Repairing Concavities in ROC Curves

Peter A. Flach
Department of Computer Science
University of Bristol
Bristol, United Kingdom
Peter.Flach@bristol.ac.uk

Shaomin Wu
School of Construction Management
and Engineering, University of Reading
Reading, United Kingdom
Shaomin.Wu@reading.ac.uk

Abstract

In this paper we investigate methods to detect and repair concavities in ROC curves by manipulating model predictions. The basic idea is that, if a point or a set of points lies below the line spanned by two other points in ROC space, we can use this information to repair the concavity. This effectively builds a hybrid model combining the two better models with an inversion of the poorer models; in the case of ranking classifiers, it means that certain intervals of the scores are identified as unreliable and candidates for inversion. We report very encouraging results on 23 UCI data sets, particularly for naive Bayes where the use of two validation folds yielded significant improvements on more than half of them, with only one loss.

1 Introduction

There is an increasing amount of work on model selection and model combination in the machine learning and data mining literature: for instance, model selection based on ROC space [Provost and Fawcett, 2001], model combination by means of bagging [Breiman, 1996], boosting [Freund and Schapire, 1996], arcing [Breiman, 1998], the mixture of experts method [Jacobs *et al.*, 1991], to name just a few. A review on ensembles of learning machines can be found in [Valentini and Masulli, 2002].

Typically, these methods assume a set of given models with fixed performance, and the issue is how best to combine these models to obtain a better ensemble model. There is no attempt to analyse the performance of the given models to determine a region where performance is sub-standard. This paper investigates methods to improve given models using ROC analysis.

ROC (Receiver Operating Characteristic) analysis is usually associated with classifier selection when both class and misclassification cost distribution are unknown at training time. However, ROC analysis has a much broader scope that is not limited to cost-sensitive classification. A categorical classifier is mapped to a point in ROC space by means of its false positive rate on the X-axis and its true positive rate on the Y-axis. A probabilistic classifier results in a ROC curve, which aggregates its behaviour for all possible decision thresholds. The quality of a probabilistic classifier can be measured by the Area Under the ROC Curve (AUC), which measures how well the classifier separates the two classes without reference to a decision threshold. A good classifier should have a large AUC, and AUC=1 means that there is a decision threshold such that the corresponding categorical classifier has 100% accuracy.

We use the term *model repair* to denote approaches that modify given models in order to obtain better models. In contrast, ensemble methods produce hybrid models that leave the original models intact. An approach to model construction using ROC space is given in [Blockeel and Struyf, 2002], where the authors identify and assemble parts of a decision tree that perform well in different areas of ROC space. Our approach in this paper is to identify 'bad' areas, or concavities, in a ROC curve and repair them by manipulating the corresponding low-quality predictions. The approach is experimentally validated using both naive Bayes and decision tree, but the approach has much wider scope as it can be applied to any classifier that computes class scores.

To illustrate the approach, we describe in Section 2 the RepairPoint algorithm that combines three models based on different thresholds of the same probabilistic classifier, and creates a new model which theoretically should improve upon the worst of the three models. In Section 3 we introduce the main algorithm RepairSection, that mirrors an entire concave region (a region of the curve that is below its convex hull). In Section 4 we present experimental results on 23 data sets from the UCI repository. Section 5 reviews some related work on model ensembles, gives the main conclusion and suggests further work.

2 Basics of repairing classifiers in ROC space

Assume that the confusion matrix of a classifier evaluated on a test set is as in Table 1. Then the true positive rate of the classifier is $a/(a+b)$ and the false positive rate of the classifier is $c/(c+d)$. The point $(c/(c+d), a/(a+b))$ in the XY plane (i.e., ROC space) will be used to represent the performance of this classifier.

If a model is under the ascending diagonal in ROC space, this means that it performs worse than random. Models A and B in Figure 1 are such worse-than-random models.

	predicted positive	predicted negative
actual positive	*a*	*b*
actual negative	*c*	*d*

Table 1: A confusion matrix.

However, there is a very useful trick to obtain better-than-random models: simply invert all predictions of the original model. This corresponds to exchanging the columns in the contingency table, leading to a new true positive rate of $b/(a+b) = 1-a/(a+b)$, i.e. one minus the original true positive rate; similarly we obtain a new false positive rate of $d/(c+d) = 1-c/(c+d)$. Geometrically, this corresponds to mirroring the original ROC point through the midpoint on the ascending diagonal.

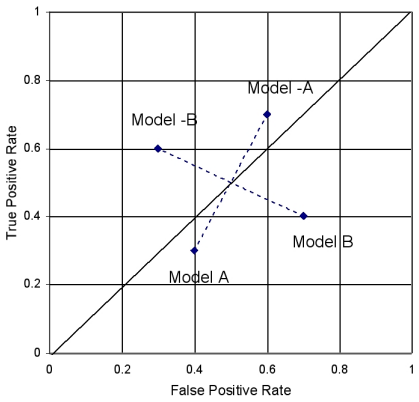

Figure 1: By inverting their predictions, worse-than-random models A and B below the diagonal can be transformed into better-than-random models -A and -B above the diagonal.

Notice that the ascending diagonal really connects two classifiers: the classifier which always predicts negative in (0,0), and the classifier which always predicts positive in (1,1). This suggests that the above repair procedure can be generalised to line segments connecting arbitrary classifiers. For instance, consider Figure 2. Denote the sets of true and false positives of Model *i* by TP_i and FP_i, then we can construct Model 4 under the condition that $TP_1 \subseteq TP_3 \subseteq TP_2$ and $FP_1 \subseteq FP_3 \subseteq FP_2$. In particular, these *inclusion constraints* are satisfied if Models 1, 2 and 3 are obtained by setting thresholds on the same probabilistic model, which is what we assume throughout the paper.

Model 4 operates as indicated in Table 2. The inclusion constraints guarantee that the geometric configuration of Figure 2 holds. This is formally stated in the following theorem.

Theorem 1 *Assuming that $TP_1 \subseteq TP_3 \subseteq TP_2$ and $FP_1 \subseteq FP_3 \subseteq FP_2$, the model produced by Algorithm* RepairPoint *has true and false positive rates $TPr_4 = TPr_1 + TPr_2 - TPr_3$ and $FPr_4 = FPr_1 + FPr_2 - FPr_3$, where TPr_i and FPr_i denote true and false positive rates of Model i.*

Proof. Under the inclusion constraints expressed in the theorem, there are four disjoint groups of examples: those classified positive by Models 1, 3 and 2 ($TP_1 \cup FP_1$); those classified negative by Model 1 and positive by Models 3 and 2 ($(TP_3 - TP_1) \cup (FP_3 - FP_1)$); those classified negative by Models 1 and 3 and positive by Model 2 ($(TP_2 - TP_3) \cup (FP_2 - FP_3)$); and those classified negative by all three models ($(POS - TP_2) \cup (NEG - FP_2)$, where *POS* and *NEG* are the sets of all positive and all negative examples, respectively). By construction, Model 4 classifies the first group as positive, the second group as negative, the third group as positive, and the fourth group as negative. The true positives of Model 4 are thus $TP_1 \cup (TP_2 - TP_3)$; because of the inclusion constraints the result follows (analogous for false positives).

Given three models Model 1, Model 2 and Model 3, output Model 4 that operates as follows:

1. If both Model 1 and Model 2 predict negative, then predict negative;
2. If both Model 1 and Model 2 predict positive, then predict positive;
3. If Model 1 predicts negative and Model 2 predicts positive, then predict the opposite of what Model 3 predicts;
4. Otherwise, predict what Model 3 predicts.

Table 2: Algorithm RepairPoint. The last clause does not apply under the inclusion constraints and is only added for completeness.

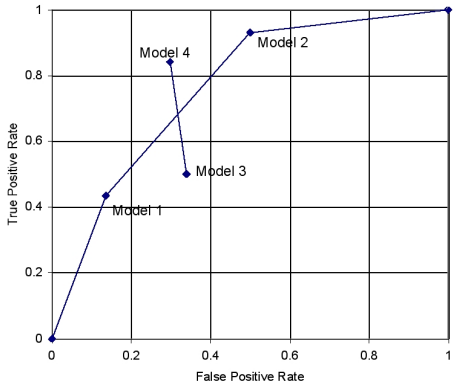

Figure 2: Model 3 is mirrored to Model 4 with the help of Models 1 and 2.

An equivalent construction is the following. Remove from the test set all instances classified positive by Model 1, and all instances classified negative by Model 2. We can imagine this as a smaller nested ROC space in which Model 1 represents (0,0) and Model 2 represents (1,1); because of the inclusion constraints the position of Model 3 remains unchanged. Note that Model 3 performs worse than a random model in this nested ROC space. We then construct Model 4 as in Figure 1, by inverting the predictions of Model 3 on the remaining test examples.

We end this section by noting that the true and false positive rates derived for Model 4 only hold for the same test set on which Models 1, 2 and 3 were evaluated. On a second, independent test set the ROC locations of the 4 classifiers may be different – in particular, Model 3 may not be below the

line connecting Models 1 and 2, in which case Model 4 will be evaluated worse than Model 3. In our experiments, we therefore use validation sets to decide whether concavities are stable across different samples. This will be further discussed in Section 4.

3 Identifying and repairing concavities in a ROC curve

The previous section outlined the main ideas underlying repairing concavities in ROC curves. However, preliminary experiments indicated that the three-point approach is too crude to work well in practice. In this section we introduce our main algorithm, which manipulates a whole section of a ROC curve. A ROC curve is obtained by evaluating a probabilistic classifier on a test set and varying the decision threshold, resulting in a step curve [Hand and Till, 2001]. An efficient way of constructing this curve is by ranking the instances corresponding to their predicted probability of being positive [Fawcett, 2003]. Figure 3 shows both the ROC curve for a probabilistic model evaluated on a small test set[1] and its convex hull [Provost and Fawcett, 2001].

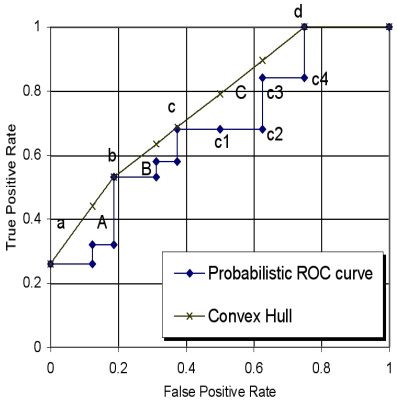

Figure 3: A probabilistic ROC curve and its convex hull.

Four points of the ROC curve (point a, b, c and d) are located on the convex hull. Each of these points corresponds to a probability threshold, and thus each segment of the convex hull corresponds to a probability interval. For instance, the convex hull in Figure 3 has three segments corresponding to three disjoint probability intervals. Three out of these five segments delineate concave regions of the ROC curve, indicated as A, B and C. For example, area A is delineated by line ab and the ROC curve between point a and point b. In general, a concave area means that the ranking obtained from the probabilistic model in this probability interval is worse than random. For instance, consider area C which is the largest concavity. One way to repair this concavity is by ignoring the score calculated by the probabilistic model in this interval, and output a constant score (e.g., the mid-point of the interval). Assuming that ties are broken by assigning a random rank, this would replace the concave region of the ROC curve with the line segment cd. It is interesting to note that if this procedure is followed for all concavities, this corresponds to constructing the convex hull by discretising the probability scores.

However, in theory we should be able to do better than that: we can invert the ranking of the instances in the probability interval, which can be seen as applying Algorithm RepairPoint to all thresholds in this interval. For instance, by applying this algorithm to the model corresponding to threshold c2, this point would be point-mirrored through the midpoint on line segment cd to the other side of the convex hull. The same can be done for the other thresholds. The resulting ROC curve is shown in Figure 4. We can note that the area C under the curve has been replaced by an equally large area C′ above the curve. The AUC of the repaired curve is therefore larger than both the AUC of the original curve and the AUC of its convex hull.

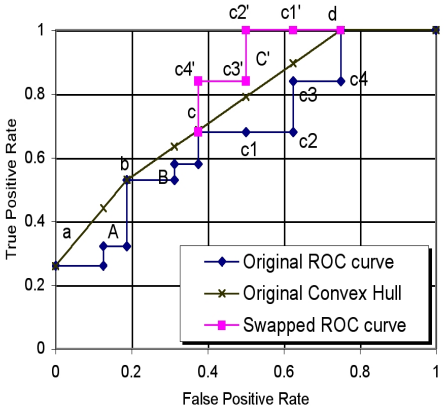

Figure 4: Mirroring a concave part of the ROC curve.

The algorithm to produce the model with the repaired ROC curve is given in Table 3. The procedure works for any model that calculates a score; a probabilistic model is a special case.

Given a scoring model M and two thresholds $T_1 > T_2$, construct a scoring model M' predicting scores as follows. Let S be the score predicted by M:
1. If $S > T_1$, then predict S;
2. If $S < T_2$, then predict S;
3. Otherwise, predict $T_1 + T_2 - S$.

Table 3: Algorithm RepairSection. The algorithm effectively inverts the ranking of all instances whose score as predicted by M falls in the interval $T_1 < S < T_2$.

[1] For illustration purposes, the test set contains only a few examples and the resolution of the ROC curve is low. In practical circumstances a step curve with much higher resolution is obtained.

4 Experimental evaluation

We describe a number of experiments to evaluate our approach. We used 23 two-class data sets from the UCI repository [Blake and Merz, 1998]. Table 4 shows their numbers of attributes, numbers of examples, and relative size of the majority class.

Dataset	#Attrs	#Exs	%MajClass
Australia	14	690	55.51
Sonar	60	208	51.92
Glass	9	214	67.29
German	20	1000	69.40
Car	6	1728	69.68
Anneal	38	798	74.44
Monk1	6	566	50.00
Monk2	6	601	65.72
Monk3	6	554	55.41
Hepatitis	19	155	78.71
House	16	435	62.07
Tic-tac-toe	9	958	64.20
Heart	13	270	55.56
Ionosphere	34	351	64.10
Breast Cancer	9	286	70.28
Lymphography	17	148	56.81
Primary Tumor	17	339	55.75
Soybean-large	35	683	55.51
Solar-Flare	12	323	56.35
Hayes-Roth	4	133	60.91
Credit	15	690	55.51
Balance	4	625	53.92
Bridges	12	108	66.67

Table 4: UCI data sets used in our experiments.

The experimental procedure for the first experiment was as follows. We split a data set into ten folds and use eight of them for training, one for validation and one for testing. We trained a naive Bayes model and a decision tree model[2] on the training data, and chose two thresholds that delineate a concavity. We then produced a new model by repairing the probabilities between the thresholds; we only use the new model if it improves AUC on the validation set. The detailed procedure is given in Table 5.

We ran Experiment RepairSection ten times and obtained m pairs of AUC values. Since we use a validation set, we are able to decide whether or not repair resulted in a better model – if not, we discard it and use the original, unrepaired model. For this reason, we only report results on the non-discarded models, so m may be smaller than 100. The average AUCs of the unrepaired curve and the repaired curve for each data set are given in Tables 6 and 7. We also performed a paired t-test with $m-1$ degrees of freedom and level of confidence 0.1 to test the significance of the average difference in AUC. The results are favourable: the significance tests yield 10 wins and 3 losses for repaired naive Bayes models and 11 wins and 5 losses for repaired decision tree models.

Experiments without using a validation set yielded worse results, so the use of a validation set appears crucial. We

[2]Notice that decision trees can be viewed as scoring classifiers [Ferri *et al.*, 2002; Provost and Domingos, 2003].

1. Train a naive Bayes or decision tree model M on the training data; construct a ROC curve C and its convex hull H on the training data.
2. Find adjacent points on H such that in this interval the area between C and H is largest. Let T_1 and T_2 be the corresponding score thresholds.
3. Produce a new probabilistic model M' by calling RepairSection(T_1, T_2).
4. Evaluate M and M' on the validation set, construct their ROC curves and calculate their AUCs. If AUC(M') $\leq$ AUC(M) then go to 6.
5. Evaluate M and M' on the test set, construct their ROC curves and calculate their AUCs.
6. Go to 1. until each fold has been used as a test set.

Table 5: Experiment RepairSection.

Dataset	AUC (Original)	AUC (Repaired)	Better ?
Australia	90.9 ± 0.88	90.6 ± 0.94	$\times$
Sonar	77.5 ± 1.30	76.8 ± 1.31	
Glass	76.6 ± 1.43	80.6 ± 1.36	$\vee$
German	79.3 ± 0.88	78.9 ± 0.88	$\times$
Car	99.0 ± 0.079	99.0 ± 0.08	
Anneal	86.7 ± 0.48	90.2 ± 0.47	$\vee$
Monk1	75.4 ± 0.70	77.4 ± 0.70	$\vee$
Monk2	64.6 ± 0.71	66.1 ± 0.92	$\vee$
Monk3	96.2 ± 0.29	97.7 ± 0.24	$\vee$
Hepatitis	86.3 ± 1.94	85.1 ± 1.22	
House	96.3 ± 0.35	96.3 ± 0.35	
Tic-Tac-Toe	74.1 ± 0.61	75.6 ± 0.58	$\vee$
Heart	91.0 ± 1.04	90.2 ± 1.08	$\times$
Ionosphere	93.2 ± 0.69	93.1 ± 0.690	
Breast Cancer	76.1 ± 1.63	77.7 ± 1.57	$\vee$
Lymphography	90.3 ± 1.39	91.0 ± 1.49	$\vee$
Primary Tumor	79.6 ± 1.02	80.6 ± 1.03	$\vee$
Soybean-Large	91.9 ± 0.46	91.9 ± 0.46	
Solar-Flare	91.1 ± 0.71	91.2 ± 0.74	
Hayes-Roth	89.4 ± 1.53	91.3 ± 1.58	$\vee$
Credit	90.8 ± 0.68	90.7 ± 0.65	
Balance	98.4 ± 0.27	98.4 ± 0.29	
Bridges*	92.7 ± 1.66	92.9 ± 1.89	
Average	86.41	87.1	

Table 6: Results of Experiment RepairSection with naive Bayes. *For the Bridges data set we used 5-fold cross-validation.

therefore conducted an experiment with two validation folds: we would only use the repaired model if its AUC was higher on both validation folds. The results for naive Bayes are shown in Table 8. We now obtain 12 significant wins and only 1 significant loss, and the average increase in accuracy is more than a percentage-point. Interestingly, two validation folds didn't work well for decision trees.

Finally, we conducted an experiment with naive Bayes whereby we selected all concavities, and repaired those that occurred on two validation sets. The results were similar to the results in the first experiment (11 significant wins and three losses), with some of the wins and losses occurring on

Dataset	AUC (Original)	AUC (Repaired)	Better ?
Australia	82.27 ±0.50	82.28 ±0.49	
Sonar	87.35 ±1.20	88.92 ±1.01	∨
Glass	76.09 ±1.34	80.69 ±1.15	∨
German	80.78 ±0.55	81.01 ±0.54	∨
Car	99.05 ±0.074	99.07 ±0.081	
Anneal	86.86 ±0.41	90.64 ±0.37	∨
Monk1	75.82 ±0.74	78.05 ±0.704	∨
Monk2	66.46 ±0.66	67.95 ±0.662	∨
Monk3	96.30 ±0.32	98.14 ±98.14	∨
Hepatitis	91.92 ±2.84	91.92 ±2.84	
House	96.18 ±0.31	96.66 ±0.29	∨
Tic-Tac-Toe	74.81 ±0.54	75.97 ±0.56	∨
Heart	91.72 ±1.34	90.77 ±1.21	×
Ionosphere	96.57 ±0.91	95.88 ±0.62	
Breast Cancer	74.82 ±1.71	75.75 ±1.51	
Lymphography	87.71 ±1.86	87.88 ±1.89	∨
Primary Tumor	76.63 ±1.23	75.51 ±1.36	×
Soybean-Large	90.93 ±0.50	91.10 ±0.52	×
Solar-Flare	86.61 ±1.59	85.65 ±1.65	×
Hayes-Roth	86.87 ±2.24	88.33 ±2.08	∨
Credit	91.09 ±0.65	90.92 ±0.64	
Balance	99.37 ±0.18	99.42 ±0.189	
Bridges	85.57 ±4.36	81.71 ±3.35	×
Average	86.16	86.71	

Table 7: Results of Experiment RepairSection with decision trees.

Dataset	AUC (Original)	AUC (Swapped)	Better ?
Australia	87.17±2.28	86.86±2.33	×
Sonar	81.34±2.46	84.61±2.66	∨
Glass	74.67±1.61	74.75±1.42	∨
German	82.33±0.62	82.72±0.60	∨
Car	99.25±0.091	99.29±0.098	
Anneal	87.06±0.53	90.32±0.48	∨
Monk1	75.68±1.12	78.65±0.92	∨
Monk2	65.74±0.89	67.24±0.95	∨
Monk3	96.75±0.47	98.54±0.25	∨
Hepatitis	92.67±3.79	93.19±3.37	
House	96.76±0.37	96.76±3.44	∨
Tic-Tac-Toe	75.62±0.69	76.92±0.71	∨
Heart	92.91±1.29	92.98±1.42	
Ionosphere	97.07±0.77	97.53±0.57	
Breast Cancer	78.20±2.43	79.16±2.12	
Lymphography	86.94±2.41	90.49±1.94	∨
Primary Tumor	78.14±1.31	80.19±1.27	∨
Soybean-Large	91.36±0.45	91.63±0.44	
Solar-Flare	89.69±1.90	89.94±1.88	
Hayes-Roth	84.89±3.56	88.33±3.09	∨
Credit	89.03±1.56	88.67 ±1.70	
Balance	99.53±0.027	99.10±0.078	
Average	86.49	87.63	

Table 8: Results with naive Bayes using two validation folds.

different data sets. It appears that repairing only the largest concavity and using two validation sets is the best strategy (at least for naive Bayes).

5 Discussion and conclusions

The work reported in this paper bears some similarity with ensemble methods. Bagging and boosting are two well-known ensemble approaches. Both approaches are implemented by re-sampling methods. In bagging [Breiman, 1996], the ensemble is formed by making bootstrap replicates of the training data sets and then multiple generated hypotheses are used to get an aggregated predictor. Boosting algorithms [Freund and Schapire, 1996] assign different weights to training instances depending on whether they are correctly classified. The approaches presented in this paper do not make use of re-sampling techniques.

Another relevant ensemble method is majority voting [Kimura and Shridar, 1991; Lam and Sue, 1997], in which the class predicted by the ensemble is the most predicted class among the base classifiers. Algorithm RepairPoint in this paper uses a kind of voting: when both Model 1 and Model 2 (see Figure 2) agree on the classification of an instance, then we choose that class. Otherwise, we choose the class not predicted by Model 3. In majority voting, on the other hand, we would choose the class predicted by Model 3. The difference is that majority voting does not take the quality of the different models into account, whereas our repair scheme knows that Model 3 is sub-optimal and therefore corrects it predictions in the relevant region.

ROC curves contain a wealth of information about the performance of one or more classifiers, which can be utilised to construct better models. They have been used to find optimal labelling of decision trees [Ferri *et al.*, 2002] and to find good decision thresholds for probabilistic classifiers [Lachiche and Flach, 2003]. In this paper we have proposed a novel approach to construct new models by repairing concavities in a ROC curve. The first method, RepairPoint, works on a probabilistic classifier with three probability thresholds, and tries to improve the poorest model with help of the other two. Preliminary experimental results (not reported) showed that this didn't work too well, but this may be due to the fact that the selection of the threshold for Model 3 is not easy. The threshold is a point chosen based on the ROC curve of the training data set; this point (see point c3 in Figure 3) has the farthest distance to the convex hull. If this threshold is not optimal on the test data (for instance, the position c3 in the ROC curve on the test data set unfortunately is located in the position c2), the AUC after repair becomes worse. Still, we believe that the idea of mirroring models around lines in ROC space will prove to be very useful. An interesting investigation for future work is whether a similar method can be made to work if the models are not obtained from a single scoring model (this would invalidate Theorem 1, i.e., the position of Model 4 may be different from the point obtained by point-mirroring).

The second method, RepairSection, locates and repairs an entire concave region of a ROC curve. Experimental results were very encouraging for both naive Bayes and decision trees. For naive Bayes we were able to improve results even further by using two validation folds, but this didn't work for decision trees. We are currently investigating why this is so. One possible explanation is that ROC curves ob-

tained from decision trees have lower resolution (because all instances in a leaf receive the same predicted probability), which may mean that concavities are less stable across samples. Pruning may be another factor, as it has been shown that pruning is detrimental for probability prediction [Provost and Domingos, 2003; Ferri *et al.*, 2003].

There are several other ways in which this work could be taken further. One is to investigate how much repair is possible, by concentrating on ROC curves with large concavities, possibly from artificial data sets. Another is to work with averaged ROC curves that are obtained by cross-validation (since each instance occurs in the test fold exactly once, an averaged ROC curve can be simply constructed by combining all instances with their predicted probabilities).

Acknowledgments

A preliminary version of this paper (without the experimental results with decision trees and two validation folds) appeared as [Flach and Wu, 2003]. We gratefully acknowledge the constructive comments made by the anonymous reviewers. We would also like to thank Rich Roberts for performing additional experiments.

References

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

[Blockeel and Struyf, 2002] H. Blockeel and J. Struyf. Deriving biased classifiers for improved ROC performance. *Informatica*, 26(1):77–84, 2002.

[Breiman, 1996] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Breiman, 1998] L. Breiman. Arcing classifiers. *Annals of Statistics*, 26(3):801–849, 1998.

[Fawcett, 2003] T. Fawcett. ROC graphs: Notes and practical considerations for data mining researchers. Technical Report HPL-2003-4, HP Laboratories, Palo Alto, CA, USA, 2003. `www.purl.org/net/tfawcett/papers/HPL-2003-4.pdf`.

[Ferri *et al.*, 2002] C. Ferri, P. Flach, and J. Hernandez-Orallo. Decision tree learning using the area under the ROC curve. In C. Sammut and A. Hoffman, editors, *Proceedings of the 19th International Conference on Machine Learning*, pages 139–146. Morgan Kaufmann, 2002.

[Ferri *et al.*, 2003] C. Ferri, P. Flach, and J. Hernandez-Orallo. Improving the AUC of probabilistic estimation trees. In N. Lavrač, D. Gamberger, L. Todorovski, and H. Blockeel, editors, *Proceedings of the 14th European Conference on Machine Learning*, volume 2837 of *Lecture Notes in Computer Science*, pages 121–132. Springer-Verlag, 2003.

[Flach and Wu, 2003] P. Flach and S. Wu. Repairing concavities in ROC curves. In J. Rossiter and T. Martin, editors, *Proceedings of the 2003 UK Workshop on Computational Intelligence*, pages 38–44. University of Bristol, 2003.

[Freund and Schapire, 1996] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In L. Saitta, editor, *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

[Hand and Till, 2001] D. Hand and R. Till. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.

[Jacobs *et al.*, 1991] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[Kimura and Shridar, 1991] F. Kimura and M. Shridar. Handwritten numerical recognition based on multiple algorithms. *Pattern Recognition*, 24(10):969–983, 1991.

[Lachiche and Flach, 2003] N. Lachiche and P. Flach. Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. In T. Fawcett and N. Mishra, editors, *Proceedings of the 20th International Conference on Machine Learning*, pages 416–423. AAAI Press, 2003.

[Lam and Sue, 1997] L. Lam and C. Sue. Application of majority voting to pattern recognition: An analysis of its behaviour and performance. *IEEE Transactions on Systems, Man and Cybernetics*, 27(5):553–568, 1997.

[Provost and Domingos, 2003] F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(3):199–215, 2003.

[Provost and Fawcett, 2001] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

[Valentini and Masulli, 2002] G. Valentini and F. Masulli. Ensembles of learning machines. In M. Marinaro and R. Tagliaferri, editors, *13th Italian Workshop on Neural Nets*, volume 2486 of *Lecture Notes in Computer Science*, pages 3–22. Springer-Verlag, 2002.