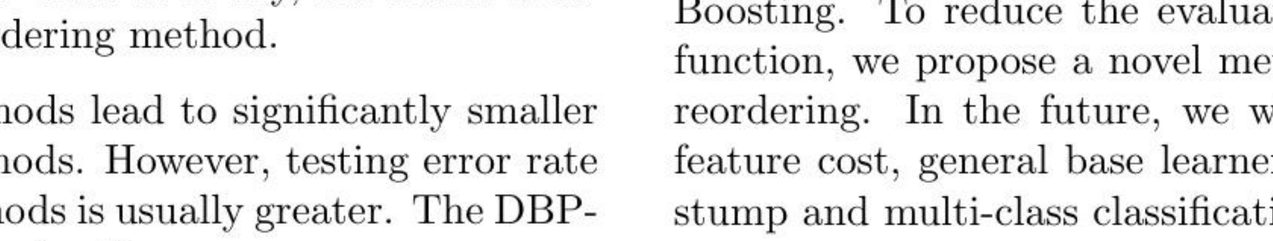
Table 1. Results for FastExit with natural order and three reordering methods: SORT, TD and SIFTING. Results for original Boosting, as baseline, is shown at first row. At each entry, the first cell is EPL and the second is error rate, both in testing stage. Lower EPL in **boldface**.

	dataset#1		dataset#2		dataset#3		dataset#4		dataset#5		dataset#6		dataset#7		dataset#8		dataset#9		dataset#10	
Boosting	26	0.83%	136	3.29%	54	0.28%	87	0.29%	220	1.32%	314	1.9%	518	4.02%	256	3.31%	475	4.68%	22	0.00%
fastexit	11.54	0.83%	72.79	3.29%	27.01	0.28%	40.37	0.29%	138.88	1.32%	222.41	1.9%	409.09	4.02%	179.03	3.31%	409.30	4.68%	6.68	0.00%
fastexit-SORT	9.37	0.83%	54.74	3.29%	23.14	0.28%	32.15	0.29%	119.86	1.32%	200.83	1.9%	370.52	4.02%	163.52	3.31%	311.35	4.68%	4.98	0.00%
fastexit-TD	9.56	0.83%	56.27	3.29%	24.31	0.28%	34.73	0.29%	130.60	1.32%	237.48	1.9%	425.62	4.02%	172.82	3.31%	449.77	4.68%	5.18	0.00%
fastexit-SIFT	**9.07**	0.83%	**50.86**	3.29%	**22.25**	0.28%	**31.00**	0.29%	**115.40**	1.32%	**190.16**	1.9%	**349.95**	4.02%	**159.30**	3.31%	**223.02**	4.68%	**4.59**	0.00%

Table 2. Results for FastExit and BDD with reordering by SIFTING. At each entry, the first cell is EPL and the second is error rate, both in testing stage.



Given a fixed T' for the approximate representation DBPSYN, we find none of the three reordering methods is able to have both lower EPL and lower test error. Therefore, for each reordering method we vary the T' to generate a sequence of (EPL, test error) pairs and connect those points as a curve (*i.e.*, the Pareto front), as is shown in Figure 7 (More results in supplement). The closer to bottom left corner the curve, the better the reordering method.[2]

As can be seen in Table 1 and Figure 7, all three reordering methods yield improved EPL. Besides, we find that:

1) SIFT $>$ SORT $>$ TD on a majority of the datasets for both exact and approximate representation, where $>$ means outperform. That is to say, SIFTING is always a preferred reordering method.

2) Approximate methods lead to significantly smaller EPL than exact methods. However, testing error rate for approximate methods is usually greater. The DBPSYN seems a good trade off.

We also compare two exact representations, *i.e.*, FastExit and BDD, with reordering by SIFTING. Basically, in BDD we need store too many nodes which is beyond our computational resources. So we implement a fixed small step LogitBoost. After training LogitBoost and combining the duplicate decision stumps, the outputs of base learners can thus be quantized to integers. This largely reduces the space complexity for BDD. Even so, it is still space demanding and the base learner swapping is very slow. In Table 2, we report the results for part of the datasets.

From Table 2, it seems that BDD shows little improvements over its close relative FastExit. We believe it's due to the fact that in BDD the redundant nodes within $\mathcal{PI}$ are actually very rare. Recalling the connection between BDD and the work in (Busa-Fekete et al., 2012) (section 5.1), we thus cautiously suggest that the Action "Skip" in MDP (Busa-Fekete et al., 2012) be not necessary provided a proper base learner ordering be found (In (Busa-Fekete et al., 2012) the natural order is taken and no reordering is attempted).

7. Conclusion and Future Work

We discuss the representation of decision function in Boosting. To reduce the evaluation time of decision function, we propose a novel method for base learner reordering. In the future, we will take into account feature cost, general base learner other than decision stump and multi-class classification.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61225008, 61020106004, 61021063, and by the Ministry of Education of China under Grant 20120002110033. Thanks Scott Sanner for helpful discussion and pointing to us the important role of data structure in machine learning. Thanks Tengyu Sun, Han Hu, Jianjiang Feng for reading the manual script. Thanks all anonymous reviewers' comments that significantly improve this paper's quality. Thanks Minmin Chen for clarifying discussion.

[2] Comparison using Pareto front by varing T' is suggested by a reviewer.