A Comparative Study on Wrapper and Filter Methods for Feature Selection in Data Mining

Authors

Anand Rai¹, Dharmendra Kumar²

Department of Computer Science and Engineering United College of Engineering and Research
Allahabad, India

Email-¹anand160489@gmail.com,²kumar.dharmendra@rediffmail.com

ABSTRACT

In recent years, the size of databases has increased enormously. This has led the world to grow interest in the development of tools that can extract knowledge automatically from data. Here, data mining proves to be the useful tool to discover the knowledge from huge data repositories. This requires proper classification methods and algorithms. In a particular dataset, we require minimal set of features that can lead us to the proper classification of instances of the dataset. Here, the need of the good feature selection algorithm comes to existence. In this paper, we perform a comparative study of the two most popular methods used for the feature selection: Wrapper and Filtering method. A number of algorithms exist that fall under these two categories. In this paper relative merits of the respective algorithms and their performance on a number of datasets is analyzed and depicted graphically also.

Keywords— Data Mining, Feature Selection, Classification Algorithm, Decision Tree

INTRODUCTION

Data mining is a process of extraction of information from collection of data which is critical but has potential value. It is a new data analysis technology, and has found its wide application in the areas of insurance, transport, finance, government, and national defence. The most important part in data mining is considered to be data classification. A number of classification methods already exist. The common classification models include neutral network, decision tree, genetic algorithm, statistical model, etc [6], [12], [14]. A dataset may contain numerous attributes. For classification purpose we need the most influencing attributes that affect the existence of an instance into a particular class. Finding out this set of feature is done with the help of a technique called Feature selection. It is the process of selecting a subset of the terms occurring in the dataset and using only this subset as features in classification. Feature selection gains its importance because it serves two main purposes. First [3], it makes training and testing through a
classifier more efficient by decreasing the size of the effective vocabulary. This is of particular importance for classifiers that are expensive to train. Second[7], feature selection often increases accuracy of classification algorithm by eliminating noise features.

**Feature Selectors**

The two methods used for the feature selection are:

**Wrapper Method:**

In Wrapper method [6], [9], the subset selection of attributes takes place based on the learning algorithm used to train the model itself. Every subset that is proposed by the subset selection measure is evaluated in the context of the learning algorithm. This means that computationally intensive learning algorithms cannot be used. The method consists of two phases. In first phase program finds out the possible subsets and in the second phase the subsets are evaluated for their merits.

In this paper the algorithm used for the evaluation of subsets is ClassifierSubsetEval. The ClassifierSubsetEval evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low interrelation are preferred.

ClassifierSubsetEval uses a classifying algorithm for the evaluation of the relative merits of the subset. We, in this experiment, are using NaiveBayes algorithm for this purpose. The selection of subsets is done with the help of BestFirstSearch algorithm.

**Filter Method:**

In Filter method [6], the subset selection procedure is independent of the learning algorithm and is generally a pre-processing step. Obviously, this leads to a faster learning pipeline but it is possible for the criterion used in the pre-processing step to result in a subset that may not work very well downstream in the learning algorithm. The method actually ranks all the attributes of the dataset so that the sufficient amount of attributes may be used for the classification purpose, picking up the attributes sitting at the high end of the rank list.

In this paper, the algorithm used is InfoGainAttributeEval. InfoGainAttributeEval evaluates the worth of an attribute by measuring the information gain with respect to the class.

**Data Classifiers**

Weka provides a number of algorithms [13] which are applicable on different types of datasets. In this paper, basically two classifiers, NaiveBayes and J48 decision tree algorithm are used for classification. The comparison is made on accuracy and sensitivity using true positive and false positive in confusion matrix generated by the respective algorithms. Also we can use the correctly and incorrectly classified instances that give us a most efficient method for classification by using the confusion matrix [6].
Decision tree algorithm J48:

J48 classifier is the extension C4.5 decision tree for classification. It constructs a binary tree. The decision tree approach is really useful in classification problems. With this technique[15], tree is constructed to model the classification process. After successfully building the tree, it is applied to each instance of the dataset and results in the classification of that instance [4], [11].

Algorithm J48:

**INPUT:**

\( D : \) Training data

**OUTPUT:**

\( T : \) Decision tree

\[ \text{DTBUILD} (*D): \]

\( T=\emptyset; \)

T'= Create root node and label with splitting attribute;

T= Add arc to root node for each split predicate and label;

For each arc do:

D= Database created by applying splitting predicate to D;

If stopping point reached for this path, then

T'= create leaf node and label with appropriate class;

Else

T= DTBUILD (D);

T= add T' to arc;

J48 ignores the missing values when it builds a tree. The value for that item can be predicted based on what is known about the attribute values for the other instances. The basic idea is to divide the data into range based on the attribute values for that item that are found in the training sample. J48 allows classification via either decision trees or rules generated from them.

Naive Bayes Classifier:

The Naive Bayes classifier [10] uses Bayes theorem and assumes all attributes to be independent given the value of the class variable. The Naive Bayes classifier is a probabilistic classifier that calculates a set of probabilities by counting the frequency and combinations of values in a given data set. This conditional independence assumption rarely holds true in real world applications, hence the characterization as Naive yet the algorithm tends to perform well and learn rapidly in various supervised classification problems [8]. Naive Bayesian classifier is based on Bayes’ theorem and the theorem of total probability. The probability that a dataset \( d \) with vector \( x = <x_1,\ldots, x_n> \) belongs to hypothesis \( h \)[4] is:
Here, \( P(h_1/x_i) \) is posterior probability, while \( P(h_1) \) is the prior probability associated with hypothesis \( h_1 \). For \( m \) different hypotheses, we have

\[
P(x_i) = \sum_{j=1}^{m} P(x_i|h_j)P(h_j)
\]

(2)

Thus, we have,

\[
P(h_1/x_i) = \frac{P(x_i|h_1)P(h_1)}{P(x_i)}
\]

(3)

**Measuring Performance**

Every algorithm is developed to perform some task. We need some performance measures that can determine how well a particular algorithm executes that task and how it is better than the other. A classification algorithm is said to perform well if it classifies maximum number of instances accurately. Classification accuracy is usually calculated by determining the percentage of instances placed in a correct class. This ignores the fact that there also may be a cost incurred with an incorrect assignment to the wrong class. The ROC (receiver operating characteristic) curve or an OC (operating characteristics) curve or ROC (relative operating characteristic) curve shows the relationship between false positives and true positives. An OC curve was originally used in communication areas examining false alarm rates. It has also been used in information retrieval to examine fall out (percentage of retrieved that are not relevant) VS recall (percentage of retrieve that are relevant) [1], [2].

**Confusion Matrix:**

A confusion matrix shows the accuracy of the classification performed by the classification algorithm. A confusion matrix contains information about actual and predicted classifications done by a classification algorithm. Performance of such systems is commonly evaluated using the data in the matrix.

Given \( n \) classes a confusion matrix is a \( n \times n \) matrix, where \( C_{ij} \) indicates the number of instances from dataset \( D \) that were assigned to class \( C_j \) but where the correct class is \( C_i \). The optimal solution is the solution having zero values outside the diagonal [1]. Here is an example confusion matrix:

\[
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix}
\]

The entries in the confusion matrix have the following meaning:
- \( a \) is the number of correct predictions that an instance is negative,
- \( b \) is the number of incorrect predictions that an instance is positive,
c is the number of incorrect predictions that an instance is negative, and
d is the number of correct predictions that an instance is positive.

Following is the terminology used in classification:

**True Positive (TP):** If the algorithm predicts the class as p and the actual value is also p, then it is called a true positive.

**False Positive (FP):** If the algorithm predicts the class as p and the actual value is not p, then it is called a false positive.

**Precision And Recall:** Precision is the fraction of retrieved instances that are relevant, and recall is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance. Precision can be seen as a measure of exactness or quality, whereas recall is a measure of completeness or quantity. Recall is nothing but the true positive rate for the class.

All the experimental work, in this paper, has been done using WEKA (Waikato environment for knowledge analysis) tool. The tool provides the mechanisms for the comparison of Feature selection algorithms. We have calculated the efficiency based on accuracy in predicting the instances.

The dataset which we have used here for feature selection and classification is available on web URL http://www.cs.bme.hu/~kiskat/adatb/bank-datatrain.arff. This bank relation contains 9 attributes with 300 instances.

**EXPERIMENTAL WORK AND RESULTS**

A number of tools are available for classification purpose. WEKA [5] tool for the feature selection and classification purpose, provides a set of algorithms that can be directly applied to the datasets and the results can be used for the further analysis.

Here is the outcome of the Naïve Bayes classifier after execution on the bank dataset:

---

**Fig 1:** Naive Bayes classification
Now we apply the subset evaluator, the ClassifierSubsetEval that comes under the wrapper method. The algorithm used for predicting the merit of the subsets is NaiveBayes and greedy method is used for the selection of the subsets. Figure 2 shows the attributes selected by the algorithm.

```
--- Attribute Selection on all input data ---

Search Method:
  Greedy Stepwise (forwards).
  Start set: no attributes
  Merit of best subset found: 0.327

Attribute Subset Evaluator (supervised, Class (nominal): 9 pop)
Classifier Subset Evaluator
  Learning scheme: weka.classifiers.bayes.NaiveBayes
  Scheme options:
  Hold out/test set: Training data
  Accuracy estimation: classification error

Selected attributes: 1, 6, 7, 8 : 4
  age
  children
  car
  mortgage
```

**Fig 2:** Feature Selection with ClassifierSubsetEval

Now the results obtained by selecting only the attributes told by the ClassifierSubsetEval algorithm are shown in Figure 3. We see a considerable improvement in the statistics.

```
Time taken to build model: 0 seconds
--- Stratified cross-validation ---
--- Summary ---
Correctly Classified Instances   132   64   %
Incorrectly Classified Instances  108   36   %
Kappa statistic                  0.3050
Mean Absolute Error             6.5273
Root mean squared error         6.6886
Relative Absolute Error         94.0601%
Root relative squared error     97.0294%
Total Number of Instances      240

--- Detailed Accuracy By Class ---

<table>
<thead>
<tr>
<th></th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>ROC Area</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.522</td>
<td>0.259</td>
<td>0.632</td>
<td>0.522</td>
<td>0.571</td>
<td>0.639</td>
<td>YES</td>
</tr>
<tr>
<td>b</td>
<td>0.741</td>
<td>0.478</td>
<td>0.741</td>
<td>0.49</td>
<td>0.639</td>
<td>0.639</td>
<td>NO</td>
</tr>
</tbody>
</table>

Weighted Avg.  0.64

--- Confusion Matrix ---

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>72</td>
<td>44</td>
</tr>
<tr>
<td>b</td>
<td>42</td>
<td>120</td>
</tr>
</tbody>
</table>
```

**Fig 3:** NaiveBayes classification after feature selection

The results which we achieved by applying the J48 algorithm on the same dataset are following. The Figure 4 shows the classification with no attributes removed from the dataset.
Selection of features with the help of ClassifierSubsetEval results into the statistics shown in Figure 5.

**Fig 4: J48 classification**

This shows the considerable improvement in the classification after the attributes are removed which are not present in the attribute list of subset evaluator.

**Fig 5: Feature Selection with Classifier Subset Eval**
Fig 6: J48 Classification After Feature Selection

Till now we dealt with the algorithms which fall under the wrapper method category. It is time to analyse the performance of the algorithm which falls under the Filter method category. For this purpose we use InfoGainAttributeEval algorithm. The filtering algorithms actually present the ranks of all the attributes in the dataset unlike the wrapper methods where the best subset is presented after the application of the algorithm. The Figure 7 shows the ranks of all the attributes in the dataset after application of the InfoGainAttributeEval.

Fig 7: J48 Attribute Ranks From Filter Method
We select all the attributes present in the dataset except the attributes age and sex because they sit at the bottom of the ranking list. So removing the stated attributes and applying the NaiveBayes classification algorithm we get the results shown in Figure 8.

<table>
<thead>
<tr>
<th>Time taken to build model: 0 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>--- Stratified cross-validation ---</td>
</tr>
<tr>
<td>--- Summary ---</td>
</tr>
<tr>
<td>Correctly Classified Instances: 184</td>
</tr>
<tr>
<td>Incorrectly Classified Instances: 114</td>
</tr>
<tr>
<td>Kappa statistic: 0.2089</td>
</tr>
<tr>
<td>Mean absolute error: 0.4443</td>
</tr>
<tr>
<td>Root mean squared error: 0.4993</td>
</tr>
<tr>
<td>Relative absolute error: 99.439%</td>
</tr>
<tr>
<td>Root relative squared error: 99.3691%</td>
</tr>
<tr>
<td>Total Number of Instances: 300</td>
</tr>
</tbody>
</table>

--- Detailed Accuracy By Class ---

<table>
<thead>
<tr>
<th>Class</th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>ROC Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>0.735</td>
<td>0.528</td>
<td>0.62</td>
<td>0.735</td>
<td>0.67</td>
<td>0.67</td>
</tr>
<tr>
<td>NO</td>
<td>0.410</td>
<td>0.300</td>
<td>0.612</td>
<td>0.413</td>
<td>0.606</td>
<td>0.607</td>
</tr>
</tbody>
</table>

--- Confusion Matrix ---

<table>
<thead>
<tr>
<th>a</th>
<th>b &lt;- classified as</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>a = YES</td>
</tr>
<tr>
<td>43</td>
<td>b = NO</td>
</tr>
</tbody>
</table>

**Fig 8:** Classification After Applying Filter Method

Figure 9 shows the curve for the cost benefit analysis when applying the NaiveBayes Classifier on the bank dataset. Figure 10 shows the same curve after the application of the wrapper feature selection method as described above and finally the Figure 11 shows the curve after the application of the filter method.

**Fig 9:** Cost Benefit Analysis (Naivebayes Classifier)

**Fig 10:** Cost Benefit Analysis (Naivebayes Classifier) After Application Of Wrapper Feature Selection Method
CONCLUSION

In this paper, we have presented the extensive empirical comparison between wrapper and filter methods of feature selection for classification of data. After performing a number of experiments on different datasets we come to the conclusion that feature selection can increase the quality of the results while reducing the complexity of the learning task.

Looking at the results from our experiments and as widely reported in the literature, wrapper methods tend to be superior to filters. The wrapper method provides the most efficient subset of features that can lead to the maximum number of correctly classified instances whereas the filter method provides us with all the features of the dataset together with their respective ranks. The ranks of the features provided by the filter method algorithms help to select the most distinguishing features for classification and thus reducing the overall complexity of the classification process.

Finally, it would be interesting to perform additional comparisons employing different wrapper and filter approaches.

REFERENCES


[14] Li Xiong. *Data mining: Concepts and technique*