Cost-Sensitive Decision Trees with Pre-pruning

Jun Du[1], Zhihua Cai[1,2], and Charles X. Ling[2]

[1] School of Computer Science
China University of Geosciences, Wuhan, 430074, P.R. China
[2] Dept. of Computer Science
The University of Western Ontario, London, Ontario, N6A 5B7, Canada
j.du@hotmail.com, zcai6@uwo.ca, cling@csd.uwo.ca

Abstract. This paper explores two simple and efficient pre-pruning strategies for the cost-sensitive decision tree algorithm to avoid overfitting. One is to limit the cost-sensitive decision trees to a depth of two. The other is to prune the trees with a pre-specified threshold. Empirical study shows that, compared to the error-based tree algorithm C4.5 and several other cost-sensitive tree algorithms, the new cost-sensitive decision trees with pre-pruning are more efficient and perform well on most UCI data sets.

1 Introduction

For most previous research on classification, the main goal is to develop algorithms that minimize the number of errors on previously unseen examples. This is valid only when the costs of different errors are equal. In many real-world applications, however, it is far from the case. For example, in medical diagnosis, the errors for diagnosing someone as healthy carries a very high cost when that person in fact has a life-threatening disease, compared to the cost from mistakenly diagnosing a healthy one as having the disease. Cost sensitive classification deals with such cases where misclassification costs are not equal.

Generally, there are three main types of strategies for cost sensitive classification, implemented by manipulating one of the three components respectively: the train data, the learning algorithm, and the output of the learned model [1]. Many approaches have been developed in the past few years in making the traditional cost-insensitive classification algorithms cost-sensitive. For example, [2], [3], [4] discussed neural networks for cost-sensitive classification; [5] and [6] worked on cost-sensitive evolutionary algorithm; [7] made support vector machines sensitive to the cost; [8], [9] and [10] focused on the ensemble techniques such as bagging and boosting; decision tree algorithms, one of the most popular machine learning techniques, have also been studied fo cost-sensitivity.

Current research in cost-sensitive decision trees falls into two categories. The first category is concerned with making the attribute splitting criterion sensitive to cost [11,12,13]. The other category develops new or modified pruning algorithms to minimize the expected cost [14,15]. In this paper, we propose two simple and efficient pre-pruning strategies for cost-sensitive decision

Z. Kobti and D. Wu (Eds.): Canadian AI 2007, LNAI 4509, pp. 171–179, 2007.

trees to avoid overfitting. Compared to the classical error-based tree algorithm C4.5 and several other cost-sensitive tree algorithms, our cost-sensitive decision trees with pre-pruning are more efficient and perform well on most UCI data sets.

The rest of the paper is organized as follows. In section 2, we first review unpruned cost-sensitive decision tree [13], which uses an attribute splitting criterion for reducing the total cost including the misclassification cost and test cost. Our new pre-pruning cost-sensitive trees are proposed after that. Then, we present our experiment results in section 3. Finally, section 4 draws conclusions and suggests future work.

2 Cost-Sensitive Decision Trees

2.1 Unpruned Cost Reduction Based Decision Tree

[13] proposes a new attribute splitting criterion for building cost-sensitive decision trees by minimizing the sum of misclassification and test cost [16]. In the decision tree building process, the algorithm directly chooses an attribute that reduces and minimizes the total cost (the sum of the misclassification cost and test cost) for the split, instead of choosing an attribute that minimizes the entropy (as in C4.5).

Similar to the traditional error-based decision tree algorithms, this cost minimization algorithm may have the deficiency of overfitting the training examples. That is, when a decision tree is built, some branches may be built reflecting anomalies in the train data due to noise or outliers, and this often leads to good performance on the train data but bad on test data. Pre-pruning and post-pruning are two typical methods to remove the least reliable branches and generally result in faster and better classification ability for independent test data.[1]

2.2 Cost-Sensitive Decision Trees with Pre-pruning

The algorithms we proposed in this paper are based on [13], incorporating two simple pre-pruning methods, described below.

2-Level Tree. With this approach, we just build the tree with no more than 2 levels. [18] and [19] have used similar approaches for error-based tree building, and shown that simpler trees often work quite well in many data sets. In this paper, we use the same idea in the cost-sensitive tree building process. The

[1] [13]'s work includes both misclassification costs and attribute costs. Attribute costs can act as a natural pruning mechanism, because an expensive attribute is unlikely to be chosen to split the data further, unless there is a large gain in the reduction of the misclassification cost. Nevertheless, overfitting could still happen, especially when the attribute cost is small or zero (as we study here). [17] incorporates post-pruning in cost-sensitive decision trees.

empirical study in section 3 will show that indeed the simple approach works quite well.

Threshold Pruning Tree. Another common approach for pre-pruning is imposing a pre-specified threshold on the splitting measure. Using cost reduction alone, the unpruned tree [13] would be expanded until the cost reduction is smaller than or equal to 0. We set a threshold on the cost reduction to avoid overfitting. We assume that the tree expansion is worthwhile only when the cost reduction is greater than the sum of False Positive(FP) and False Negative(FN) cost (we assume that the cost of True Positive and True Negative is 0). That is:

$$Threshold = FP + FN$$

For cost-sensitive trees with both pre-pruning methods, the following is used to label leaves. If the cost reduction is 0 or negative (for the 2-level trees), or if the cost reduction is less than the threshold (pre-specified threshold pruning), a leaf node is formed, and it should be labeled as the class minimizing the expected cost according to train data falling into the node. If no instance is falling into a node, then a leaf is also formed labeled as the class minimizing the expected cost of its parent node.

3 Empirical Study

3.1 Configuration

We conduct experiments on the new algorithms above and compare them against the classical error-based algorithm C4.5 and cost-sensitive algorithms including

Table 1. UCI data sets used in the empirical study

Data set	No. of attributes	No. of examples	Class distribution
Breast-cancer	9	277	196/81
Breast-w	9	699	458/241
Colic	22	368	232/136
Credit-a	15	690	307/383
Credit-g	20	1000	700/300
Diabetes	8	768	500/268
Heart-statlog	13	270	150/120
Hepatitis	19	155	32/123
Ionosphere	34	351	126/225
Kv-vs-kp	36	3196	1669/1527
Labor	17	57	20/37
Sick	29	3772	3541/231
Sonar	60	208	97/111
Vote	16	435	267/168

the original unpruned cost reduction tree [13], weighting approach (with and without minimum expected cost) [20] and MetaCost [9], all implemented in Weka [21]. Only misclassification cost is considered in this paper (other types of cost will be studied in the future work).

14 data sets from UCI Machine Learning Repository [22] are used in the empirical study, all of which have discrete attributes and binary class without missing values. Information on these data sets is tabulated in Table 1.

Five cost matrices are used on these UCI data sets to evaluate the effects of different cost ratios. The costs of true positive and true negative are always set as 0, while false positive is always set to 1, and false negative cost is set to be 2, 5, 10, 20 and 50. This makes the cost ratio as 2, 5, 10, 20 and 50, which is supposed to show algorithms' performance with different cost ratios.

Under each cost matrix, 10-fold cross validation is performed on each data set. The experiment is repeated for 10 times and the average cost (total cost divided by the size of the test data) is recorded in the final results. Two-tailed t-test with a 95% confidence level is conducted to examine statistical significance.

3.2 Experiments Results

A total of 7 algorithms are compared; they are: pruned C4.5 labeled as "C4.5(P)", unpruned cost reduction tree [13] labeled as "CR", instance-weighting approach (with and without minimum expected cost) [20] with C4.5 labeled as "C4.5cs-mc"and "C4.5cs", MetaCost [9] with C4.5 labeled as "MetaCost", the proposed 2-level tree labeled as "CR-2", and the proposed threshold pruning tree labeled as "CRPrune". Table 2 lists the average misclassification cost, and Table 3 lists the corresponding summary on the t-test. Each entry $w/t/l$ in Table **??** means that the algorithm at the corresponding row wins in w data sets, ties in t data sets, and loses in l data sets, compared to the algorithm at the corresponding column. The same notation is used in Table 5.

Table 2. Average misclassification cost on UCI data sets

Cost Ratio = 2							
Data set	C4.5(P)	CR	C4.5cs	C4.5cs-mc	MetaCost	CR-2	CRPrune
breast-cancer	0.046	0.047	0.050	0.048	0.047	0.047	0.047
breast-w	0.009	0.008	0.008	0.008	0.008	0.009	0.007
colic	0.026	0.029	0.027	0.027	0.026	0.029	0.029
credit-a	0.023	0.024	0.021	0.022	0.021	0.021	0.020
credit-g	0.045	0.047	0.042	0.043	0.042	0.042	0.040
diabetes	0.046	0.045	0.036	0.040	0.040	0.039	0.039
heart-statlog	0.032	0.046	0.030	0.032	0.033	0.041	0.040
hepatitis	0.026	0.035	0.025	0.027	0.023	0.027	0.028
ionosphere	0.015	0.014	0.014	0.014	0.013	0.018	0.012
kr-vs-kp	0.001	0.027	0.001	0.001	0.001	0.034	0.028
labor	0.018	0.024	0.013	0.015	0.018	0.018	0.026
sick	0.003	0.003	0.003	0.003	0.004	0.003	0.003
sonar	0.041	0.049	0.040	0.040	0.037	0.035	0.043
vote	0.006	0.006	0.006	0.006	0.006	0.006	0.006

Table 2. (*Continued*)

Cost Ratio = 5							
Data set	C4.5(P)	CR	C4.5cs	C4.5cs-mc	MetaCost	CR-2	CRPrune
breast-cancer	0.110	0.094	0.083	0.076	0.074	0.068	0.068
breast-w	0.018	0.017	0.014	0.015	0.014	0.018	0.012
colic	0.058	0.050	0.050	0.052	0.053	0.048	0.045
credit-a	0.049	0.073	0.030	0.037	0.032	0.037	0.036
credit-g	0.099	0.087	0.062	0.068	0.065	0.060	0.063
diabetes	0.105	0.082	0.052	0.054	0.050	0.051	0.052
heart-statlog	0.066	0.106	0.051	0.054	0.052	0.065	0.063
hepatitis	0.047	0.060	0.021	0.026	0.025	0.026	0.021
ionosphere	0.022	0.027	0.012	0.019	0.015	0.019	0.015
kr-vs-kp	0.002	0.034	0.002	0.002	0.002	0.034	0.034
labor	0.027	0.041	0.023	0.014	0.018	0.016	0.019
sick	0.008	0.006	0.005	0.006	0.006	0.006	0.005
sonar	0.078	0.096	0.047	0.069	0.051	0.054	0.063
vote	0.011	0.009	0.009	0.010	0.010	0.009	0.009
Cost Ratio = 10							
Data set	C4.5(P)	CR	C4.5cs	C4.5cs-mc	MetaCost	CR-2	CRPrune
breast-cancer	0.216	0.181	0.072	0.075	0.071	0.069	0.071
breast-w	0.032	0.023	0.019	0.019	0.022	0.014	0.015
colic	0.112	0.139	0.076	0.076	0.068	0.081	0.080
credit-a	0.092	0.144	0.039	0.056	0.051	0.047	0.047
credit-g	0.189	0.099	0.068	0.088	0.073	0.074	0.070
diabetes	0.204	0.145	0.060	0.073	0.065	0.063	0.058
heart-statlog	0.123	0.197	0.064	0.076	0.065	0.070	0.057
hepatitis	0.083	0.108	0.021	0.026	0.021	0.034	0.021
ionosphere	0.033	0.048	0.025	0.025	0.016	0.022	0.014
kr-vs-kp	0.003	0.034	0.004	0.004	0.004	0.034	0.034
labor	0.042	0.072	0.035	0.016	0.021	0.019	0.035
sick	0.015	0.008	0.007	0.010	0.009	0.009	0.008
sonar	0.138	0.179	0.047	0.091	0.062	0.077	0.048
vote	0.020	0.015	0.019	0.017	0.015	0.016	0.015
Cost Ratio = 20							
Data set	C4.5(P)	CR	C4.5cs	C4.5cs-mc	MetaCost	CR-2	CRPrune
breast-cancer	0.427	0.346	0.071	0.080	0.071	0.084	0.077
breast-w	0.061	0.051	0.022	0.031	0.028	0.021	0.019
colic	0.218	0.262	0.066	0.085	0.064	0.088	0.070
credit-a	0.177	0.272	0.046	0.053	0.044	0.045	0.045
credit-g	0.368	0.137	0.074	0.104	0.070	0.075	0.070
diabetes	0.401	0.268	0.065	0.079	0.064	0.068	0.069
heart-statlog	0.237	0.365	0.056	0.100	0.060	0.082	0.056
hepatitis	0.154	0.204	0.021	0.032	0.021	0.052	0.021
ionosphere	0.056	0.087	0.025	0.039	0.018	0.027	0.018
kr-vs-kp	0.007	0.034	0.006	0.006	0.008	0.034	0.034
labor	0.072	0.135	0.035	0.019	0.019	0.026	0.035
sick	0.030	0.014	0.011	0.017	0.016	0.016	0.010
sonar	0.260	0.345	0.047	0.126	0.059	0.097	0.047
vote	0.039	0.029	0.025	0.031	0.027	0.030	0.027

Table 2. (*Continued*)

Cost Ratio = 50							
Data set	C4.5(P)	CR	C4.5cs	C4.5cs-mc	MetaCost	CR-2	CRPrune
breast-cancer	1.062	0.895	0.071	0.094	0.071	0.136	0.071
breast-w	0.149	0.125	0.031	0.053	0.031	0.030	0.030
colic	0.538	0.632	0.063	0.121	0.063	0.133	0.063
credit-a	0.432	0.646	0.044	0.067	0.044	0.050	0.044
credit-g	0.906	0.255	0.070	0.159	0.070	0.082	0.070
diabetes	0.993	0.635	0.065	0.106	0.069	0.080	0.065
heart-statlog	0.579	0.882	0.056	0.172	0.063	0.121	0.056
hepatitis	0.367	0.490	0.021	0.050	0.021	0.104	0.021
ionosphere	0.126	0.206	0.036	0.080	0.019	0.044	0.018
kr-vs-kp	0.017	0.034	0.010	0.011	0.011	0.034	0.034
labor	0.161	0.325	0.035	0.030	0.019	0.047	0.035
sick	0.075	0.033	0.014	0.039	0.036	0.038	0.011
sonar	0.625	0.831	0.047	0.248	0.047	0.177	0.047
vote	0.095	0.073	0.057	0.107	0.083	0.088	0.080

Table 3. Summary of the t-test on average misclassification cost

C R		C4.5(P)	CR	C4.5cs	C4.5cs-mc	MetaCost
2	CR-2	5/5/4	7/4/3	4/3/7	5/4/5	2/6/6
	CRPrune	6/4/4	9/4/1	7/0/7	5/4/5	5/3/6
5	CR-2	10/3/1	9/5/0	4/2/8	6/5/3	4/5/5
	CRPrune	12/1/1	12/2/0	3/6/5	8/3/3	7/3/4
10	CR-2	13/0/1	11/2/1	4/3/7	8/5/1	4/5/5
	CRPrune	12/1/1	11/3/0	4/7/3	10/2/2	7/4/3
20	CR-2	13/0/1	11/2/1	0/5/9	8/4/2	1/3/10
	CRPrune	13/0/1	13/1/0	3/8/3	11/1/2	3/6/5
50	CR-2	12/1/1	11/1/2	0/3/11	8/3/3	0/3/11
	CRPrune	12/1/1	12/2/0	2/10/2	12/1/1	2/10/2

Table 4 lists the average model training time (on a PC with Intel P4 3.0G Hz CPU and 512M memory), and Table 5 lists the corresponding summary with the t-test. As the cost ratio does not affect the model training time, we take only one cost ratio (cost ratio = 10) for the result.

Table 4. Average model training time on UCI data sets

Data set	C4.5(P)	CR	C4.5cs	C4.5cs-mc	MetaCost	CR-2	CRPrune
breast-cancer	0.004	0.005	0.004	0.005	0.046	0.003	0.002
breast-w	0.004	0.011	0.004	0.005	0.056	0.005	0.006
colic	0.011	0.013	0.011	0.010	0.095	0.004	0.005
credit-a	0.017	0.027	0.015	0.013	0.146	0.007	0.004
credit-g	0.043	0.017	0.043	0.038	0.378	0.008	0.006
diabetes	0.013	0.017	0.016	0.013	0.133	0.004	0.005
heart-statlog	0.006	0.008	0.007	0.005	0.062	0.002	0.001

Table 4. *(Continued)*

hepatitis	0.003	0.003	0.003	0.003	0.032	0.004	0.001
ionosphere	0.009	0.016	0.013	0.010	0.104	0.007	0.009
kr-vs-kp	0.117	0.065	0.095	0.120	1.260	0.070	0.065
labor	0.002	0.001	0.001	0.001	0.013	0.001	0.000
sick	0.057	0.105	0.066	0.061	0.630	0.075	0.096
sonar	0.011	0.025	0.012	0.012	0.129	0.009	0.003
vote	0.005	0.005	0.005	0.005	0.057	0.003	0.004

Table 5. Summary of the t-test on model training time

	C4.5(P)	CR	C4.5cs	C4.5cs-mc	MetaCost
CR-2	8/5/1	9/5/0	8/6/0	8/5/1	14/0/0
CRPrune	10/3/1	11/3/0	10/3/1	10/3/1	14/0/0

From the experiment results, several interesting observations can be made:

First, when the cost ratio is rather low (cost ratio = 2), CR-2 and CRPrune perform better than CR: the $w/t/l$ value on the average misclassification cost is 7/4/3 between CR-2 and CR, and 9/4/1 between CRPrune and CR. However, they do not outperform other algorithms. In fact, no algorithm wins all the time on all data sets; even cost-based algorithms do not always perform significantly better than error-based algorithms. The low cost ratio makes the task similar to error-based classification, where cost-sensitive approaches have no particular advantage.

Second, when the cost ratio is high (cost ratio ≥ 5), the proposed algorithms significantly outperform C4.5, CR and even C4.5cs-mc. In addition, CRPrune are comparable to C4.5cs and MetaCost, while CR-2 performs worse than them.

Third, the proposed new algorithms CR-2 and CRPrune are the definite winner on the model training time compared with all other algorithms. Simplicity and efficiency are certainly significant advantages of the proposed algorithms.

4 Conclusions and Future Work

This paper explores two simple and efficient pre-pruning strategies for the cost-sensitive decision tree algorithm to avoid overfitting. One is to limit the cost-sensitive decision trees to a depth of two. The other is to prune the trees with a pre-specified threshold. Empirical study shows that, compared to the error-based tree algorithm C4.5 and several other cost-sensitive tree algorithms, our cost-sensitive decision trees with pre-pruning are more efficient and perform well on most UCI data sets.

In the future, we plan to incorporate other pruning methods in our algorithms. In addition, it is also valuable to extend our pre-pruning cost-sensitive trees to include other types of cost.

References

1. Margineantu, D.: Methods for Cost-Sensitive Learning. PhD thesis, Oregon State University
2. Kukar, M., Kononenko, I.: Cost-sensitive learning with neural networks. In: Proceedings of the 13th European Conference on Artificial Intelligence. (1998) 445–449
3. Wan, C., Wang, L., Ting, K.: Introducing cost-sensitive neural networks. In: Proceedings of the 2nd International Conference on Information, Communications and Signal Processing, Singapore (1999) 445–449
4. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering **18**(1) (2006) 63–77
5. Turney, P.: Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. Journal of Artificial Intelligence Research **2** (1995) 369–409
6. Kwedlo, W., Kretowski, M.: An evolutionary algorithm for cost-sensitive decision rule learning. In: Proceedings of the 12th European Conference on Machine Learning. (2001) 288–299
7. Peng, Y., Huang, Q., Jiang, P., Jiang, J.: Cost-sensitive ensemble of support vector machines for effective detectionof microcalcification in breast cancer diagnosis. Lecture Notes in Computer Science **3614** (2005) 483–493
8. Ting, K., Zheng, Z.: Boosting trees for cost-sensitive classifications. In: Proceedings of the 10th Eruopean Conference on Machine Learning, Germany (1998) 191–195
9. Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining, ACM Press (1999) 155–164
10. Fan, W., Stolfo, S., Zhang, J., Chan, P.: Adacost: Misclassification cost-sensitive boosting. In: Proceedings of the 16th International Conference on Machine Learning. (1999) 97–105
11. Drummond, C., Holte, R.: Exploiting the cost (in) sensitivity of decision tree splitting criteria. In: Proceedings of the 17th International Conference on Machine Learning. (2000) 239–246
12. Ferri, C., Flach, P., Orallo, J.: Learning decision trees using the area under the roc curve. In: Proceedings of the 19th International Conference on Machine Learning. (2002) 139–146
13. Ling, C., Yang, Q., Wang, J., Zhang, S.: Decision trees with minimal costs. In: Proceedings of the 21st International Conference on Machine Learning. (2004)
14. Bradley, A., Lovell, B.: Cost-sensitive decision tree pruning: Use of the roc curve. In: Proceedings of the 18th Australian Joint Conference on Artificial Intelligence, Australia (1995) 1–8
15. Bradford, J., Kunz, C., Kohavi, R., Brunk, C., Brodley, C.: Pruning decision trees with misclassification costs. In: Proceedings of the European Conference on Machine Learning. (1998) 131–136
16. Turney, P.: Types of cost in inductive concept learning. In: Proceedings of the Workshop on Cost-Sensitive Learning at the 17th International Conference on Machine Learning. (2000)
17. Ling, C., Sheng, V., Bruckhaus, T., Madhavji, N.: Maximum profit mining and its application in software development. In: Proceedings of The 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06) (Industrial Applications Track). (2006) 929–934

18. Holte, R.: Very simple classification rules perform well on most commonly used datasets. Machine Learning **11** (1993) 63–91
19. Auer, P., Holte, R., Maass, W.: Theory and applications of agnostic pac-learning with small decision trees. In: Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann (1995) 21–29
20. Ting, K.: Inducing cost-sensitive trees via instance weighting. In: Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery, Springer-Verlag (1998) 23–26
21. Witten, I., Frank, E., eds.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005)
22. Blake, C., Keogh, E., Merz, C.: Uci repository of machine learning databases. Department of Information and Computer Science, University of California (1998) http://www.ics.uci.edu/˜mlearn/MLRepository.html.