Combining Pruned Tree Classifiers with Feature Selection Strategies to Improvise Classification Accuracy

Shweta Rajput, Sapna Saxena

Abstract—Feature subset selection minimize the attribute space of a feature set by selecting subset of relevant, non redundant and most contributing features. In this paper we present a comparative study of attribute space selection techniques for the C4.5 algorithm which highlights some efficiency improvements. As the dimensionality of the data increases, many types of data analysis and classification problems become significantly harder .So only worthy attributes were selected by picking features which maximize predictive accuracy. As a preprocessing step we have used feature selection methods such as Information gain, Gain ratio, ReliefF and OneR. It lessen overfitting of learning classifier c4.5 and increase the computation speed of prediction. The pruned c4.5 algorithm implemented using feature selection outperformed the unpruned c4.5 in terms of predictive accuracy. The experimental results show that filter methods, Gain Ratio, ReliefF, Information Gain, and oneR allow the classifiers to achieve the highest increase in classification accuracy.

Index Terms— Weka, C4.5, spam classification, pruning, feature subset selection, dimensionality reduction

1 Introduction

CLLASIFICATION of spam emails was done using decision tree classification techniques. The most causative features are selected to achieve improvisation in predictive accuracy and to reduce time complexity. Feature ranking techniques are applied to a training data. After the feature selection subset with the top merit is used to reduce the dimensionality of both the original training data and the testing data. Both reduced datasets were then passed to a tree classifier for training and testing. Filter methods were applied to shunt best features attribute set, results are then obtained by using c4.5 pruned and c4.5 unpruned classification technique. The degree of improvement in case of feature selection will depend on many factors; the type of classifier, the effectiveness of the feature selection and the worth of the features. In the case of c4.5 classifier, the feature selection deletes noisy features and reduces the feature-space dimension. In spambase e-mail filtering, feature space contains various properties of e-mail messages. Any feature is considered high-quality if features were very much correlated with class, yet uncorrelated with each other. Aim was to identify a representative set of features and screens irrelevant, redundant and noisy features such that their relevance does not strongly depend on other features. A decision tree is a tree data structure consisting of decision nodes and leaves. A leaf specifies a class value. A decision node specifies a test over one of the attributes, which is called the attribute selected at the node. For each possible outcome of the test, a child node is present. The algorithm constructs a decision tree starting from a training set T S, which is a set of cases, or tuples in the database terminology. The class specified at the leaf is the class predicted by the decision tree. Aim is to identify a representative set of features. Feature selection degraded machine learning performance in cases where some features were eliminated which were highly predictive of very small areas of the instance space. It focuses on those aspects of the data most useful for analysis and future prediction.

2 Feature Ranking and Subset Selection Techniques

In this work we consider four approaches to feature selection for the attribute selection. These algorithm use rankers method on features and evaluate the feature by ranking them from most important to least important.

2.1 Information Gain

It is a measure to evaluates the worth or relevance of an attribute with respect to the class based on the concept of entropy. It is expected decrease in entropy caused by partitioning the examples according to a given attribute A. It is the amount of information gained about Y after observing X and vice versa[3]. The expected value of information gain is the mutual information of target variable (X) and independent variable (A). It is the reduction in entropy of target variable (X) achieved by learning the state of independent variable (A).Consider an attribute X and class attribute Y, the information gain of a given attribute X with respect to class attribute Y is the reduction in uncertainty about the value of Y when the value of X is known. The value of Y is measured by its entropy, H(Y). The uncertainty about Y, given the value of X is given by the conditional probability of Y given X, H (Y|X).

$$I(Y, X) = H(Y) - H(Y/X)$$

where Y and X are discrete variables that take values in $\{y_1.....y_k\}$ and {x1....xl}

The entropy (or measure of the impurity in a collection of items) of Y is given by:

$$H(Y) = -\sum_{i=1}^{i=k} P(Y = yi)\log2(P(Y = yi))$$

The conditional entropy of Y given X is

$$H(Y|X) = -\sum_{j=1}^{l} P(X = x_j) H(Y|X = x_j)$$

Alternatively the information gain is given by: I(Y, X) = H(Y) +H(X)-H(X, Y)
Where H(X, Y) is the joint entropy of X and Y:

$$H(X, Y) = -\sum_{i=1}^{k} \sum_{j=1}^{l} P(X = x_j, Y = y_i) \log2 P(X = x_j, Y = y_i)$$

High Entropy means X is from a uniform (non worthy) distribution and low Entropy means X is from varied distribution (useful).The major drawback of using information gain is that it tends to favours attributes with large numbers of distinct values over attributes with fewer values.

2.2 Information Gain

It is an extension to the information gain. Gain ratio overcomes problem of information gain biasing by introducing an extra term taking into account how the feature splits the data. The attribute with the maximum gain ratio is selected as the splitting attribute[3]. The split information value represents the potential information generated by splitting the training data set D into v partitions corresponding to v outcomes on attribute A

$$\text{SplitInfo}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * \log2\left(\frac{|D_j|}{|D|}\right)$$

For each possible outcome, it considers the number of tuples having that outcome with respect to the total number of tuples in D. The gain ratio is calculated as:

Gain Ratio (A) = Gain (A) / SplitInfo(A)

2.3 ReliefF

The general idea of this method is to choose the features that can be most distinguished between classes. At each step of an iterative process, an instance x is chosen at random from the dataset and the weight for each feature is updated according to the distance of x to its Near miss and Near Hit[2]. The process of ranking the features in relief follows three basic steps:
1. Calculate the nearest miss and nearest hit.
2. Calculate the weight of a feature.
3. Return a ranked list of features or the top k features according to a given threshold.
The basic idea is to draw instances at random, compute their nearest neighbors, and adjust a feature weighing vector to give more weight to features that discriminate the instance from neighbors of different classes. Relief is feature weighting algorithm work by approximating the following difference of probabilities for the weight of a feature X.

W_x =P(different values of X|nearest instance of different class)-P(different value of X|nearest instance of same class)

BY removing the context sensitivity provided by the “nearest instance” condition, attributes are treated as independent of one another:

Relief_X = P (different values of X|different class)-P (different value of X|same class)

2.4 OneR Attribute Evaluation

It is a rule based algorithm to generate compact, easy-to-interpret rules by concentrating on a specific class at a time. A classification rule can be defined as r = (a, c) where a is a pre-condition which performs a series of tests that can be evaluated as true or false and c is a class that apply to instances covered by rule r. Aim is to find a general rule that predicts the class given the values of attribute[2]. Rule base algorithms work on a specific class at a time and follows three steps: Generate rule R on training data S, remove the training data covered by rule and repeat the process.OneR is the simplest approach to finding a classification rule as it generates one level decision tree. OneR constructs rules and tests a single attribute at a time and branch for every value of that attribute. For every branch, the class with the best classification is the one occurring most often in the training data.

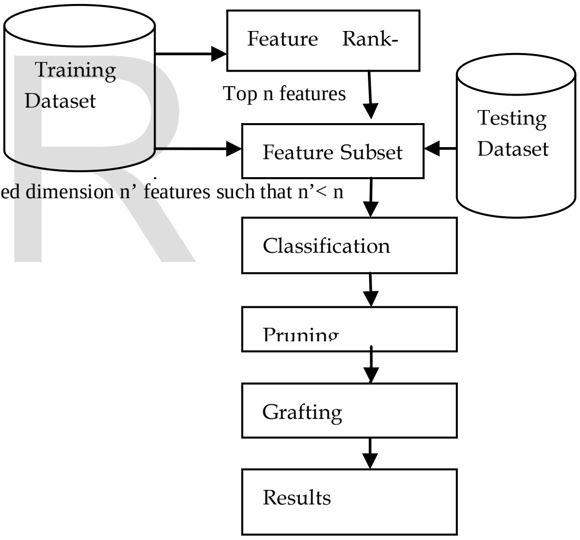

Figure 1.2 Framework of proposed model

3 Classification Method and Pruning Techniques

3.1 C4.5 Classifier

C4.5 [1] is an evolution and refinement of ID3 algorithm developed by J. Ross Quinlan. The spambase data set was tested using the J48 algorithm in WEKA and then after the result is visualized for decision tree. It generates non binary tree and uses measure called gain ratio to construct decision tree, the attribute with highest normalized gain ratio is taken as the root node and the dataset is split based on the root element values. Again the information gain is calculated for all the sub-nodes individually and the process is repeated until the prediction is completed. Error–based pruning is performed after the growing phase. J48 can handle both continuous and discrete attributes, training data with missing attribute values and attributes with differing costs and provide an option to prune trees after creation. There are a number of parameters related to tree pruning in the J48 algorithm and should be used with care as they can make a noteworthy difference in the quality of results. J48 employs two post pruning methods, namely subtree replacement and subtree raising.

Though the decision tree generated by C4.5 was accurate and efficient, but they result in bulky trees leading to problem of overfitting. Overfitting decision trees are more complex than necessary, so pruning is required to obtain small and accurate models, avoids unnecessary complexity and helps in optimizing the classification accuracy. There are two strategies for pruning, pre-pruning and post-pruning. Post-pruning is preferred in practice as pre pruning can stop early.

3.2 Post-Pruning

Take a fully-grown decision tree and discard unreliable parts in a bottom-up fashion is known as post-pruning. To decide whether to do post pruning or not, calculate error rate before and after the pruning. If generalization error improves after trimming, replace sub-tree by a leaf node. Class tag of leaf node is calculated from majority class of instances in the subtree.There are couple of methods for post pruning like reduced error pruning, Error complexity pruning etc, covered in later sections. J48 employs two pruning methods. The first is subtree replacement, nodes Replacement is performed if the error estimate for the prospective leaf is no greater than the sum of the error estimates for the current leaf nodes of the subtree.This process starts from the leaves of the fully formed tree, and works backwards toward the root node. The second type of pruning used in J48 is subtree raising, it replaces a subtree with its most populated majority branch if this does not increase the estimated error. In the Weka J48 classifier, lowering the confidence factor decreases the amount of postpruning. Lowering confidence factor filter irrelevant nodes.Subtree raising is replacing a tree with one of its subtrees.Subtree replacement consists of replacing the subtree rooted at the father of the node by the subtree rooted at the node, tree is considered for replacement once all its subtrees have been considered.

3.2.1 Reduced error pruning

It is a post-pruning method used by C4.5 algorithm that divides dataset into three parts, namely training set, testing set and hold-out set (Pruning set). It uses a hold-out set (a fraction of the training data) for making pruning decisions and to estimate generalization error. So less data is used to determine the whole structure of the tree as compare to other pruning methods. However, once the structure has been summarized, the full training data can be back-fitted against the structure in order to find the node and leaf class distributions[5]. Node is then replaced with its majority classification. If the performance of the modified tree is just as good or better on the validation set as the current tree then set the current tree equal to the modified tree.Subtree is then replaced by leaf node, means pruning is done. Nodes are removed only if the resulting tree performs no worse on the validation set. Nodes are pruned iteratively, at each iteration the node whose removal most increases accuracy on the validation set is pruned. Pruning stops when no pruning increases accuracy. The problem with this approach is that it potentially "wastes" training data on the validation set, reducing the amount of data available for training. If test set is smaller than training set, it may lead to overpruning but it has the advantage of simplicity and speed. The popular C4.5 algorithm adds to the reduced error based pruning method the subtree replacement operator. Numfold parameter is used to achieve reduced error pruning.

3.3 Pre-pruning

Generating a smaller and simpler tree with fewer branches and while building keep on checking whether tree is overfitted or not is known as pre-pruning. It stop growing a branch when information becomes unreliable and is based on statistical significance test.Pre pruning techniques include minimum no of object pruning and chi square pruning. The minObj parameter means minimum no of object per branch is available in Weka in J48.This option tells c4.5 limit the minimum number of examples each leaf could have and not to create a new tree branch unless that branch contains greater than or equal to specified number of instances[4]. This prevents overfitting of data when number of instances remaining to be classified is small. This parameter is varied in J48 algorithm to test predictive accuracy. If split result in child leaf that denotes less than minobj from dataset, parent node and children node are compressed to a single node.

4 DATASET DESCRIPTION AND MACHINE LEARNING TOOL

4.1 Spambase Dataset

Spambase dataset contains 4601 instances and 58 attributes, taken from UCI machine learning repository. It consists of 1 – 57 continuous attributes and1 nominal class label. Most of the attributes indicate whether a particular word or character was frequently occurring in the e-mail.Attributes 1 to 48 word_freq_WORD percentage of words in the e-mail that match WORD.Then 49 to 54 char_freq_CHAR percentage of characters in the e-mail that match CHAR. 55th attribute is capital_run_leng th_average average length of uninterrupted sequences of capital letters.56th is capital_run_leng th_longest length of longest uninterrupted sequence of capital letters

- *Author name is currently pursuing masters degree program in electric power engineering in University, Country, PH-01123456789. E-mail: author_name@mail.com*
- *Co-Author name is currently pursuing masters degree program in electric power engineering in University, Country, PH-01123456789. E-mail: author_name@mail.com*
  (This information is optional; change it according to your need.)

.57th is capital_run_leng th_total sum of length of uninterrupted sequence of capital letters.58 Class attribute tells e-mail is spam (1) or not (0) .48 continuous real attributes of type word_freq_WORD, i.e. 100 * (number of times the WORD appears in the e-mail) /total number of words in e-mail. A "word" in this case is any string of alphanumeric characters bounded by non alphanumeric characters or end-of string. Six continuous real attributes of type char_freq_CHAR denotes percentage of characters in the e-mail that match CHAR, i.e. 100 * (number of CHAR occurrences) / total characters in e-mail.The runlength attributes (55-57) measure the length of sequences of consecutive capital letters. The last column of 'spambase.arff' denotes whether the e-mail was considered spam (1) or not (0).Class Distribution is 1813 spam and 2788 ham.

4.2 WEKA Data Mining Tool

The email spam classification has been implemented in Weka. Feature ranking and feature selection is done by using the methods such as Information gain, Gain ratio, Relief, OneR as a preprocessing step so as to select feature subset for building the learning spam model. Classification algorithm c4.5 decision tree is an effective tool in prediction. First we saved the trained c4.5 classifier. Then the saved model is load with right click menu on result list panel using weka option 'load model'. In test option, select 'supplied test' option to load test data file, then 'no class' option is selected from list of attributes, choose 'plain text' option from output prediction and then apply model on test data using 're-evaluate model on current test set' option in weka.The option 'unpruned' is set true for c4.5 to generate unpruned tree else it will generate pruned tree. Various available options for c4.5 in weka is

- binarySplits: It split numeric attribute into 2 ranges using an inequality.
- confidenceFactor: smaller values(means we have less confidence in training data)will lead to more pruning as it filter out irrelevant nodes.
- minNumObj: tells minimum number of instances per leaf node
- numFolds: calculates the amount of data for reduced error pruning – one fold used for pruning, the rest for growing the tree.
- reducedErrorPruning: parameter tells whether to apply reduced error pruning or not.
- subtreeRaising: whether to use subtree raising during pruning.
- Unpruned: whether pruning takes place at all. Value is change to "True" to build a pruned tree in c4.5.
- useLaplace: whether to use Laplace smoothing at leaf node.

5 Results

Work was done to reduce feature space. As a part of our implementation, we have divided the dataset into two parts, training data used to generate the predictive spam model, and the other part is used as test data to test the accurateness of model. Spam model is trained using 10-fold cross validation.Spambase dataset is used for training purpose as well as for the testing purpose. After preprocessing step top features are considered while building training model and testing because there is a significant performance improvement. Prediction accuracy, correctly classified instances, incorrectly classified instances, kappa statistic, mean absolute error, root mean square error, time taken to build model, number of leaves and size of tree are used as performance measures of the system. More than 97% prediction accuracy is achieved by C4.5 for testing data with all the four feature selection methods in consideration; whereas highest 98.3699% prediction accuracy is achieved by infogainAttribute filter method for unpruned c4.5 tree classifier. Training and testing results are given in Tabular data. Both for training and testing data performance results were improved when filters were applied on dataset.Noteworthy change was noticed that filter methods for unpruned c4.5in case of testing data outperforms as compare to pruned c4.5 for spambase dataset. High hike was observed in predictive accuracy of unpruned c4.5 classifier when the filter methods were applied on testing data.

Table 1
GainRatioAttributeEval using Ranker Search Method for training data

Parameters	Pruned J48 Classifier	Unpruned J48 Classifier
Total Instances	4601	4601
Correctly classified	4280	4257
Incorrectly classified	321	344
Kappa statistic	0.8538	0.8435
Mean absolute error	0.0898	0.0844
Root mean square error	0.2558	0.2625
Time taken to build model	4.87 sec	5.13 sec
Number of leaves	115	178
Accuracy	93.02	92.52
Size of tree	229	355

Table 2
InfoGainAttributeEval using Ranker Search Method for training data

Parameters	Pruned J48 Classifier	Unpruned J48 Classifier
Total Instances	4601	4601
Correctly classified	4280	4259
Incorrectly classified	321	342
Kappa statistic	0.8538	0.8445
Mean absolute error	0.0898	0.0842
Root mean square error	0.2558	0.2625
Time taken to build model	4.13 sec	3.95 sec
Number of leaves	115	178
Accuracy	93.02	92.5668
Size of tree	229	355

Table 3
OneRattributeEval using ranker search method for training data

Parameters	Pruned J48 Classifier	Unpruned J48 Classifier
Total Instances	4601	4601
Correctly classified	4278	4252
Incorrectly classified	323	349
Kappa statistic	0.8529	0.8412
Mean absolute error	0.0901	0.085
Root mean square error	0.2568	0.2643
Time taken to build model	4.21 sec	4.73 sec
Number of leaves	111	182
Accuracy	92.97	92.41
Size of tree	221	363

Table 4
ReliefFAttributeEval using ranker search method for training data

Parameters	Pruned J48 Classifier	Unpruned J48 Classifier
Total Instances	4601	4601
Correctly classified	4279	4261
Incorrectly classified	322	340
Kappa statistic	0.8533	0.8453
Mean absolute error	0.0897	0.0838
Root mean square error	0.2567	0.2623
Time taken to build model	5.05 sec	4.74 sec
Number of leaves	114	186
Accuracy	93	92.61
Size of tree	227	371

Table 5

GainRatioAttributeEval for testing data calculated using option “re-evaluate model on current test set” in Weka

Parameters	Pruned J48 Classifier	Unpruned J48 Classifier
Total Instances	4601	4601
Correctly classified	4486	4521
Incorrectly classified	115	80
Kappa statistic	0.9475	0.9635
Mean absolute error	0.0459	0.0316
Root mean square error	0.1515	0.1256
Accuracy	97.5005	98.2612

Table 6
InfoGainAttributeEval using Ranker Search Method for testing data

Parameters	**Pruned J48 Classifier**	**Unpruned J48 Classifier**
Total Instances	4601	4601
Correctly classified	4477	4526
Incorrectly classified	124	75
Kappa statistic	0.9433	0.9658
Mean absolute error	0.0503	0.0296
Root mean square error	0.1587	0.146
Accuracy	97.304	98.3699

Table 7
OneRattributeEval using ranker search method for testing data

Parameters	**Pruned J48 Classifier**	**Unpruned J48 Classifier**
Total Instances	4601	4601
Correctly classified	4476	4522
Incorrectly classified	125	79
Kappa statistic	0.9429	0.964
Mean absolute error	0.0508	0.0313
Root mean square error	0.1594	0.125
Accuracy	97.2832	98.283

Table 8

ReliefFattributeEval using ranker search method for testing data

Parameters	**Pruned J48 Classifier**	**Unpruned J48 Classifier**
Total Instances	4601	4601
Correctly classified	4488	4524
Incorrectly classified	113	77
Kappa statistic	0.9485	0.9649
Mean absolute error	0.0456	0.0302
Root mean square error	0.151	0.1229
Accuracy	97.544	98.3265

5 CONCLUSION

In this paper, we have compared and evaluated the approaches based on the factors such as dataset used, feature ranked and selected. Feature selection helps to find subset of relevant dataset in features in a given model construction.Performance results were enhanced when filters were applied on dataset. The experiments were conducted on spambase dataset. The results demonstrated that filter methods for unpruned c4.5 brings a noteworthy change in case of testing data. Both in case of unpruned c4.5 and in pruned c4.5 information gain attribute filter outperforms for training data (93.02% accuracy) as well as for testing data(98.3699% accuracy).For spambase dataset, we acquired the best percentage accuracy of 98.3699% with c4.5 for testing data. It's concluded that the implemented FS can improve the accuracy of C4.5 classifier by performing feature selection.

REFERENCES

[1] J. R.. Quinlan, "C4.5: Programs for Machine Learning," M,organ Kaufmann Publishers Inc., 1993.

[2] R.Parimala,Dr. R. Nallaswamy, "A Study of Spam E-mail classification using Feature Selection package" , Global Journal of Computer Science and Technology, Vol. 11(7) May 2011.

[3] *T. M. Mitchell*, "Machine Learning", McGraw-Hill, 1997.

[4] SamDrazin and MattMontag, "Decision Tree Analysis using WEKA," Machine Learning-Project II,University of Miami.

[5] Shweta Rajput and Amit Arora, "Designing Spam Model- Classification Analysis using Decision Trees," International Journal of Computer Applications, vol. 75(10), Aug. 2013.