Discretization of continuous attributes using ReliefF

Marko Robnik Šikonja
Igor Kononenko
University of Ljubljana
Faculty of Electrical Engineering and Computer Science
Tržaška 25,
61001 Ljubljana,
Slovenia
tel.: +386 61 1768-386, fax: +386 61 264-990
{Marko.Robnik, Igor.Kononenko}@fer.uni-lj.si

Abstract

Many existing learning algorithms expect the attributes to be discrete. Discretization of continuous attributes might be difficult task even for domain experts. We have tried the non-myopic heuristic measure ReliefF for discretization and compared it with well known dissimilarity measure and discretizations by experts. An extensive testing with several learning algorithms on six real world databases has shown that none of the discretizations has clear advantage over the others.

1 Introduction

Discretization divides values of the continuous attribute into a number of intervals. We represent each interval by a symbol and treat the attribute as discrete.

Many inductive learning algorithms expect continuous attributes to be discretized before learning; they use some discretization algorithm as preprocessing. Discretization is often considered a task for domain expert and used as background knowledge for specific problem, but often she is unavailable or unable to specify exact intervals. With the help of automatic procedure the task becomes easier.

Discretization of attributes can reduce the learning complexity and help to understand the dependence between attributes and the target concept. Also, in the constructive induction process, the discretized attributes are needed for most operators [13].

There are several methods we can use to discretize an attribute. We can divide them into two groups.

- Class blind techniques (equal width, uniform frequency, maximal marginal entropy) are easy to implement, but require user to specify the number of intervals. As they ignore class of examples when partitioning them into different intervals, the classification algorithm might no longer be able to separate examples of different classes. The essential information may be lost and consequently the method performs poorly in many situations [11].

- Class aware methods try to partition the examples into intervals that most closely correspond to the concept. They differ about the measure they use to estimate the closeness.

The heuristic measure we use is ReliefF [6] (an extension of Relief [4]) which is a powerful attribute estimator performing well on artificial as well as real world problems [8].

We have used ReliefF to guide the discretization process and compared the results with another discretization which uses attribute dissimilarity [9, 3]. We have tested both measures with several machine learning algorithms and databases.

2 Discretization algorithm

We have used the following greedy algorithm for discretization of attributes (similar algorithm was used by [3]):

BestDiscretization = \{\}
SetOfBoundaries = \{\}
repeat
    add the split which maximizes the heuristic measure to the SetOfBoundaries
    if SetOfBoundaries is best so far then
        BestDiscretization = SetOfBoundaries
    until n times the heuristic is worse than in previous step

At each step the algorithm searches for the value which, when added to the current set of boundaries,
maximizes the heuristic estimate of the discretized attribute.

We compared performance of two heuristic estimates.

ReliefF [6] is an extension of Relief [4] which can deal with missing values and multivalued classes and also extends the power of estimation.

Intuitively, we can present ReliefF as estimating probabilities that two near learning examples from the same class have the same value of the attribute and that two near examples from different class have the same value of the attribute. The first probability testifies for the predicative power of attribute, while the second testifies against it. Two near examples are defined as having small distance in the space of all attributes. These two probabilities for different near surroundings are combined into a single estimate with range from -1 to 1. The larger the estimate the more significant the attribute.

For comparison to our novel approach we have used a well known measure of dissimilarity between attribute and class [9]. It is defined as:

$$D(A, C) = \frac{H(A|C) + H(C|A)}{H(A&C)}.$$  \hspace{1cm} (1)

where $H(X)$ represents entropy of variable $X$. It is defined as

$$H(X) = -\sum_x p(x) \log p(x)$$  \hspace{1cm} (2)

where $x$ runs over all possible values of $X$.

The measure takes into account the conditional entropy of both the dependent and the independent variable, and normalizes them with entropy of their co-occurrence.

3 Learning algorithms

The main advantage of ReliefF to other attribute estimators is its non-myopic behaviour. ReliefF can namely predict the quality of strongly dependent attributes. To observe if this feature helps in discretization of continuous attributes, we have selected three learning algorithms capable of some sort of constructive induction, and two without such capability.

All the algorithms except the Semi-naive Bayesian classifier are top-down decision tree builders equipped with pruning and handling of missing values similar to Assistant's [2]. We have reimplemented [12] LFC (Lookahead Feature Construction) [10], an algorithm who, at each node of the tree, joins attributes with conjunction, disjunction and negation and uses these constructs instead of original attributes to split the learning set. RCI (ReliefF in Constructive Induction) is similar to LFC, but it uses ReliefF instead of the posterior entropy to estimate the quality of attributes (constructs). It has an additional constructive operator (equality).

The naive Bayesian classifier presumes independence of attributes and therefore uses the naive Bayesian formula to predict classes of unseen examples. The Semi-Naive Bayesian classifier [5] is its extension: it tries to join certain values of attributes into new attributes, hoping to capture crucial dependencies. Formation of new attributes is a kind of constructive induction.

Figure 1 summarizes important features of learning algorithms.

<table>
<thead>
<tr>
<th>Name</th>
<th>Heuristic</th>
<th>Construction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assistant-I</td>
<td>entropy</td>
<td>no</td>
</tr>
<tr>
<td>LFC</td>
<td>entropy</td>
<td>yes</td>
</tr>
<tr>
<td>Assistant-R</td>
<td>ReliefF</td>
<td>no</td>
</tr>
<tr>
<td>RCI</td>
<td>ReliefF</td>
<td>yes</td>
</tr>
<tr>
<td>SN-Bayes</td>
<td>Bayes</td>
<td>yes</td>
</tr>
</tbody>
</table>

4 Experimental scenario

We have undertaken an extensive testing of both discretization methods on all five learning algorithms on six real world and one artificial domain. Figure 2 gives short description of testing domains.

<table>
<thead>
<tr>
<th>Domain</th>
<th>attr</th>
<th>cont</th>
<th>cl</th>
<th>ex</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast cancer</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>288</td>
<td>80</td>
</tr>
<tr>
<td>Diabetes</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>768</td>
<td>65</td>
</tr>
<tr>
<td>Heart</td>
<td>13</td>
<td>7</td>
<td>2</td>
<td>270</td>
<td>56</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>19</td>
<td>6</td>
<td>2</td>
<td>155</td>
<td>79</td>
</tr>
<tr>
<td>Iris</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>150</td>
<td>33</td>
</tr>
<tr>
<td>Rheumatology</td>
<td>32</td>
<td>21</td>
<td>6</td>
<td>355</td>
<td>66</td>
</tr>
<tr>
<td>XOR2-cont</td>
<td>12</td>
<td>12</td>
<td>2</td>
<td>512</td>
<td>50</td>
</tr>
</tbody>
</table>

Figure 2: Description of testing domains (attr=number of all attributes, cont=number of continuous attributes, cl=number of classes, ex=number of examples, %=majority class percentage)

All the real world domains are well known in machine learning community and can be acquired from various sources (e.g. Irvin machine learning database repository at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/). Each of the these databases
already has the discretization boundaries available. These boundaries were mostly suggested by domain experts. We have also tested the algorithms on these discretizations. Results on these discretizations can be considered an informative benchmark.

The artificial domain represents continuous version of order 2 parity (XOR) problem. Two attributes are informative while others are given random values.

Figure 3 illustrates this problem.

For each domain we have made 10 splits of the data into training and testing set consisting of 70% and 30% of examples respectively. For each split we have used the training set to perform discretization and learning, then we tested on testing set. The results are averaged over ten runs.

We compared discretizations and learning methods in terms of the prediction accuracy and the information score [7]. The later takes the prior probabilities into account and is defined as

\[ \text{Inf} = \frac{\sum_{i=1}^{N} \text{Inf}_i}{N} \]  

\[ \text{Inf}_i = \begin{cases} \frac{-\log_{2} P(C_i) + \log_{2} P(C_i)}{\log_{2} P(C_i)} & \text{if } P(C_i) > P(C_i) \\ \frac{-\log_{2} P(C_i) + \log_{2} P(C_i)}{\log_{2} P(C_i)} & \text{if } P(C_i) < P(C_i) \\ \end{cases} \]

where \( P(C) \) is prior probability of correct class \( C \), \( P(C) \) the probability predicted by classifier and \( N \) is the number of tested examples.

5 Results

As one can see from figures 4, 5, 6, 7 and 8, for real world problems there is no dear winning discretization with neither of learning algorithm.

Results for the artificial problem given in figure 9 clearly indicates advantage of discretization with ReliefF. Only this discretization has preferred the boundary 4 for both informative attributes. All the algorithms were able yo exploit this information, especially the ones with constructive induction.

6 Conclusion

The results show that in spite of ReliefF's theoretical advantage both discretizations perform similarly on real world domains.

We also have to mention that ReliefF's discretization is computational more demanding.

The constructive induction algorithms did not show any advantage over the plain inductive learners. This leads as to the conclusion that the tested real world domains are too easy (the attributes are probably independent), so neither the ReliefF's discretization nor the constructive induction could show its real power.

Our conclusion in a nutshell is: if you suspect your domain has dependent attributes, try ReliefF's discretization, otherwise such heavy artillery is needless.

References

<table>
<thead>
<tr>
<th>Domain</th>
<th>LFC</th>
<th>RCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast</td>
<td>75.9</td>
<td>73.8</td>
</tr>
<tr>
<td>Diab.</td>
<td>69.0</td>
<td>73.8</td>
</tr>
<tr>
<td>Heart</td>
<td>77.3</td>
<td>78.0</td>
</tr>
<tr>
<td>Hepa.</td>
<td>80.6</td>
<td>82.7</td>
</tr>
<tr>
<td>Iris</td>
<td>94.2</td>
<td>94.6</td>
</tr>
<tr>
<td>Rheum.</td>
<td>62.4</td>
<td>57.8</td>
</tr>
</tbody>
</table>

Figure 6: Classification accuracy (in %) for algorithms with construction for given (g), dissimilarity (d) and ReliefF’s (R) discretization

<table>
<thead>
<tr>
<th>Domain</th>
<th>LFC</th>
<th>RCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast</td>
<td>-0.01</td>
<td>-0.02</td>
</tr>
<tr>
<td>Diab.</td>
<td>0.27</td>
<td>0.36</td>
</tr>
<tr>
<td>Heart</td>
<td>0.53</td>
<td>0.54</td>
</tr>
<tr>
<td>Hepa.</td>
<td>0.25</td>
<td>0.31</td>
</tr>
<tr>
<td>Iris</td>
<td>1.46</td>
<td>1.46</td>
</tr>
<tr>
<td>Rheum.</td>
<td>0.42</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Figure 7: Information score (in bits) for algorithms with construction for given (g), dissimilarity (d) and ReliefF’s (R) discretization

Classification and regression trees. Wadsworth Inc., Belmont, California, 1984


