An Analysis of Decision Tree Classifier

2011010539, Chen Liu, 刘晨

1.Introduction

This experiment is aimed to implement a decision tree classifier on the basis of several thousands of person's information such as education, workplace, working hours per week, etc. What we need to do is to build a decision tree to decide whether or not a person's income is over 5000 dollars using background information stated above.

The experiments includes two parts, non-pruning classifier and pruning classifier. The later one implements the algorithm of post-pruning to avoid over-fitting phenomenon.

2.Design

There are several barriers we need to go over to implement the whole algorithm. Some are listed below:

1. Discretize continuous property

Some properties such as working hours are continuous rather than discrete, So we need to turn them into a discrete properties. I come up with some ideas when confronted with this problem. I find that these properties are all presented as a non-negative integer, so what we need to is to divide the range [0,infinite) into several sub-ranges. Considering the size of the data, I choose to divide the whole range into 4 sub-ranges.

The values of a certain property of all the people is regarded as a set, we should first sort the elements in order and then divide them into 4 subsets. The easiest way is to divide into 4 subset consistent in value and same of size. The problem arises while the most values of working hours are 0 and result will be [0,0),[0,0),[0,0),[0,infinite), which is ridiculous. So the symmetric division is unreasonable. Another method is based on the presence of values excluding repeat. This method is simple and works well in reality. We can also use the concept of entropy to complete this task, in which we need to take 2 steps. We choose a value to divide the whole range into 2 parts in order to minimize the entropy, then we divide each subsets into 2 parts using the same algorithm. This method seems more scientific and perform best in all.

2. Dealing with the absence of some information

After discretion, each property of a person can be represented by a integer. However there are some special conditions, some properties of some people is unknown to us, marked by '?', we use -1 to represent them. If the key property (income in this experiment) is lost, we consider this person's information is invalid.For other properties, in training and testing period, we replace -1

with any valid value randomly.

3. Pruning strategies

There are many pruning methods. In this experiment, we use post-pruning on the base of accuracy. When testing in validation set, if the accuracy of a node in the decision tree is higher when we simply assign the predicted value to 1 or -1, compared to the property classifier, we cut the branch below this node. This method guarantees better performance in the validation set.

3.Implementation

We use Java language to implement this algorithm. The Java class in the whole project includes the decision tree,its node, the property, the people and the training and testing process. Under the root folder of the project, training data, testing data and decision tree are in the sub-folder ‘doc’.

4.Result

1． Task 1:

If training data is 5% records randomly taken from the training file, The average accuracy is 98% when tested on the training data and 80.5% when tested on the data from testing file.For 50% records, the two numbers change to 94.6% and 82.6%. For 100% records, the two number change to 93.3% and 83.0%.

I find that the dataset is asymmetric, so I also use recall rate to revaluate the performance of the algorithm in another aspect. For 100% records, the recall rate of positive type and negative type is respectively 64.5% and 87.8%.

2． Task 2:

The average size of decision tree is 2795 for 5% records after pruning, compared to 3006 before pruning. For 50% records, the two numbers are 28149 and 34559. For 95% records, the two numbers are 50251 and 65269.

3． Full Result

Column 2-4 are data before pruning and column 5-7 are data after pruning. (using symmetric ways excluding repeat to discretize continuous property)

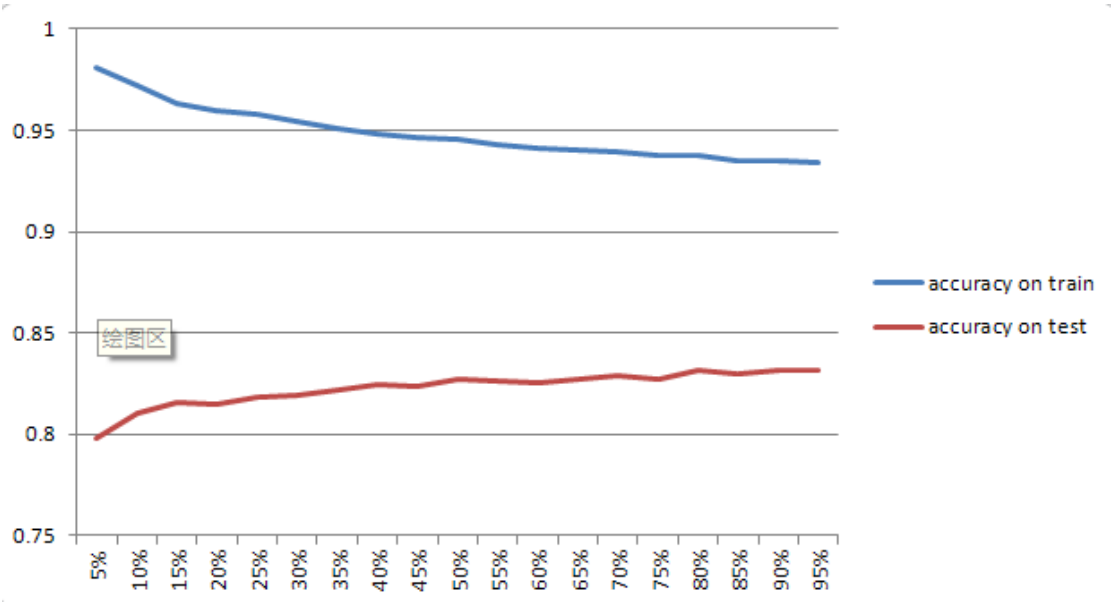
training size	tree size	accuracy on train	accuracy on test	tree size	accuracy on train	accuracy on test
5%	3015	0.98065463	0.797657618	2919	0.953616446	0.754157929
10%	6661	0.972313761	0.810476520	6081	0.945722789	0.777560055
15%	11123	0.963589395	0.815315027	9644	0.929732562	0.787593664
20%	13730	0.959285239	0.814686753	11787	0.928689037	0.796723858
25%	17499	0.957828187	0.818387006	14839	0.929625362	0.806114829

30%	21333	0.954357609	0.819594835	18127	0.924815983	0.806823781
35%	25761	0.950838887	0.822189228	21318	0.921951312	0.812238229
40%	28797	0.948520679	0.824096031	23701	0.921807282	0.814555500
45%	32062	0.946805545	0.823847191	26210	0.922036854	0.815730248
50%	35129	0.945930151	0.827560784	28384	0.920705421	0.817899631
55%	38944	0.943085766	0.826352160	31307	0.920635363	0.821625208
60%	40766	0.941296418	0.825646683	32128	0.91787808	0.819405477
65%	45662	0.939848099	0.826880411	35867	0.918548513	0.82264637
70%	48152	0.938966552	0.829007423	37204	0.918855253	0.824929378
75%	50184	0.937406884	0.827026305	38756	0.918375338	0.826124391
80%	54784	0.937204396	0.831480018	41639	0.919455353	0.827849892
85%	57577	0.934973699	0.829484947	44494	0.91850977	0.826350828
90%	60420	0.934714633	0.831490715	46033	0.917490892	0.827987078
95%	63665	0.934117768	0.831180709	47777	0.919190307	0.829897483

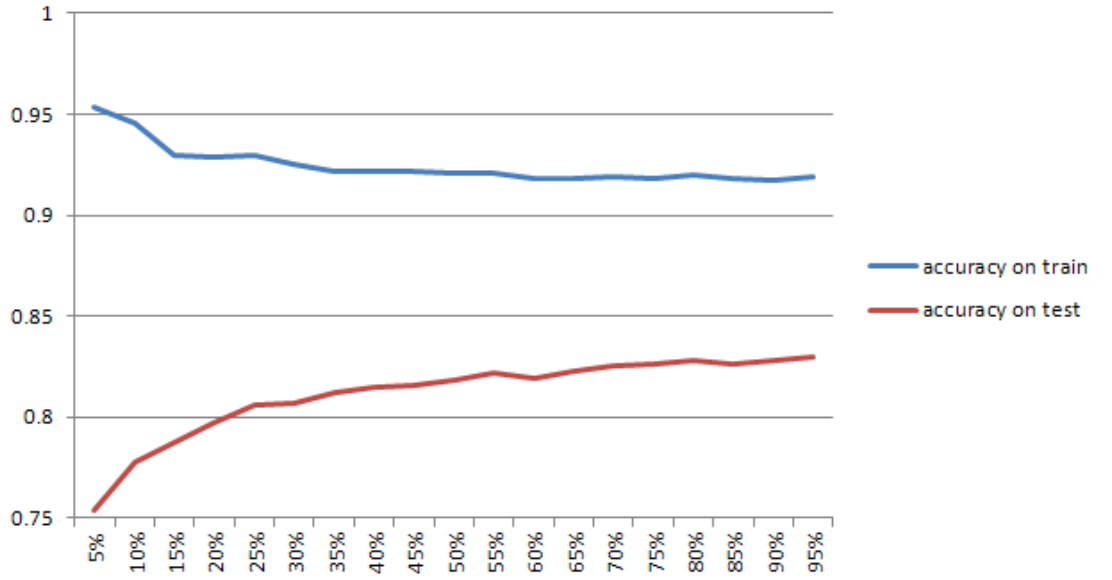
Column 2-4 are data before pruning and column 5-7 are data after pruning. (using entropy to discretize continuous property)

training size	tree size	accuracy on train	accuracy on test	tree size	accuracy on train	accuracy on test
5%	7269	0.905177	0.806544562	7023	0.884874	0.752640
10%	10983	0.899985	0.815282022	10103	0.878232	0.776676
15%	16537	0.891053	0.810016031	14944	0.870835	0.779776
20%	20467	0.885250	0.819418669	18423	0.867287	0.794330
25%	24226	0.882710	0.815894429	21444	0.866965	0.793431
30%	26069	0.879204	0.819463217	22835	0.862895	0.801287
35%	29961	0.878391	0.817188225	26332	0.86347	0.803272
40%	31879	0.876091	0.821265961	27746	0.863876	0.808890
45%	34570	0.874747	0.819519977	29488	0.862206	0.808066
50%	37784	0.872182	0.821590896	32117	0.860099	0.811676
55%	42302	0.870735	0.821567375	35345	0.859987	0.813664
60%	43066	0.869507	0.822920173	36309	0.858317	0.812874
65%	45481	0.869136	0.823075944	38434	0.859663	0.815941
70%	46580	0.868725	0.824831474	38912	0.858912	0.815428
75%	50234	0.868921	0.823255971	41818	0.859583	0.816741
80%	50875	0.866880	0.823800587	42067	0.858824	0.818164
85%	54555	0.866095	0.824792279	45210	0.857795	0.817997
90%	55985	0.865306	0.823871437	46047	0.857763	0.819608
95%	59006	0.865202	0.824536352	47871	0.857744	0.820016

The tendency of the accuracy on training set and testing set before pruning.



The tendency of the accuracy on the training set and testing set after pruning.



5.Analysis & Conclusion

From the tables and charts above, we can conclude that with the training set becoming larger and larger, the accuracy on the training set is decreasing while on the testing set is increasing.That is because with a larger training data, the decision tree will have more branches, so accuracy on the testing set is increasing. However, on the training set, larger training set means more conflicts, which are the cases that two persons are same in all properties except the output property(income in this experiment), the decision tree will unavoidably make more and more wrong decisions, so the accuracy on the training test is decreasing.

On the other hand, post-pruning on the basis of accuracy will increase the accuracy on the validation set. The numbers above also shown that the post-pruning will play a more important

role with the decision tree growing bigger. For 5% record, the post-pruning will cut about 3% branches, but for 100% record, this proportion grows to over 20%, that's because conflicts will make more deep branches in the tree unnecessary. Although on the testing set, the decision tree perform poorer after the pruning, the reduction of tree size is worth the little expense of performance.

By comparing two methods to discretize the continuous properties, we can easily find that the method based on entropy generate a decision tree of fewer nodes. Although poor performance on the training set but almost the same performance as other methods on the testing set.

I gain a better understanding of the decision tree algorithm and over-fitting via this experiment, the substance of the over-fitting phenomenon may be more conflicts with the larger training set.In deep branches of the tree, the size of corresponding data is small and convey little information, there is no need to build more deeper branches. In addition, without pruning strategies, deeper branches is more sensitive to noise data, which results in high accuracy in training set but low in testing set.