A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods

Bernard Ženko, Ljupčo Todorovski, and Sašo Džeroski
Department of Intelligent Systems, Jožef Stefan Institute
Jamova 39, Ljubljana, Slovenia
{Bernard.Zenko, Ljupco.Todorovski, Saso Dzeroski}@ijs.si

Abstract. *Meta decision trees (MDTs) are a method for combining multiple classifiers. We present an integration of the algorithm MLC4.5 for learning MDTs into the Weka data mining suite. We compare classifier ensembles combined with MDTs to bagged and boosted decision trees, and to classifier ensembles combined with other methods: voting and stacking with three different meta-level classifiers (ordinary decision trees, naive Bayes, and multi-response linear regression - MLR).*

Meta decision trees. Techniques for combining predictions obtained from multiple base-level classifiers can be clustered in three combining frameworks: voting (used in bagging and boosting), stacked generalization or stacking [7] and cascading. Meta decision trees (MDTs) [5] adopt the stacking framework of combining base-level classifiers. The difference between meta and ordinary decision trees (ODTs) is that MDT leaves specify which base-level classifier should be used, instead of predicting the class value directly. The attributes used by MDTs are derived from the class probability distributions predicted by the base-level classifiers for a given example. An example MDT, induced in the image domain from the UCI Repository, is given below. The leaf denoted by an asterisk (*) specifies that the IBk classifier is to be used to classify an example, if the entropy of the class probability distribution predicted by IBk is smaller than or equal to 0.002369.

```
IBk:Entropy <= 0.002369:  IBk (*)
IBk:Entropy > 0.002369
|    J48:maxProbability <= 0.909091:  IBk
|    J48:maxProbability > 0.909091:  J48
```

The original algorithm MLC4.5 [5] for inducing MDTs was an extension of the C4.5 [3] algorithm for induction of ODTs. We have integrated the algorithm for inducing MDTs in the Weka data mining suite [6]. We have implemented MLJ4.8, a modification of J4.8 (the Weka re-implementation of C4.5): the differences between MLJ4.8 and J4.8 closely mirror the ones between MLC4.5 and C4.5. Integrating MDTs into Weka lets us perform a variety of experiments in combining different sets of base level classifiers, as well as comparisons to other methods for combining classifiers.

Experimental setup. In order to compare the performance of MDTs with that of other combining schemes, we perform experiments on a collection of twenty-one data sets from the UCI Repository of Machine Learning Databases and Domain Theories. Three learning algorithms are used in the base-level experiments: the tree learning algorithm J4.8, which is a re-implementation of C4.5 [3], the k-nearest neighbor (k-NN or IBk) algorithm and the naive Bayes (NB) algorithm. In all experiments, classification errors are estimated using 10-fold stratified cross validation. Cross validation is repeated ten times using different random generator seeds resulting in ten different sets of folds.

At the meta-level, the performances of seven algorithms for combining classifiers are compared. These are bagging and boosting of decision trees, voting, stacking with three different meta-level learning algorithms (J4.8, naive Bayes, and MLR), and stacking with MDTs. The performance of each of these algorithms is assessed in terms of its error rate. The performance of MDTs is compared to that of the other combining approaches. The relative accuracy improvement of classifier C_1 as compared to classifier C_2 is $1 - \texttt{error}(C_1)/\texttt{error}(C_2)$ (in our case C_1 = MDTs). The average relative improvement is calculated using geometric mean: $1 - \texttt{geometric\_mean}(\texttt{error}(C_1)/\texttt{error}(C_2))$. The statistical significance of the difference in classification errors is tested using the paired t-test (exactly the same folds are used for C_1 and C_2) with significance level of 95%.

Results. Stacking with MDTs performs better than bagging and boosting of decision trees, which are the state of the art methods for learning ensembles of classifiers: In both cases MDTs are significantly better in 11 and worse in 3 domains, with a 20% and 15% relative accuracy improvement, respectively. A previous study of MDTs [5] shows that MDTs

0-7695-1119-8/01 $17.00 © 2001 IEEE

Table 1. The performance of stacking with MDTs (error rate in %); the relative improvement in accuracy (in %) achieved by stacking with MDTs as compared to bagging, boosting, voting, stacking with J4.8, naive Bayes and MLR; and its significance (+/–: significantly better/worse, x: insignificant).

	Sta. MDT	Bag. J48		Boo. J48		Voting		Sta. J48		Sta. NB		Sta. MLR	
Data set	abs. err.	rel. im.	sig.	rel. im.	sig.	rel. im.	sig.	rel. im.	sig.	rel. im.	sig.	rel. im.	sig.
australian	13.77±0.38	-0.74	x	11.63	x	0.31	x	5.75	x	4.04	x	2.76	x
balance	8.51±0.19	50.83	+	60.39	+	4.49	+	-41.49	–	7.16	+	10.14	+
breast-w	2.69±0.07	45.98	+	27.41	+	22.31	+	3.09	+	6.93	+	1.57	+
bridges-td	16.08±0.84	-7.89	–	17.17	+	-1.86	–	4.09	+	7.34	+	-13.89	–
car	5.02±0.27	25.96	+	-20.75	–	22.73	+	-208.54	–	-89.30	–	10.62	+
chess	0.60±0.05	1.55	x	-56.55	x	59.10	x	20.42	x	20.42	x	0.00	x
diabetes	24.74±0.54	-0.48	x	13.28	x	-3.04	x	3.85	x	2.01	x	-4.05	x
echo	27.71±0.76	12.53	+	18.24	+	5.22	+	-4.31	–	1.09	+	3.20	+
german	25.60±0.30	2.92	+	12.42	+	-1.63	–	-0.51	–	5.50	+	-5.09	–
glass	31.78±1.19	-22.08	–	-37.10	–	-7.09	–	17.68	+	37.21	+	-2.72	–
heart	16.04±0.46	18.91	+	26.36	+	6.28	+	8.84	+	5.25	+	-4.84	–
hepatitis	15.87±0.84	10.22	x	13.07	x	8.89	x	16.04	x	8.55	x	-1.23	x
hypo	0.79±0.07	-1.62	x	24.62	x	40.09	x	4.56	x	32.34	x	-9.61	x
image	2.53±0.09	0.68	x	-37.65	x	13.72	x	22.92	x	61.16	x	10.82	x
ionosphere	8.83±0.62	-12.73	–	-37.78	–	-23.02	–	-44.86	–	-24.00	–	-20.16	–
iris	4.73±0.42	17.44	+	18.39	+	-12.70	–	22.83	+	5.33	+	-5.97	–
soya	7.06±0.14	2.43	x	0.21	x	-4.55	x	12.04	x	-7.59	x	2.23	x
tic-tac-toe	0.96±0.06	85.87	+	72.04	+	89.60	+	-130.00	–	20.69	+	-64.29	–
vote	3.54±0.17	9.94	+	21.03	+	50.16	+	12.99	+	30.00	+	0.00	x
waveform	14.40±0.11	20.00	+	22.50	+	9.44	+	-0.15	–	4.20	+	-0.53	–
wine	3.26±0.60	36.26	+	19.44	+	-87.10	–	14.71	+	6.45	+	-13.73	–
Average	11.17±0.39	19.89		14.78		18.34		-4.24		10.59		-4.07	
W/L		11+/3–		11+/3–		8+/6–		7+/7–		12+/2–		4+/9–	

perform better than voting and stacking with ODTs. Our study confirms these findings and proves that they are independent of a specific implementation (we used their re-implementation in Java programming language) and the set of base-level classifiers (we used a different and smaller set). (Comparing MDTs to ODTs shows a 4% decrease in accuracy, but this is mostly due to the data sets car and tic-tac-toe, where all combining methods perform very well: if we exclude these two data sets a 7% increase is obtained; MDTs are also much smaller than ODTs).

Stacking with naive Bayes performs poorly. Stacking with MLR slightly outperforms stacking with MDTs (a 4% relative improvement in accuracy). Note that stacking with MDTs performs comparably while using less information (only aggregate data on the class probability distribution is used by MDTs, while the complete class probability distribution is used by MLR). The attributes used in MDTs are domain independent once we fix the set of base-level classifiers and the language of MDTs is the same for all domains. Another advantage of the MDTs is their understandability: they provide information about the relative areas of expertise of the base-level classifiers.

References

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

[2] Freund, Y. and Schapire, R. E. (1996) Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148-156. Morgan Kaufmann, San Francisco.

[3] Quinlan, J. R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco.

[4] Ting, K. M. and Witten, I. H. (1999) Issues in stacked generalization. *Journal of Artificial Intelligence Research*, 10: 271–289.

[5] Todorovski, L. and Džeroski, S. (2000) Combining multiple models with meta decision trees. In *Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, pages 54–64. Springer, Berlin.

[6] Witten, I. H. and Frank, E. (1999) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco.

[7] Wolpert, D. (1992) Stacked generalization. *Neural Networks* 5(2): 241–260.