Role of Attribute Selection in Classification Algorithms

S. Dinakaran, Dr. P. Ranjit Jeba Thangaiah,

Abstract— Feature selection is the technique of removing irrelevant features and to reduce dimensionality of the feature. This paper proposed attribute selection of information gain attribute evaluator and ranker search method to selected attribute and each selected attribute is ranked based on the filter and wrapper method. Tree based J48 classifier is used with different test options namely 10 fold cross validation, use training set, supplied test set, and percentage split default of 66% are compared with all options and to generate best accuracy results; Labor dataset is implementing to test the above methods.

Index Terms— Feature selection, dimensional reduction, Information gain, Ranker , Decision tree, Cross Validaton, Attribute ranking,

1 INTRODUCTION

Feature selection is to reduce the dimensionality, remove irrelevant and redundant data the feature selection is used. Due to extensive growths in data storage and data attainment, data pre-processing techniques such as feature selection have become ever more popular in classification tasks [1]. Most datasets contain a large number of attributes due to large number the accuracy results may not much better, so to perform best result attribute selection is very essential. Data mining contains many techniques in that classification is one which classify different tasks, e.g. in a class the student categories are good, average and poor, here placement officer may guess whether the student is getting placed or not here predict definite tags are "good", "average", "poor" for student categories.. There are lots of decision tree algorithms the J48 is mostly used and proposed by Quinlan 1993. J48 is a re-implementation of C4.5 release 8 in Java. A portion of time has been accomplishing the similar results as the original C4.5. J48 implements both C4.5's confidence-based post- pruning and sub-tree rising [5]. Information gain (IG) measures the amount of information in bits about the class prediction, if the only information existing is the presence of a feature and the corresponding class distribution. Strongly, it measures the expected reduction in entropy [2]. Examples of filter techniques for feature selection which ranks individual features according to feature relevance score. The correlation-based feature selection (CFS) technique [3] scores and ranks subsets of features, rather than individual features. To partition the input space such that the training data classification has taken place in all partitions with minor in decision a training data set, decision trees use a node splitting conditions [4]. Test options are mostly what extent the test is to be taken and there introduced different options that shown as percentage split default of 66% shown best accuracy of all other testing options.

- *Dinakaran is currently pursuing Ph. D degree in the Department of Computer Applications in Karunya University, India, PH-9894559747. E-mail:dinakaran77@gmail.com*
- *Dr. P. Ranjit Jeba Thangaiah is currently an Assistant Professor(SG) in the Department of Computer Applications in Karunya University, India, E-mail: ranjit@karunya.edu*

2 METHODOLOGY

2.1 Feature Selection

The feature selection approach is divided into filters [6, 7], wrappers [9] and embedded approaches [8]. Filter method is independent of exact learning algorithms and it evaluates the stimulating of feature by the measurements of data content [10]. Embedded and wrapper approaches whose computational cost may be very expensive they seriously depend on exact learning algorithm [9].

2.2 Information Gain

Information gain is attributed evaluator used in feature selection when information gain chooses then default the ranker search method gets selected. Information gain is biased towards multivalued attributes, the attribute select measure information gain select the attribute with the highest information gain Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$ Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D) \quad [11]$$

2.3 Ranker Search Method

Ranker method is ranked attributes by their individual evaluations Use in conjunction with attribute evaluators (ReliefF, Gain Ratio, Entropy etc.) with the parameter generate ranking (true or false), number to select, and threshold values is set threshold by which attributes can be discarded. Default value results in no attributes are discarded. Use either this option or number to select to reduce the attribute set. The classification, variable ranking is a filter method: it is a preprocessing step, independent of the choice of the predictor [9]. The ranker method generally performs the rank which attributes should be obtain high or low rank according to the selected attribute in the given datasets. Ranker is providing a rating of the attributes, orderly by their score to the evaluator.

2.4 Tree Based J48

From the classification algorithm deals with classifier and finds a different algorithm namely bayes, functions, rules, trees etc., the J48 is selected from tree induction. The supervised methods used are Naïve Bayes classifier, J48 Decision Trees and Support Vector Machines [12]. The cost of making a J48 decision tree (WEKA implementation of the classic C4.5 decision tree) without sub tree resin is O (mnlogn) where m is the number of the attributes, and n is the number of training examples for the J48 decision tree algorithm [13]. J48: Java implementation of C4.5 algorithm. Based on the Hunt's algorithm, pruning takes place by replacing internal nodes with a leaf node. Top-down decision tree/voting algorithm [18].
BinarySplits -- Whether to use binary splits on nominal attributes when building the trees.
ConfidenceFactor -- The confidence factor used for pruning (smaller values incur more pruning).
Debug -- If set to true, classifier may output additional info to the console.
MinNumObj -- The minimum number of instances per leaf.
NumFolds -- Determines the amount of data used for reduced-error pruning. One fold is used for pruning, the rest for growing the tree.
Seed -- The seed used for randomizing the data when reduced-error pruning is used.
SubtreeRaising -- Whether to consider the subtree raising operation when pruning.
Unpruned -- Whether pruning is performed.
UseLaplace -- Whether counts at the leaves are smoothed based on Laplace.

2.5 Test Options

In every classification algorithm there have different test options namely use training set, supplied test set, cross validation, and percentage split.

1. Use training set: It was trained on how well it predicts the class of the instance to evaluate the classifier.
2. Supplied test set: Choose the file from the dialog box to test set, that the classifier is evaluated on how well it predicts the class of a set of instances loaded from a file.
3. Cross-validation: The classifier is evaluated by cross-validation, using the number of folds that are entered in the Folds text field.
4. Percentage split: Certain percentage of the data to classifier is evaluated on how well it predicts which is held out for testing. The amount of data thought out depends on the value entered in the percentage % textbox [14].

Further from the above options there have fewer of the options but here not using that more options

3 Experiments

3.1 Dataset

In this paper to examine the labor dataset from weka data folder has been taken, this dataset contains 17 attributes, 57 instances and type contain both 8 numeric attributes and 9 nominal attributes. For all attribute contain statistical values like maximum, minimum, mean, and standard deviation according to their ability. Also for each attribute it shows missing values with percentage, distinct value and unique value with a percentage. The weka tool is used for performing the results, after preprocessing the dataset; the classifier J48 decision tree is performed. According to the Stratified result summarizes the number of class instance is differ before and after the attribute selection and also to the test options.

TABLE 1:
DETAILED ACCURACY BY CLASS WEIGHTED AVERAGE (BEFORE SELECTING ATTRIBUTE)

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
10 Fold Cross Validation	0.737	0.28	0.748	0.737	0.74	0.695
Training & Test Set	0.877	0.089	0.896	0.877	0.88	0.918
Spilt percentage (66%)	0.895	0.138	0.895	0.895	0.895	0.814

The attribute is selected and ranking is performed for all input data, first chosen the information gain attribute evaluator then repeats it select the default ranker search method next with the support of an attribute selection mode, full training set chosen to attribute selection and ranking is performed and attributes are ranked accordingly is shown as that 16 out of 17 attribute is selected and ranked.
=== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 17 class):
Information Gain Ranking Filter

Ranked attributes:
0.2948 2 wage-increase-first-year

0.1893 3 wage-increase-second-year
0.1624 11 statutory-holidays
0.1341 14 contribution-to-dental-plan
0.1164 16 contribution-to-health-plan
0.1091 12 vacation
0.0855 13 longterm-disability-assistance
0.0717 9 shift-differential
0.0548 7 pension
0.0484 5 cost-of-living-adjustment
0.0333 15 bereavement-assistance
0.0307 4 wage-increase-third-year
0.024 10 education-allowance
0.0195 8 standby-pay
0 1 duration
0 6 working-hours

Selected attributes: 2,3,11,14,16,12,13,9,7,5,15,4,10,8,1,6 : 16

Ranked attributed are displayed according to the attribute selection that 0.2948 is with lead rank shown in 2nd attribute name as wage-increase-first-year and stand first rank, 0 is with least rank shown in 6th attribute name as working an hour and 1st attribute name as duration and stand 16th and 15th rank accordingly.

TABLE 2:
DETAILED ACCURACY BY CLASS WEIGHTED AVERAGE (AFTER SELECTING ATTRIBUTE)

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
10 Fold Cross Validation	0.649	0.297	0.629	0.649	0.628	0.635
Training & Test Set	0.865	0.113	0.869	0.865	0.866	0.896
Spilt percentage (66%)	0.643	0.357	0.577	0.643	0.559	0.662

3.2 Tree Pruning

One essential type of knowledge that can be attained from data mining is the decision tree (DT), which is built from existing data to classify upcoming data [15]. A decision tree consists of nodes, edges and leaves. Decision tree pruning is initiated using cross-validation through the perfect important stage for applying important test [16]. Pruning methods were developed for solving this dilemma [17]. Thus here show the J48 tree pruning for labor dataset before removing the missing values. The pruned tree is predicting the class with good and bad.

J48 pruned tree
wage-increase-first-year <= 2.5: bad (15.27/2.27)

wage-increase-first-year > 2.5
| statutory-holidays <= 10: bad (10.77/4.77)
| statutory-holidays > 10: good (30.96/1.0)

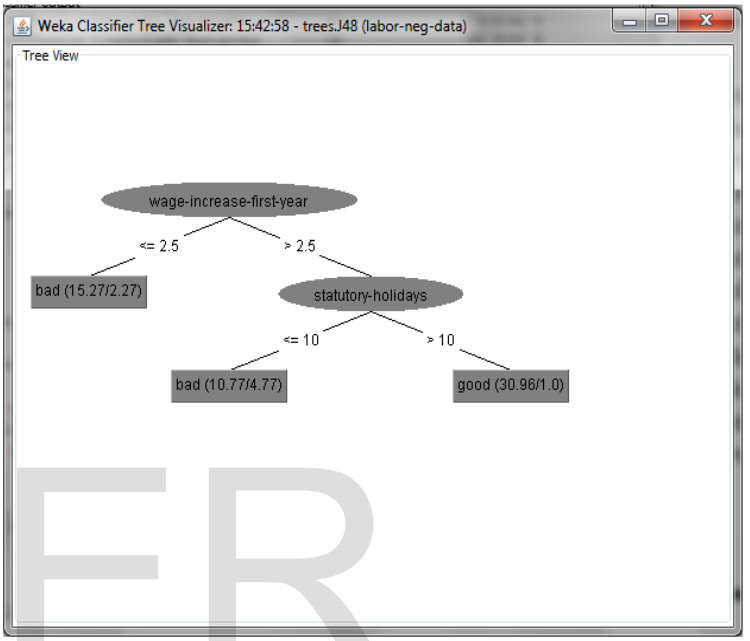

Fig 1: J48 Tree view before attribute selection

Thus here show the J48 tree pruning for labor dataset after removing the missing values. The pruned tree is predicted with the contribution to dental plan is normally shown with none, half and full.

J48 pruned tree
duration <= 1: none (5.14/0.14)
duration > 1
| contribution-to-dental-plan = none
| | longterm-disability-assistance = yes
| | | wage-increase-second-year <= 4.4: full (3.08/0.17)
| | | wage-increase-second-year > 4.4: half (2.24/0.24)
| | longterm-disability-assistance = no: none (2.37/0.3)
| contribution-to-dental-plan = half
| | wage-increase-third-year <= 4.6
| | | duration <= 2: half (3.48/1.21)
| | | duration > 2: full (5.07/0.74)
| | wage-increase-third-year > 4.6: half (4.08/0.84)
| contribution-to-dental-plan = full: full (11.53/1.36)

TABLE 3: PERCENTAGE OF DIFFERENT CLASSIFIER MODE (BEFORE AND AFTER SELECTING ATTRIBUTE)

	10 Fold Cross Validation		Training and Test Set		Split percentage (66%)	
	Before	**After**	**Before**	**After**	**Before**	**After**
Correctly Classified Instances	73.6842 %	64.8649 %	87.7193 %	86.4865 %	89.4737 %	64.2857 %
Incorrectly Classified Instances	26.3158 %	35.1351 %	12.2807 %	13.5135 %	10.5263 %	35.7143 %
Kappa statistic	0.4415	0.3729	0.745	0.7722	0.7564	0.3269
Mean absolute error	0.3192	0.2932	0.195	0.1484	0.2381	0.3105
Root mean squared error	0.4669	0.4254	0.304	0.2479	0.3419	0.4108
Relative absolute error	69.7715 %	72.2549 %	42.6664 %	36.6804 %	52.4444 %	75.3359 %
Root relative squared error	97.7888 %	94.6528 %	63.6959 %	55.3247 %	72.9745 %	89.2476 %

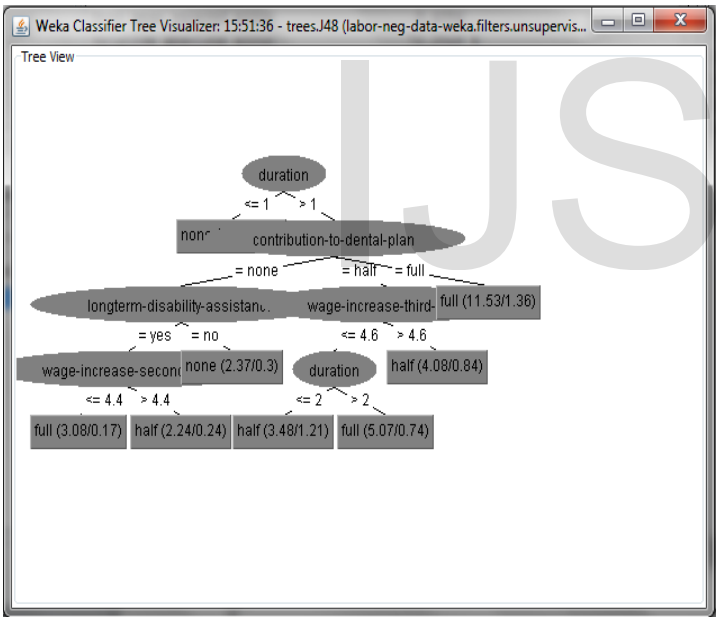

Fig 2: J48 Tree view after attribute selection

The above diagram shows a decision tree view for J48 with cross validation before and after the attribute selector. 1st Diagram showing with 17 attributes with class "good" and "bad", 2nd diagram show with 16 attribute were finally decided with "full" and "half". When the decision tree contains before select about 3 Leaves and 5 trees, and after select contain about 8 leaves and 14 trees.

4 CONCLUSIONS

The classification algorithm J48 with decision tree using different test options. The test option is tested for both after and before the attribute selection and a weighted average is calculated in percentages. Although cross validation and split the percentage is huge differences, but the training and test but the training and test set almost produced nearby result.

REFERENCES

[1] Caio Soares, Philicity Williams, Juan E. Gilbert, Gerry Dozier, "A Class-specific Ensemble Feature Selection Approach for Classification Problems" ACMSE '10, Oxford, MS, USA, April 15-17, 2010.

[2] T. Mitchell. Machine Learning. McGraw-Hill, New York, 1997.

[3] Hall MA. Correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of the 17th international conference on machine learning. California: Stanford University/Morgan Kaufmann Publishers; 2000.

[4] B. Chandra, RaviKothari, PallathPaul "A new node splitting measure for decision tree construction", Science direct Pattern Recognition 43 (2010) 2725–2731

[5] https://list.scms.waikato.ac.nz/pipermail/wekalist/2009-February/042331.html

[6] Almuallim H., Dietterich T.G. . Learning with many irrelevant features.. In: *Proc. AAAI-91*, Anaheim, CA, pp. 547-552,1991.

[7] Kira K., Rendell L.A. . The feature selection problem: traditional methods and a new algorithm.. In: *Proc. AAAI-92*, San Jose, CA, pp. 122-126, 1992.

[8] Blum A.I., Langley P. "Selection of relevant features and examples in machine learning". *Artificial Intelligence*, Vol 97, pp. 245-271, 1997.

[9] Kohavi R., John G.H., "Wrappers for feature Subset Selection." Artificial Intelligence, vol. 97, pp 273-324, 1997.

[10] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 491–502.
[11] Jiawei Han and Micheline Kamber, "Data mining concepts and techniques" Morgan Kaufman Publishers, 2006 Elsevier pp. 297- 298
[12] http://www.d.umn.edu/~padhy005/Chapter5.html
[13] Quan Sun "Sampling-based Prediction of Algorithm Runtime" The University of Waikato.September 2009 pp 11-13
[14] Richard Kirkby, Eibe Frank "WEKA Explorer User Guide for Version 3-4-3" The university of Waikato, November 9, 2004 pp 6
[15] Yen-Liang Chen a, Hsiao-Wei Hua, Kwei Tang "Constructing a decision tree from data with hierarchical class Labels", Elsevier, Expert Systems with Applications 36 (2009) 4838–4847
[16] Eibe Frank, "Pruning Decision Trees", thesis, University of Waikato, Jan 2000, pp 128
[17] Breiman L., Friedman J., Olshen R., and Stone C.. Classification and Regression Trees. Wadsworth Int. Group, 1984.
[18] Salzberg SL: C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Mach Learn* 1994, 16:235-240.