*2009 International Conference on Machine Learning and Computing*
*IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore*

# Feature Selection as an Improving Step for Decision Tree Construction

Mahdi Esmaeili [1], Fazekas Gabor [2] [+]

[1] Department of Computer Science, Islamic Azad University (Kashan Branch), Iran
[2] Faculty of Informatics, University of Debrecen, Hungary

**Abstract.** The removal of irrelevant or redundant attributes could benefit us in making decisions and analyzing data efficiently. Feature Selection is one of the most important and frequently used techniques in data preprocessing for data mining. In this paper, special attention is made on feature selection for classification with labeled data. Here an algorithm is used that arranges attributes based on their importance using two independent criteria. Then, the arranged attributes can be used as input one simple and powerful algorithm for construction decision tree (Oblivious Tree). Results indicate that this decision tree using featured selected by proposed algorithm outperformed decision tree without feature selection. From the experimental results, it is observed that, this method generates smaller tree having an acceptable accuracy.

**Keywords:** Decision Tree, Feature Selection, Classification Rules, Oblivious Tree

## 1. Introduction

Feature selection plays an important role in data mining tasks. Methods always perform better with lower-dimensional compared to higher-dimensional data. Irrelevant or redundant attributes as useless information often interfere with useful ones. In the classification task, the main aim of feature selection is to reduce the number of attributes used in classification while maintaining acceptable classification accuracy.

In optimal feature selection, all possible feature combinations should be searched. This searched space is exponentially prohibitive for exhaustive search even with a moderate attributes. In this case, the high computational cost is still a problem unsolved. Under certain circumstances, suboptimal feature selection algorithms are an alternative. Though suboptimal feature selection algorithms do not guarantee the optimal solution, the selected feature subset usually leads to a higher performance in the induction system (such as a classifier). Search may also be started with a randomly selected subset in order to avoid being trapped into local optimal [1].

Each feature selection algorithm needs to be evaluated using a certain criterion. An optimal subset selected utilizing one criterion may not be optimal according to another criterion.

An evaluation criterion can be broadly categorized into two groups based on their dependency on mining algorithms that will finally be applied on the selected feature subset [2]. An independent criterion, as the name suggests, tries to evaluate a feature subset by characteristics of the training data without involving any mining algorithm. Some popular independent criteria are distance measures, information measures, dependency measures, and consistency measures [3][4][5]. Instead, a dependent criterion requires a predetermined mining algorithm in feature selection and uses the performance of the mining algorithm applied on the selected subset to determine which features are selected.

There are two main techniques for feature subset selection, i.e. the filter and wrapper methods. All filter methods use heuristics based on general characteristics of the data rather than a learning algorithm to

[+] Tel.: +98 361 5550055; fax: +98 361 5550056.
*E-mail address*: M.Esmaeili@iaukashan.ac.ir , Fazekas.Gabor@icrc.unideb.hu .

evaluate the merit of feature subsets. Wrapper methods for feature selection use an induction algorithm to estimate the merit of feature subsets. Filter methods are in general much faster than wrapper methods and more practical for using on high-dimensional data. Feature wrappers often achieve better results than filter due to this fact that they are tuned to the specific interaction between an induction algorithm and its training data [6]. Early research efforts mainly are focused on feature selection for classification with labeled data where class information is available [1][2][7][8].

Divide-and-conquer algorithms such as *ID3* choose an attribute to maximize the information gain; proposed algorithm which we will describe chooses an attribute to maximize the probability of the desired classification.

Experiments with a decision tree learner (*C4.5*) have shown that adding to standard datasets a random binary attribute generated by tossing an unbiased coin affects classification performance, causing it to deteriorate (typically by 5% to 10% in the situations tested). This happens because at some point in the trees those are learned the irrelevant attribute is invariably chosen to branch on, causing random errors when test data is processed [9]. We should know that there is no single machine learning method which be appropriate for all possible learning problems. The universal learner is an idealistic fantasy.

In this paper be used an algorithm that arrange attributes based on importance by two independent criteria. Then, ranked attributes are used as input for construction one decision tree. Our goal is to consider influences data preprocessing (feature selection) on classification.

This paper is organized as follows. Section 2 is detailed description of the proposed method. Section 3 describes the data sets, results and discussion. Finally, section 5 concludes the research.

## 2. Proposed Method

Proposed method is described in this section. Schematic diagram of method shows in Figure 1.

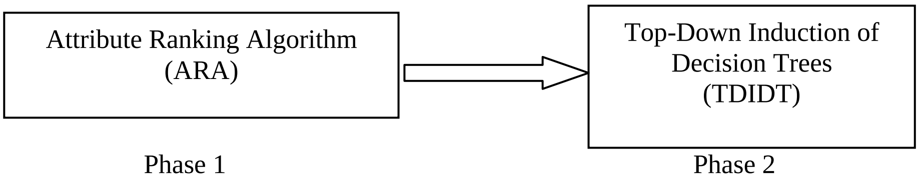
Attribute Ranking Algorithm (ARA)
Top-Down Induction of Decision Trees (TDIDT)
Phase 1
Phase 2

Fig. 1: Schematic Diagram of Proposed Method

### 2.1. The First Phase

In the first phase, it is used attribute ranking algorithm (ARA) before rule generation. In particular, we want to address the inducer to optimize the model through feature selection. In ARA algorithm be used a measure which a kind of this measure was proposed in [10] for determining importance of the original attributes. Then, ranked attributes obtained based on this algorithm are fed as inputs to the second phase.

As mentioned previously, distance measures and dependency measures are two popular independent criteria. In distance measures we try to find the feature that can separate the two classes as far as possible. Dependency measures are also known as correlation measures or similarity measures. They measure the ability to predict the value of one variable from the value of another. In feature selection for classification, we look for how strongly a feature is associated with the class [2].

The ARA includes two parts, class distance ratio and an attribute-class correlation measure. Class distance ratio is measured from two parameters. These parameters are calculated with the kth attribute omitted from each instance. Equation (1) and (2) show how to do this.

$$\textbf{\textit{Distance1}}= \sum_{i=1}^{c} P_i \sum_{k=1}^{n_i} \left[ (X_{ik} - m_i)(X_{ik} - m_i)^T \right]^{1/2} \quad (1)$$

$$\textbf{\textit{Distance2}}= \sum_{i=1}^{c} P_i \left[ (m_i - m)(m_i - m)^T \right]^{1/2} \quad (2)$$

C is the number of classes in the data set and Pi is the probability of the ith class. mi and m are the mean vector of the ith class and mean of all instances in the data set, respectively. ni is number of instances in the ith class, and N is the total number of instances in the data set, i.e., N=n1 +n2 +…+nc

On the other side, the attribute-class correlation measure is used to evaluate the power of each attribute affecting the class label for each instance. The larger the correlation factor, the more important the attribute is for determining the class labels of instances. A great magnitude of attribute class correlation shows that there is a close correlation between class labels and attribute, which indicates the great importance of this attribute in classifying the instances, and vice versa. Equation (3) indicates attribute-class correlation. Equation calculated for attributes that not belong to the same class.

$$\textbf{\textit{Attribute class correlation}}= \sum_{i\#j} \left| X_{ik} - X_{jk} \right| \qquad (3)$$

## 2.2. The Second Phase

In the second phase simple but very powerful algorithm is used for generating rules called *Top-Down Induction of Decision Trees* (*TDIDT*). This has been known since the mid-1960s and has formed the basis for many classification systems, two of the best known being *ID3* and *C4.5*, as well as being used in many commercial data mining packages [11]. Figure 2 shows this algorithm.

```
IF all the instances in the training set belong to the same class THEN
    Return the value of class
ELSE  (a) Select an attribute A from ranked list
      (b) Sort the instances in the training set into subsets, one for
          each value of attribute A
      (c) Return a tree with one branch for each non-empty subset,
          Each branch having a descendant subtree or a class value
          Produced by applying the algorithm recursively
```

Fig. 2: Top-Down Induction of Decision Trees(TDIDT)

# 3. Results and Discussion

The effectiveness of newly proposed method has to be evaluated in practical experiment. For this reason we selected four data sets from *UCI repository* [12]. Table 1 shows training datasets and their characteristics.

Table 1. Data Set Description for Test

| | Number of Attributes | Number of Instances | Number of classes |
|---|---|---|---|
| Iris | 4 | 150 | 3 |
| Monk's Problems | 7 | 432 | 2 |
| Glass Identification | 10 | 214 | 6 |
| Ionosphere | 34 | 351 | 2 |

Above datasets are used as input of *ARA* algorithm, first phase of proposed algorithm. A ranked attributes list are obtained from this phase. Table 2 and Table 3 show output of *ARA* and attribute ordering, respectively.

Table 2: Output of ARA algorithm

| Data Set | Output of ARA |
|---|---|
| Iris | 1662,3471,3727,27 |
| Monk's Problems | 21388,22750,20231,22198,22725,524 |
| Glass Identification | 2452,6280,3210,2554,1413,2206,2127,2875,59 |
| Ionosphere | 1796,10766,9392,11534,9338,10705,10385,10217,9408,10496,9777,11543, 9947,12124,9096,11158,9973,11337,10297,11666,10074,11562,10454, 11081,9995,9458,11398,11370,9837,11535,10115,10441,8927,1800 |

Table 3: Importance ranking results obtained by first phase of proposed algorithm

| Data Set | Attributes Ordering |
|---|---|
| Iris | 3,2,1,4 |
| Monk's Problems | 2,5,4,1,3,6 |
| Glass Identification | 2,3,8,4,1,6,7,5,9 |
| Ionosphere | 14,20,22,12,30,4,27,28,18,16,24,2,6,10,23,32,7,19, 8,31,21,25,17,13,29,11,26,9,3,5,15,33,34,1 |

On the basis of attribute ordering in Table 3, attributes are passed to second phase which constructs a decision tree. As mentioned before in this phase of algorithm, simple and very strong algorithm is used. Figure 3 shows output of this algorithm for Iris dataset. It is obvious from Figure 3, rule ordering is same as attribute ranking. So that the most important attribute compare in the first term of rule. All of rules include this attribute.

```
Field3<=1.7 : Iris-setosa( 48)
Field3>1.7 AND Field2<=2.2 : Iris-versicolor(4/1)
Field3>1.7 AND Field2>2.2 AND Field1<=4.9 : Iris-versicolor(3/2)
Field3>1.7 AND Field2>2.2 AND Field1>4.9 AND Field4<=1.4 : Iris-versicolor(34/2)
Field3>1.7 AND Field2>2.2 AND Field1>4.9 AND Field4>1.4 : Iris-virginica(61/14)
```

Fig. 3: Rule generation by the second phase of algorithm

After tree construction and also confusion matrix, evaluation parameters such as Recall, F-measure, Precision, and Accuracy are calculated. This step is done for all of data set (Table 4).

Table 4: Detailed Accuracy by Class

| Data Set | TP Rate | FP Rate | Recall | Precision | F-measure | Class |
|---|---|---|---|---|---|---|
| Iris | 0.96 | 0 | 0.72 | 1.00 | 0.84 | Iris Setosa |
|  | 0.72 | 0.05 | 0.72 | 0.88 | 0.80 | Iris Versicolour |
|  | 0.94 | 0.14 | 0.90 | 0.77 | 0.83 | Iris Virginica |
| Monk’s Problems | 0.67 | 0.33 | 0.67 | 0.67 | 0.67 | Class 0 |
|  | 0.67 | 0.33 | 0.67 | 0.67 | 0.67 | Class 1 |
| Glass Identification | 0.10 | 0.03 | 0.07 | 0.64 | 0.13 | Building_w_f_p |
|  | 0.96 | 0.58 | 0.84 | 0.48 | 0.61 | Building_w_nf_p |
|  | 0.06 | 0.01 | 0.01 | 0.34 | 0.02 | Vehicle_w_f_p |
|  | 0.54 | 0.01 | 0.07 | 0.78 | 0.13 | Containers |
|  | 0.78 | 0.02 | 0.08 | 0.64 | 0.14 | Tableware |
|  | 0.86 | 0.01 | 0.22 | 0.93 | 0.36 | Headlamps |
| Ionosphere | 0.70 | 0.10 | 0.70 | 0.87 | 0.78 | Bad |
|  | 0.94 | 0.17 | 0.94 | 0.85 | 0.89 | Good |

As explained we need to use other algorithms for comparison them with proposed algorithm. One of the most common applications is *Weka*. The methods that we use in this application are *J48, BFTree, REPTree, and NBTree. Weka* use 10-fold cross validation for accuracy. The standard way of predicting the error rate of a learning technique given a single, fixed sample of data is to use stratified 10-fold cross validation.

The size of induced decision trees is one of the evaluation criteria. Finally we complete our overview with a comparison between proposed algorithm and *Weka* algorithms output. The result of this comparison is summarized in Table 5 and Table 6.

Table 5: Calculation Number of Leaves/Size of Tree for all dataset

| Data Set | J48 | BFTree | REPTree | NBTree | Proposed Method |
|---|---|---|---|---|---|
| Iris | 5/9 | 6/11 | 3/5 | 4/7 | 5/8 |
| Monk's Problems | 2/3 | 2/3 | 8/15 | 1/1 | 4/6 |
| Glass Identification | 30/59 | 16/31 | 12/23 | 9/17 | 14/30 |
| Ionosphere | 18/35 | 11/21 | 5/9 | 8/15 | 13/24 |

Table 6: Comparison of Error rate

| Data Set | Error Rate | | | | |
|---|---|---|---|---|---|
| | J48 | BFTree | REPTree | NBTree | Proposed Method |
| Iris | 0.04 | 0.06 | 0.06 | 0.06 | 0.12 |
| Monk's Problems | 0.25 | 0.25 | 0.15 | 0.25 | 0.33 |
| Glass Identification | 0.34 | 0.33 | 0.38 | 0.30 | 0.45 |
| Ionosphere | 0.09 | 0.10 | 0.11 | 0.10 | 0.18 |

The resulting tree is an oblivious tree. In this kind of tree each level check the same attribute. For this reason, error rating of proposed method is more than other algorithms.

## 4. Conclusion and Recommendations

In this paper, feature selection for decision tree construction is presented. Feature selection as one way of data preprocessing can effect in all steps of data mining algorithms. Attributes importance ranking obtain by running *ARA* algorithm which is the first phase of proposed algorithm. In the next phase simple algorithm is used for generating rules. Finally evaluation parameters such as size of tree, number of leaves, error rate, recall, and precision are computed. Results of comparison show that average number of leaves and size of decision tree generated by proposed method are better than other algorithm. As other data mining algorithm, the results of proposed algorithm depend on characteristic of dataset. However, this method generated smaller trees when comparing with other algorithm such as *J48 or BFTree.* It is also found that error rate is acceptable. For improving accuracy we can repeat two phase of algorithm instead of *TDIDT* method. Thus we have an algorithm with more time complexity but better accuracy.

## 5. Acknowledgments

The authors wish to thank Mansour Tarafdar. His programming and constructive comments and suggestions helped us significantly improve this work.

## 6. References

[1] J. Doak. An Evaluation of Feature Selection Methods and Their Application to Computer Security, technical report, University of California at Davis, *Department of Computer Science*, 1992

[2] Huan Liu, Lei Yu. Toward Integrating Feature Selection Algorithms for Classification and Clustering, *IEEE Transactions on Knowledge and Data Engineering,* Vol. 17, No. 4, pp. 491-502, April-2005

[3] H. Almuallim, T.G. Dietterich. Learning Boolean Concepts in the Presence of Many Irrelevant Features, *Artificial Intelligence,* Vol. 69, pp.279-305, 1994

[4] M.A. Hall. Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning, Proc. 17th Int'l conf. Machine Learning, pp. 359-366, 2000

[5] H. Liu, H. Motoda . Feature Selection for Knowledge Discovery and Data Mining. Boston :*Kluwer Academic,* 1998.

[6] Oded Maimon, Lior Rokach. The Data Mining and Knowledge Discovery Handbook, Springer ,pp. 93-111, pp. 149-164, 2005

[7] M. Dash, H. Liu. Feature Selection for classification, Intelligent Data Analysis: An Int'l J., Vol. 1, No. 3, pp. 131-156, 1997

[8] W. Siedlecki, J. Sklansky. On Automatic Feature Selection, *Int'l J. Pattern Recognition and Artificial Intelligence,* Vol. 2, pp. 197-220, 1988.

[9] Ian H.Witten, Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques, *Second Edition, Morgan Kaufmann*, pp. 288-296, 2005

[10] Lipo Wang, Xiuju Fu. Data Mining with Computational Intelligence, Springer , pp. 117-123,2005

[11] Max Bramer. Principles of Data Mining, *Springer,* pp. 47-48, 2007

[12] Blake, C.L. and Merz. C.J. UCI Repository of Machine Learning Databases. Irvine, CA: University of California, Department of Information and Computer Science. [http://www.ics.uci.edu/~mlearn/MLRepository.html]