan's C4.5 decision tree Release 8 ([15]) with its default parameter values and its pruning method (all the decision trees were pruned except for Bagrf, as specified in its original formulation).

3 McNemar Test of Significance

3.1 General Background

In this paper, we use the McNemar test [20, 17, 18] as a direct method for testing whether two sets of predictions differ significantly among themselves. Given the two algorithms A and B, this test compares the number of examples misclassified by A, but not by B (labeled M_{ab}), with the number of examples misclassified by B, but not by A (labeled M_{ba}). In the case that $M_{ab} + M_{ba} \geq 20$, if the null hypothesis H_0 is true (i.e.,if there is no difference between the algorithms' predictions), then the statistics X^2 (equation 1) can be considered as following an χ^2 distribution (with 1 degree of freedom).

$$X^2 = \frac{(|M_{ab} - M_{ba}| - 1)^2}{M_{ab} + M_{ba}} \sim \chi^2_{1,0.95} \qquad (1)$$

The hypothesis H_0 is rejected if X^2 is greater than $\chi^2_{1,0.05} = 3.841459$ (significance level $p < 0.05$). In this case, the algorithms have significantly different levels of performance. If condition $M_{ab} + M_{ba} \geq 20$ is not satisfied, the approximation of the statistical distribution cannot be used and the *exact test* described in [17] has to be performed. As this happened rarely in our experimental design, in these cases, we preferred to accept the hypothesis that the two algorithms have the same performance.

Moreover, different studies (see for instance [7, 18]) showed that this non-parametric test is also preferred to parametric ones (such as the commonly used t-test) because no assumption is required and it is independent of any evaluation measurement (error rate, kappa degree of agreement,...). Dietterich [7] also showed that McNemar has a low type I error (the probability of incorrectly detecting a difference when no difference exists) and concluded that it is one of the more acceptable tests among the most common ones if the algorithms can only be executed once.

3.2 Limiting the Number of Classifiers

When creating multiple classifier systems such as the random forests described in Section 2, we may overproduce an arbitrary large number, B, of voting classifiers. In this paper, the question is how to limit the number of classifiers to produce while being as accurate as the same MCS combining a larger number of classifiers.

We applied McNemar test of significance as described in Section 3.1 between two sets of predictions from two MCSs that differ only by their number of classifiers. Let us denote $\mathcal{L}$ a learning set and $\mathcal{T} = \{(\mathbf{x}, y)\}$ a data set independent from $\mathcal{L}$. Let $\mathcal{C}_m = \{\widehat{y} = vote\{\varphi^{(k)}(\mathbf{x}, \mathcal{L}), k = 1, \ldots, m\}\}$ be the prediction set of m voting classifiers. The classifiers $\varphi^{(k)}$ are built so that the classifier predictions are diverse and on an equal footing in terms of voting i.e. no classifier is a priori better than another with respect to any criterion (e.g. as it is the case here by building multiple random decision trees, see Section 2).

The proposed procedure consists of comparing the prediction set $\mathcal{C}_m$ to $\mathcal{C}_n$, with $n > m$, with respect to the McNemar test. Either the set of classifiers used to obtain predictions $\mathcal{C}_n$ is completely independent from the one that predicts $\mathcal{C}_m$, or it contains all or part of the m classifiers that predicts $\mathcal{C}_m$. We showed that this does not change our conclusion as it will be detailed in Section 6.

The McNemar test gives an answer d with a significance level $p < 0.05$:

$$d(m,n) = \begin{cases} 1 & \text{if } H_0 \text{ rejected } \textit{and } M_{mn} > M_{nm} \\ -1 & \text{if } H_0 \text{ rejected } \textit{and } M_{nm} > M_{mn} \\ 0 & \text{if the two prediction sets do not differ} \end{cases}$$

If $d(m,n) = 1$, then we conclude that combining n classifiers gives a higher level of performance than combining m classifiers with respect to McNemar test. So we should carry on the procedure with a higher number of classifiers than n.

If $d(m,n) = -1$, we should stop the procedure and use m classifiers only. It may only appear rarely since increasing the number of voting classifiers should not degrade the prediction accuracy significantly. As a matter of fact, this case never appears in our experiments.

If $d(m,n) = 0$, then combining m weakened classifiers does not significantly differ from combining n classifiers and we may keep $\{\varphi^{(k)}(\mathbf{x}, \mathcal{L}),\ k =, 1 \ldots, m\}$ as the multiple classifier system with the minimal number of classifiers, $m^* = m$, that limits the number of classifiers to combine.

4 Material

We applied Bagging, MFS, Bagfs and Bagrf to 5 large data bases (see Table 1). Four of these were downloaded from the UCI Machine Learning repository [3], i.e. satimage, image segmentation ('image'), letter and DNA. We also included the artificial data base 'ringnorm' used by Breiman in [5]. All these data bases have no missing values. Notice that for DNA, we gave Bagrf's parameter F a higher value since the one obtained by the original computation ($F = 7$, see Section 2) led to a low prediction accuracy.

5 Experimental Design

In the present paper, we investigated the benefit of using the McNemar test of significance as described in Section 3 to determine m^*, the minimum number of classifiers to combine for a given multiple classifier system on a given data set. We illustrated this procedure on four multiple classifier systems described in Section 2, namely Bagging, Random subspaces ('MFS'), Bagfs and another

Table 1. databases used to perform the classification tasks.

Data set	Learning Set Size	#Feat. Cont/nominal	# Classes	f_{opt}	F
ringnorm *	7400	20/0	2	8	5
satimage	6435	36/0	6	18	6
image	2310	18/0	7	7	5
DNA	3186	0/60	3	36	20
letter *	20000	16/0	26	7	5

* : databases where the examples are equi-distributed across the classes.

Random forest, 'Bagrf', applied to five data bases (as detailed in Section 4). For each of these MCSs, we *overproduced* $B = 200$ weakened decision trees. We split each data base in 3 stratified folds, a learning set $\mathcal{L}$, a validation set $\mathcal{V}$ and a testing set $\mathcal{T}$. Evaluations and comparisons of the MCSs were made on the basis of these 3 folds and we validated our approach by permuting the role played by each fold. $\mathcal{L}$ is used as the learning set to build 200 weakened decision trees. $\mathcal{V}$ is used to apply the McNemar test between the prediction set resulting from the vote of m classifiers ($m = 1 \ldots 200$) and the prediction set of the vote of n classifiers ($n > m$). Finally, we kept $\mathcal{T}$ for testing independently the procedure predictions.

Using the validation set $\mathcal{V}$, we obtained the table $D_v = d_v(m, n)$, $m, n = 1, \ldots, 200$ (subscript v is used for results carried out on the basis of the validation set $\mathcal{V}$). Once D_v is so computed, we extracted the recommended m_v^* as explained in Section 3.2. Then the remaining data set, $\mathcal{T}$, is used to determine $D = d(m, n)$ and extract m^* in order to assess the proposed procedure on the independent testing set. For each classification task and each MCS, we are then able to compare m^* to m_v^* and thus to appreciate the quality of the predicted value m_v^*.

6 Results and Discussion

On Figure 1, for each method and each data base, each dot represents $d(m, n)$, the result of McNemar test that compares the prediction set of a m-classifier system (on a row) with the prediction of a n-classifier system (on a column) ($m, n = 1..200$). Each figure is symmetrical and composed of a bright and dark region. The dark region means that the compared architectures differ significantly with respect to McNemar. The bright region means that the compared architectures do not differ significantly. These results showed that a threshold appeared distinctly between the two regions 'differ' or 'differ not significantly'. So the proposed procedure based on McNemar test led to the determination of m_v^*, a significantly lower number of classifiers on most data bases than the total of 200 classifiers overproduced for each MCS.

Table 2 shows the results' summary of the experimental design for each multiple classifier system and each data base. This table indicates in bold and in

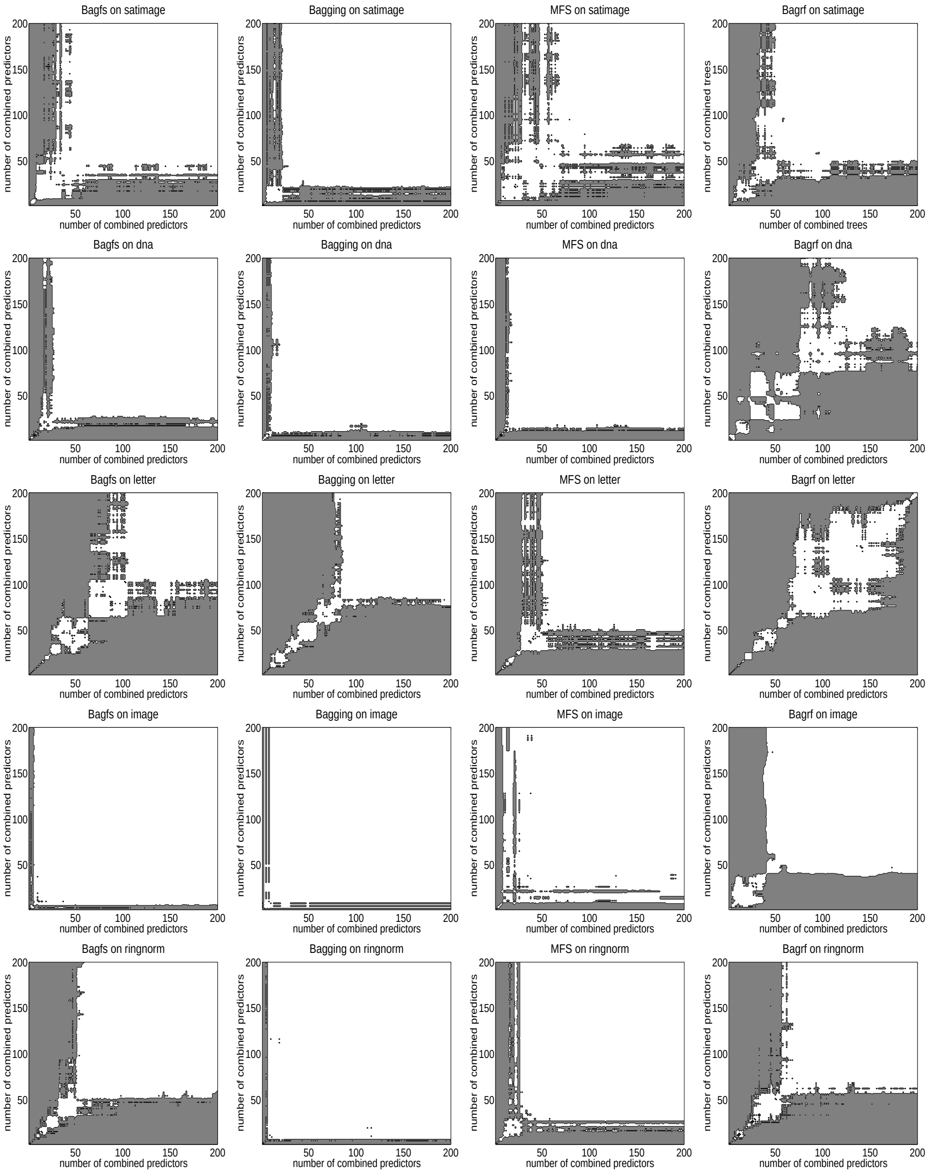

Fig. 1. Experimental results. Influence of the number of trees on the predictions with respect to McNemar test

brackets each m_v^* computed as detailed above. We also give the percentage of good classification obtained with each m_v^* to compare the performance of each MCS on the same data base.

The results obtained on the remaining independent testing set from the global data base, m^*, showed that this number was always close to the predicted value m_v^* (most of the time equal or even lower). In the too optimistic but rare cases where $m_v^* < m^*$, we observed that the difference was never larger than 10. Nevertheless, the results showed that by performing this procedure, we obtained a drastic decrease of the number of classifiers required to obtain the same level of performance than MCSs combining 200 classifiers. We also observed in Table 2 that

- Bagfs systematically exhibited a better prediction accuracy and a lower m_v^* than Bagrf with respect to McNemar test.
- By increasing the number of classifiers, Bagfs always exhibited significantly better performance than Bagging with respect to McNemar test.
- To obtain the same level of accuracy than MFS, Bagfs required less classifiers on satimage and image. On the other data bases, Bagfs exhibited significantly better results than MFS (with respect to McNemar) but it required more classifiers.

Table 2. Experimental Results. Performance in terms of the prediction accuracy (%), minimum recommended number of trees with respect to McNemar in bold and in brackets.

	C4.5	**Bag**	**MFS**	**Bagfs**	**Bagrf**
ringnorm	89.3	94.0	97.7	98.4	96.2
		(10)	**(30)**	**(50)**	**(60)**
satimage	84.0	88.2	89.8	89.6	89.1
		(20)	**(70)**	**(50)**	**(50)**
image	93.6	96.1	96.2	96.0	94.5
		(10)	**(50)**	**(10)**	**(40)**
DNA	86.5	89.5	90.0	91.5	89.8
		(20)	**(20)**	**(30)**	**(130)**
letter	81.4	88.6	90.4	91.6	89.0
		(90)	**(50)**	**(110)**	**(200)**

In the present paper, we systematically *overproduced* classifiers (200) to assess the method's performance. The results obtained on each data base with each MCS let us suggest that we could incrementally increase the number of classifiers by step of 10 classifiers and perform the direct test of McNemar at each step *instead*. Furthermore, this approach of the MCS' design would combine a limited number of classifiers, m_v^*, (i.e. predicted on a reduced validation set independent from the learning set) without any *significant* loss of accuracy and applied to a large data set of unknown cases. This question is especially interesting for huge data bases and real-time applications working on other base learning algorithms slower than decision trees (e.g. neural networks) to obtain a gain in time and memory costs.

7 Conclusion

We suggested a simple procedure based on the direct test of McNemar to limit the number of classifiers to combine in a multiple classifier system. The procedure compares the set of predictions of a MCS with a given number of classifers with the prediction set of the same MCS with a higher number of classifiers. If the prediction sets do not differ with respect to McNemar test, we concluded that the smallest number of classifiers is enough to obtain the same level of accuracy with respect to McNemar test.

Experimental results showed on four different MCSs applied to C4.5 decision tree and cross-validations on five large benchmark data bases, that it may be possible to select *a priori* a minimum number of classifiers which, once combined with the plurality voting rule, offered the same level of performance than larger numbers of trees with respect to the McNemar test. Moreover, we showed that a sharp threshold appeared between the region where the prediction sets 'differ' and the one where the prediction sets 'do not differ significantly' with respect to McNemar test.

Furthermore, we suggested a way to improve the design of a MCS without overproducing classifiers. It consists of incrementally adding new classifiers to the existing ensemble and comparing by means of a cross-validation the predictions of the resulting ensemble to the one with less classifiers.

Finally, we proposed a simple approach in this paper to improve the design of a multiple classifier system that consisted of limiting the number of classifiers to combine without a loss of prediction accuracy (with respect to a direct statistical test of comparison, McNemar) but with a gain in memory and time costs that may be significant for huge data bases and real-time applications.

Acknowledgements

Patrice Latinne and Olivier Debeir are supported by a grant under an ARC (Action de Recherche Concertée) programme of the Communauté Française de Belgique. Christine Decaestecker is a Research Associate with the 'F.N.R.S' (Belgian National Scientific Research Fund).

The authors also thank the anonymous reviewers for their pertinent and constructive remarks.

References

1. Ali and Pazzani. Error reduction through learning multiple descriptions. *Machine Learning*, 24:173–202, 1996.

2. Stephen D. Bay. Nearest neighbor classification from multiple feature subsets. In *Proceedings of the International Conference on Machine Learning*, Madison, Wisc., 1998. Morgan Kaufmann Publishers.
3. C. Blake, E. Keogh, and C.J. Merz. Uci repository of machine learning databases. [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.
4. Leo Breiman. Bagging predictors. *Machine Learning*, 24, 1996.
5. Leo Breiman. Arcing classifiers. *Annals of statistics*, 26:801–849, 1998.
6. Leo Breiman. Random forests - random features. Technical Report 567, Statistics Department, University of California, Berkeley, CA 94720, september 1999.
7. T.G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.
8. T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees : bagging, boosting and randomization. *Machine Learning*, 40:139–157, 2000.
9. Giorgio Giacento and Fabio Roli. An approach to the automatic design of multiple classifier systems. *Pattern recognition letters*, 22:25–33, 2001.
10. T.K. Ho. the random subspace method for constructing decision forests. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20:832–844, 1998.
11. Ji and Ma. Combinations of weak classifiers. *IEEE Trans. Neural Network*, 7(1):32–42, 1997.
12. Ron Kohavi and Clayton Kunz. Option decision trees with majority votes. In *Proceedings of the Fourtheeth International Conference on Machine Learning*, pages 161–169, San Francisco, CA, 1997. Morgan Kaufmann.
13. Patrice Latinne, Olivier Debeir, and Christine Decaestecker. Different ways of weakening decision trees and their impact on classification accuracy. In *Proc. of the 1st International Workshop of Multiple Classifier System*, pages 200–210, Cagliari, Italy, 2000. Springer (Lecture Notes in Computer Sciences; Vol. 1857).
14. Patrice Latinne, Olivier Debeir, and Christine Decaestecker. Mixing bagging and multiple feature subsets to improve classification accuracy of decision tree combination. In *Proc. of the Tenth Belgian-Dutch Conference on Machine Learning Benelearn'00*, pages 15–22, Tilburg University, 2000. Ed. Ad Feelders.
15. J.R. Quinlan. *C4.5 : Programs For Machine Learning.* Morgan Kaufmann Publishers, San Mateo, California, 1993.
16. J.R. Quinlan. Bagging, boosting, and c4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 725–730, 1996.
17. Bernard Rosner. *Fundamentals of Biostatistics.* Duxbury Press (ITP), Belmont, CA, USA, 4th edition, 1995.
18. Steven Salzberg. On comparing classifiers : Pitfalls to avoid and a recommended approach. *Data Mining and knowledge discovery*, 1:317–327, 1997.
19. R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.
20. S. Siegel and N.J. Castellan. *Nonparametric Statistics for the behavioral sciences.* McGraw-Hill, second edition, 1988.
21. K. Tumer and J. Ghosh. Classifier combining : analytical results and implications. In *Proceedings of the National Conference on Artificial Intelligence*, Portland, OR, 1996.
22. Zijian Zheng. Generating classifier committees by stochastically selecting both attributes and training examples. In *Proceedings of the 5th Pacific Rim International Conferences on Artificial Intelligence (PRICAI'98)*, pages 12–23. Berlin: Springer-Verlag, 1998.