Attribute Selection for Modelling

Igor Kononenko*, Se June Hong**
*University of Ljubljana, Faculty of computer and information science
Tržaška 25, SI-1001 Ljubljana, Slovenia
e-mail: igor.kononenko@fri.uni-lj.si

** IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA
e-mail: hong@watson.ibm.com

Abstract

Modelling a target attribute by other attributes in the data is perhaps the most traditional data mining task. When there are many attributes in the data, one needs to know which of the attribute(s) are relevant for modelling the target, either as a group or the one feature that is most appropriate to select within the model construction process in progress. There are many approaches for selecting the attribute(s) in machine learning. We examine various important concepts and approaches that are used for this purpose and contrast their strengths. Discretization of numeric attributes is also discussed for its use is prevalent in many modelling techniques.

Keywords: attribute quality measures, impurity function, discretization, classification, regression

1 Introduction

A precondition to any data mining is data itself. The purpose of data mining is to explore the data and to eventually discover certain relationships, rules, correlations etc. that can give some insights about the data and can also serve for prediction. The amount of data may be enormous: hundreds of thousands of records, each with hundreds of parameters (features, fields, variables, attributes). We shall use the term, *attributes*, here following the common terminology in machine learning. We are concerned with the traditional data mining task of modelling a target attribute in terms of the other attributes in the data. The target attribute may be categorical (classification) or numeric (regression). In order to efficiently develop a model from the data, heuristics are needed to guide the machine learning process which searches for an optimal model in the given family of models the process is designed for. An important issue in guiding the search is the quality of attributes. This has been a key concern for statistical modelling especially in the linear regression area.

Attributes may be relevant or irrelevant for the task at hand. When there are a large number of attributes, even some relevant attributes may be redundant in the presence of other attributes. Relevant attributes may contain useful information directly applicable to the given task by itself, or the information may be (partially) hidden among a subset of attributes. An important problem is the selection of a reasonable subset of the available attributes so that the selected subset can adequately explain (model) the target. We mention that sometimes an attribute that is strongly correlated with the target may not be a member of the selected subset, for a combination of attributes in a smaller subset without it may be just as adequate. Reducing the number of attributes from hundreds to within a few dozen, not only speeds up the learning process, but also prevents most of the learning algorithms from getting fooled into generating an inferior model by the presence of many irrelevant or redundant attributes. This is mainly because most practical learning algorithms are necessarily heuristic in nature and they often are misled by the presence of many nonessential attributes.

This paper does not deal with deriving more useful attributes from the given set (linear or other functions of subsets of attributes), nor with searching for the telling attributes in the application domain. These are very important problems in their own right. We will, however, discuss discretization of numeric attributes, for many models, including decision trees and rules, explain the target in terms of intervals which are generated either during the model building process or in a pre-processing step. Discretization is also very closely related to the attribute selection problem in that the discretized version is the one that is used in these models (except for the artificial neural networks which deal with numeric values directly).

Another kind of attribute selection takes place in the popular decision tree modelling. Here, at

each branch of the tree, one needs to select the "best" attribute to branch on next. To accomplish the selection task, one employs some *attribute quality measure* for each attribute or some goodness measure (such as the resulting accuracy of the model) on the ensemble of attributes. We shall discuss main approaches for developing the quality measures in the next section. These measures are described for classification first and then extended for regression. In the subsequent sections we examine how the measures are used for actual selection, and how discretization is done. We will also discuss future technical challenges in the concluding section.

We shall speak of examples in the instance space described by a vector of attributes $X_i, i = 1, .., n$ where n is the number of explanatory attributes, which are tagged with the target attribute X_0. These instances are points in the n-dimensional space. As we mentioned earlier, the modelling task is called classification if the target is categorical, and regression if numeric. We shall use $x_{r,i}$ to denote the value of attribute X_i of instance r. For class values we shall use $C(r) = x_{r,0}$.

Each of the attributes can be either categorical or numeric. In the former, the domain of X_i is discrete, i.e. the values x_i are from an unordered symbolic values set, $\{v_{i1}, ..., v_{in_i}\}$ where n_i is the number of values of attribute X_i (for the class attribute we shall use also $c_l = v_{0l}$). In the latter, the domain of the numeric attribute X_i is given by a numeric interval $[\min_i \ .. \ \max_i]$, which is considered continuous for the purpose of modelling. If an attribute takes on discrete values that are ordered in the domain sense, it can be treated as categorical if the number of values is relatively small, or treated as a numeric attribute. A model is a mapping function from the instance space to the target class or target regression value:

$$model : X_1 \times X_2 \times \ ... \ \times X_n \longrightarrow X_0$$

Such mapping should be able to "explain" the target value of the training instances and should be able to classify (predict the class) or produce a regression value of new instances.

2 Estimating the quality of attributes

One can not select attribute(s) without first determining the quality of the attributes in some way relevant to the modelling task. The quality of an attribute should reflect the useful information provided by that attribute. There are two major approaches to estimating the quality of an attribute:

- the quality of an attribute may be estimated by ignoring the other attributes, therefore assuming, for the purpose of estimation, the independence of attributes, or
- the quality of an attribute may be estimated in the context of other attributes.

We could say that the former approach is *myopic*, but its advantage is computational efficiency and in practice it often suffices. However, the latter approach, although computationally more demanding, has a potential of discovering high order dependencies which cannot be detected by myopic approaches.

Most approaches assign a quality measure directly to the attributes but some use indirect measures. For this, a measure is defined first on the modelling task itself using all attributes, then the same measure is computed excluding the attribute of interest. The difference is then interpreted as the contribution of the attribute. This will be discussed in Section 2.3 on the cost of exclusion approach.

2.1 Traditional impurity measures for classification

Traditionally, the quality of an attribute in classification is defined in terms of the purity of classes of training instances. We say that a set of instances is pure if all instances belong to the same class and the set is maximally impure if the proportion of instances in all classes is uniform. The *impurity function* I measures the impurity of a set of instances and achieves the minimum (usually 0) for a pure set, and maximum for a maximally impure set. Impurity functions have their origins in decision tree practices and are mainly used in selecting the best attribute to further split the current node. As such, they are defined for categorical attributes that can take on values which is used to branch out. (If appropriate probability density functions are known, these measures can be extended to directly include numeric attributes, but this is not practiced. We will discuss how the numeric attributes are handled later.)

Let the attribute X_i have n_i possible values. The attribute splits the set of all training instances T with impurity $I(T)$ into n_i subsets T_j, $j = 1..n_i$, each with impurity $I(T_j)$. The quality of attribute $q(X_i)$ is defined to be the expected decrease of the impurity, i.e. the purity gain:

$$q(X_i) = I(T) - \sum_{j=1}^{n_i} (p(v_{ij}) \times I(T_j)) \tag{1}$$

where $p(v_{ij}) = P(x_i = v_{ij})$ is the proportion (probability) of the instances with the j-th value of the attribute X_i among all the instances in the (sub)set being considered. Two well known and

commonly used impurity measures are entropy, used in the ID3 system and its successor C4.5 [1], and the gini-index, used in the CART system [2]. The entropy of a set of instances with proportions of classes $p(c_l) = P(x_0 = v_{0l}),\ l = 1..n_0$ is defined as

$$H(X_0) = -\sum_{l=1}^{n_0} (p(c_l) \times \log_2 p(c_l))$$

The gini index is defined with

$$G(X_0) = 1 - \sum_{l=1}^{n_0} p(c_l)^2$$

Using entropy H for impurity measure I in equation (1) leads to the information-gain and using G leads to the gini-gain. These measures are relatively simple to compute and work well in many applications.

An ideal attribute should enable the exact prediction of the class from the value of the attribute. The class attribute (X_0) itself is therefore an ideal attribute (but, of course, it cannot be used for classification). If the number k of values of an attribute increases, the probability that the subsets of training instances T_j will be completely pure also increases. In an extreme case an attribute with the number of values much larger than the number of training instances will usually have all T_j completely pure. Such attributes will be treated by any purity gain criterion as the best, although such attributes are useless for prediction. Namely, the reliability (probability) that the new instance with the j-th value of the attribute actually belongs to the same class as all training instances from the subset T_j decreases if the number of training instances in T_j decreases. Therefore, the estimates of these purity gains are unfairly biased towards apriori preferring the attributes with more values, which we call *variety bias*. We will discuss approaches to neutralize (by normalization) or avoid this bias later.

Although efficient to compute, these measures are myopic. During the estimation of one attribute all other attributes are ignored. This is an unrealistic assumption which may lead to a significant underestimation of the quality of the attributes when there are strong conditional dependencies. Of course, when these measures are used to select the next branching node in a decision tree, deep in the branches, the context of other attributes above the current node are automatically figured in. But there is a problem near the root node where few or no other attributes' contributions are considered at all. In Section 3.2, we will discuss more about the decision node selection as well as a few other myopic measures with different strengths.

2.2 Direct non-myopic measures for classification

For many problems the values of some subset of attributes, together as a tuple, convey information about the class while the values of an individual attribute may not. One example is when the target is a symmetric function of some attributes, which include majority functions. An extreme case of the symmetric function is the well known parity function, often denoted as EXOR. For binary attributes, EXOR produces 1 iff the sum of the attribute values are odd, and 0 otherwise. Any myopic measure including the purity gain schemes fail to recognize that the EXOR attributes are any better against possible irrelevant attributes, for the impurity of T_0 and T_1 for every EXOR attribute is maximal, and consequently the purity gain is zero, i.e. a single attribute conveys no information about the class.

To overcome this problem, it is necessary to take into account the context of other attributes when the quality of an attribute is measured. There are several approaches to include the context into the estimate, directly as described here or indirectly using the cost of exclusion approach described in Section 2.3.

2.2.1 Lookahead approach

Instead of estimating the (normalized) impurity gain for single attribute, one can estimate the (normalized) impurity gain of a subset of attributes. That is, in decision tree construction, one selects as the current node splitter the attribute that results in the best purity gain at some look-ahead levels down the tree, instead of just the next level. This approach can in principle detect higher order dependencies between attributes with a potentially exponential increase of time complexity. By guiding the lookahead depth one can control the trade-off between the myopia and the time complexity. Usual practice has been limited to one or two additional level lookahead.

2.2.2 Distance based concept of local context

A relatively new and promising approach is to estimate the quality of an attribute by measuring its contribution in class discrimination in the local subspaces of the instance space. The idea is to estimate whether the attribute is able to separate the training instances, that are very similar, i.e that are close to each other in the instance space. By considering only neighbor instances in the instance space we implicitly take into account the context of other attributes. In other words, we focus on local decision boundaries separating close pairs of instances.

To determine which instances are close to each other we need to define a distance measure D_{rs} between two instances r and s. It is defined in terms of each attribute's component distance: for an attribute X_i, the two values, $x_{r,i}$ and $x_{s,i}$ are said to have distance $d_{rs,i}$ as follows.
For a categorical attribute, X_i,

$$d_{rs,i} = \begin{cases} 0 & \text{if} \quad x_{r,i} = x_{s,i} \\ 1 & \text{if} \quad x_{r,i} \neq x_{s,i} \end{cases} \tag{2}$$

and for a numeric attribute, X_i,

$$d_{rs,i} = \min((|x_{r,i} - x_{s,i}|/t_i), 1) \tag{3}$$

where t_i is a threshold which is usually set at half the magnitude of the range of X_i. (It can be set by domain knowledge.) The notion is that if the values are different by more than t_i, the distance is one as if categorical values were different. When t_i is set to the magnitude of the range itself, this component distance is the same as the traditional measure used in the pattern recognition and machine learning (see [3] for the rationale and elaborations on the use of t_i.) The instance distance D_{rs} can be defined either as Euclidean distance (root sum of squares of component distances) which is more traditional, or as Manhattan distance, as

$$D_{rs} = \sum_{i=1}^{n} d_{rs,i} \tag{4}$$

The computation necessary to find k-nearest neighbors to a given instance r from the set of t instances is of order $t \times n$, because once the distances are computed, the time necessary to find the k smallest among t numbers is at most of order $t \times \log k$ (by maintaining an ordered heap of size k) which does not contribute to the overall complexity.

2.2.3 RELIEF family of local-context purity gain approach

An ideal attribute will be able to separate neighbor instances from different classes by having different values and will have the same values for neighbor instances from the same class.

Kira and Rendell [4] proposed a novel algorithm RELIEF that implements the above idea. Given an instance, Relief searches for its two nearest neighbors: one from the same class (called *nearest hit*) and the other from a different class (called *nearest miss*). For each attribute, the RELIEF measure counts, as contribution from these two close pairs, the nearest miss's component distance minus the nearest hit's component distance. RELIEF accumulates the contributions of m randomly selected training instances and normalizes it by m, which is a user-defined parameter.

```
for i := 1 to n do q[X_i] := 0.0;
for j := 1 to m do
    begin
      randomly select an instance r;
      find nearest hit t and nearest miss s;
      for i := 1 to n do
        q[X_i] := q[X_i] + (d_{rs,i} - d_{rt,i})/m;
    end;
```

Figure 1 RELIEF Algorithm

Let r be an instance from a random subset T' of the training set T, with $|T'| = m \leq |T|$. Furthermore, let s denote the nearest different class neighbor of r (i.e. $C(s) \neq C(r)$), and let t denote the nearest same class neighbor of r (i.e $C(t) = C(r)$). The RELIEF measure for an attribute X_i is essentially (within a constant factor of m):

$$q(X_i) = \sum_r (d_{rs,i} - d_{rt,i}) \tag{5}$$

Figure 1 provides the pseudocode of the RELIEF algorithm.

The nearest neighbor determination is done once for each r for all the attributes, hence the computational complexity for developing the measure for all n attributes is $O(m \times |T| \times n)$, which becomes $O(|T|^2 \times n)$ if $T' = T$. The myopic purity gain measures have complexity of $O(|T| \times n)$. However, RELIEF is much less myopic and can assess the importance of highly interactive attributes.

Kononenko [5] has shown that RELIEF's estimates are highly related to the gini-index with locality context. This work and the follow-on work [6] have extended and generalized RELIEF to ReliefF to deal with noisy, incomplete and multiclass problems. It uses the same rationale: a good attribute should be able to discriminate close pairs of different class instances and, furthermore, it is desirable to have the same value for close pairs of same class instances.

Instead of one near-hit and one near-miss, ReliefF uses k-nearest instances s from each class. Therefore, if c is the number of classes, for each instance r ReliefF searches for k nearest hits and $k \times (c-1)$ nearest misses, and equation (5) is accordingly generalized to:

$$q(X_i) = \sum_r \sum_{C(s) \neq C(r)} \left(\frac{P(C(s))}{1 - P(C(r))} \frac{d_{rs,i}}{k} \right) - \sum_r \sum_{C(s) = C(r)} \frac{d_{rs,i}}{k}$$

where $P(C)$ is the proportion (probability) of class C. ReliefF uses Manhattan distance for instance distances as given in equation (4). A pseudo code for implementing this computation [5] has a flow similar to the pseudo code for the regression case shown in section 2.4. The computational complexity is the same as for basic RELIEF. The k for the k-nearest is designed to make the measure more robust against noise and sparsity of instances. Kononenko et al. [6] report a fixed $k = 10$ to be reasonable for most large applications. The original RELIEF is a special case with $k = 1$ and using Eucledean distances for nearest neighbor determination. It also uses $d^2_{rs,i}$ instead of just $d_{rs,i}$ in the above equation. However, these differences have no significant effect on relative measures.

An important observation is that the ReliefF measure approximates the following difference of probabilities (it is implicit that all (r,s) pairs of instances are neighbors):

$$q(X_i) = P(x_{r,i} \neq x_{s,i} | C(s) \neq C(r)) - P(x_{r,i} \neq x_{s,i} | C(s) = C(r)) \tag{6}$$

This fact is used to extend the measure to the regression case and to show that the RELIEF family of measures are resistant to the variety bias by self-normalizing in some sense, as will be shown later.

2.2.4 Contextual merit (CM) measure

Hong [3] proposed another non-myopic quality measure for an attribute based on a different rationale. When two different class instances r and s are compared, only the attributes that have different values can help discriminate the two. Furthermore, the difficulty of separating the two instances should be a decreasing function of the distance D_{rs}, such as $1/D_{rs}$ (or $e^{-\alpha D_{rs}}$), which is called the contextual strength of the pair (r,s). The attributes that contribute to the distance are to share this "obligation" in proportion to their component distances, in the context of the fixed value settings of the attributes that do not contribute to the distance. Therefore, the contextual merit for an attribute X_i can be defined as:

$$CM(X_i) = \sum_r \sum_{C(s) \neq C(r)} \frac{d_{rs,i}}{D^2_{rs}} \tag{7}$$

where r is an instance of T (or a random subset T') and s is a different class k-nearest neighbor of r in T as before. The k value is normally set at $\log_2$ of the total number of training instances not in the class of r. Alternatively, it can be a constant, say 10, as in ReliefF for large problems. The complexity of computing the contextual merit is the same as that of ReliefF. The main difference is in the notion of contextual strength, $1/D_{rs}$, and the fact that it focuses on only the class separating

capability of an attribute without regard to whether it has the same value for the same class neighbor instances. Consequently, the contextual merit CM needs to be normalized for its variety bias, but uniquely enjoys the following desirable property.

Suppose an attribute X is classification-equivalent to a set of attributes $Y_1, Y_2, .., Y_g$, that is, whenever the g-tuple of Ys distinguishes the class, so does X. It can be shown that

$$\max_i CM(Y_i) \leq CM(X) \leq \sum_i CM(Y_i).$$

The merit of X approaches the upper bound as the number of instances in the training set increases.

2.3 Cost of exclusion approaches for a non-myopic measure

Instead of estimating the quality of an attribute directly, one can first define a measure on the classification problem with the given set of attributes, and then compare the measure with and without the given attribute. The difference is then a meaningful non-myopic measure for the attribute, since the effect of all attributes are considered in the process.

2.3.1 Cross-validation

Here we assume that a machine learning algorithm is available that is able to construct a classifier whose performance on a set of holdout test cases can be used for the estimate of the quality of the outcome. First the classifier performance for using the entire set of attributes is measured. The process is repeated each time excluding one of the attributes. The quality of an attribute is then defined as the difference between the quality of the outcome when all attributes have been used and the quality of the outcome when the given attribute was excluded. This approach is computationally demanding and is therefore impractical or even infeasible for some large real problems. This kind of quality measure forms the basis for subset selection in the *wrapper* approach discussed later.

2.3.2 Contextual merit measure revisited

Given two neighbor instances, r and s, in two different classes, we mentioned earlier that the difficulty of separating them should behave like $1/D_{rs}$. For a given attribute X_i, we can rewrite D_{rs} as the sum of $d_{rs,i}$ and the distance from the other attributes excluding X_i, $\Delta_{rs,-i}$. We have

$$\Delta_{rs,-i} = D_{rs} - d_{rs,i} = \sum_{i \neq j} d_{rs,j}$$

Now the difference of the difficulties calculated with and without X_i is:

$$\frac{1}{\Delta_{rs,-i}} - \frac{1}{D_{rs}} = \frac{d_{rs,i}}{\Delta_{rs,-i} D_{rs}} \approx \frac{d_{rs,i}}{D_{rs}^2}$$

Thus, the contextual merit measure is related to the cost of exclusion approach. (This observation is due to S. Winograd, private communication). If the difficulty is estimated as $e^{-\alpha D_{rs}}$ instead, for some small constant α, one can similarly approximate the cost of exclusion as proportional to $d_{rs,i} e^{-\alpha D_{rs}}$. This quantity can be summed for all (r,s) different class neighbor pairs as an alternate form of contextual merit instead of equation (7).

2.4 Attribute quality measures for regression

The usual impurity measure used in regression for categorical attributes is the mean squared error (MSE) [2]:

$$MSE = \frac{1}{|T|} \sum_{r=1}^{|T|} (x_{r,0} - \overline{x}_0)^2 \tag{8}$$

where

$$\overline{x}_0 = \frac{1}{|T|} \sum_{r=1}^{|T|} x_{r,0}$$

is the average class value. It has been shown in [2] that there is an interesting relation between the mean squared error and the gini-index. If the classification problem with two classes is transformed into a regression problem by labeling one class with 0 and the other with 1, then the following is true:

$$gini = 2 \times MSE.$$

MSE has deficiencies similar to those of ordinary impurity measures in classification. MSE-gain is biased towards overestimating multivalued attributes which can be alleviated by normalization (see section 2.5). Also, it is myopic as it ignores the rest of the attributes when estimating the quality of a given attribute.

Many measures that are defined for a categorical target attribute (i.e., classification problems) can be used for regression by some appropriate discretization of the target numeric attribute. (See, for instance, [7]). This approach first turns the regression problem into a classification problem and then uses the resulting classifier to compute a numeric outcome by an averaging procedure.

set all N_{dC}, $N_{dX}[X_i]$, $N_{dC\&dX}[X_i]$, $q[X_i]$ to 0;
for j := 1 **to** $m = |T'|$ **do begin**
  randomly select instance $r \in T$;
  find **k** instances {s} nearest to r;
  foreach s of k-nearest **do begin**
    $N_{dC} := N_{dC} + |x_{r,0} - x_{s,0}|/k$;
    for i := 1 **to** n **do begin**
      $N_{dX}[X_i] := N_{dX}[X_i] + d_{rs,i}/k$;
      $N_{dC\&dX}[X_i] := N_{dC\&dX}[X_i] + |x_{r,0} - x_{s,0}| \times d_{rs,i}/k$;
    end;
  end;
end;
for i := 1 **to** n **do**
  $q[X_i] := N_{dC\&dX}[X_i]/N_{dC}$ - $(N_{dX}[X_i] - N_{dC\&dX}[X_i])/(m - N_{dC})$;

Figure 2. RReliefF (Regressional ReliefF) Algorithm

Within the framework of the ReliefF approach, the problem of myopia can be solved for numeric target attributes as well. The class is continuous in regression problems, therefore the (nearest) hits and misses cannot be used. Instead of requiring the exact knowledge of whether two instances belong to the same class or not, we can introduce a kind of probability that two instances are from different classes. This probability can be modeled by the relative distance between the target (class) values of the two instances; i.e., $P(C(r) \neq C(s))$ can be proportionately measured for a given pair (r,s) as $|x_{r,0} - x_{s,0}|$. It is still not possible to estimate q[X] through equation (5), because the information about the sign of each contributing term is missing. The key equation (6) can, however, be reformulated, so that it can be directly evaluated using the probability of two instances belonging to different classes, using Bayese rule.

$$q(X_i) = \frac{P(C(s) \neq C(r)|x_{r,i} \neq x_{s,i})P(x_{r,i} \neq x_{s,i})}{P(C(s) \neq C(r))} - \frac{(1 - P(C(s) \neq C(r)|x_{r,i} \neq x_{s,i}))P(x_{r,i} \neq x_{s,i})}{1 - P(C(s) \neq C(r))} \quad (9)$$

This reformulation of the key relation leads to the RReliefF algorithm given as pseudo code in Figure 2. The subscript dC stands for "different class (target)" and dX for "different attribute value."

The computational complexity of RReliefF is the same as RELIEF. (For further details and application results, see [8]). Similar to ReliefF, RReliefF is also largely self-normalizing against variety bias. Again, variations on distance definitions discussed earlier can be used here as preferred.

2.5 Approaches to counter variety bias

When an attribute has many distinct values, whether categorical or numeric, it tends to impart more information about the target attribute. This may be reliable in some applications especially if the data being modelled is close to the entirety of instances to begin with (such as semi-exhaustive lab data for certain experiments). For most of real world applications, high variety attributes (in extreme cases, these may be unique for each instance, e.g. customer names, account numbers, etc.) are unreliable as inputs to a model as was discussed in Section 2.1. Most of the quality measures exhibit variety bias and, depending on the application domain, they may require some normalization so that the measures for all attributes can be fairly compared. The RELIEF family measures are largely immune from this bias. White and Liu [9] and Kononenko [10] analyzed the bias problem inherent in many different measures.

2.5.1 RELIEF implicitely avoids the variety bias

Recall the rationale for the RELIEF, "a good attribute should be able to discriminate close pairs of different class instances and, furthermore, *it is desirable that it has the same value for close pairs of same class instances.*" The second part of this objective has the direct effect of lowering the quality measure for attributes that have a large variety of values. Again, returning to the key relation of equation (6), the following equivalent expression can be derived [6] for the case of a categorical attribute using Beyese rule (Here we shall drop the subscripts for denoting individual X_i and instead use generic X to denote an attribute, v to denote its distinct values, and c to denote the distinct class values.):

$$q(X) \propto \sum_v P(v)^2 \times \textit{gini'-gain}(X) \tag{10}$$

where

$$\textit{gini'-gain}(X) = \sum_v \left(\frac{P(v)^2}{\sum_v P(v)^2} \times \sum_c P(c|v)^2 \right) - \sum_c P(c)^2 \tag{11}$$

On the other hand the usual gini-gain, using equation (1) is

$$\textit{gini-gain} = \sum_v \left(P(v) \times \sum_c P(c|v)^2 \right) - \sum_c P(c)^2 \tag{12}$$

So the major contrast between equations (11) and (12) is in the factors $P(v)^2/(\sum_v P(v)^2)$ of gini'-gain versus $P(v) = P(v)/(\sum_v P(v))$ of gini-gain. $\sum_v P(v)^2$ in equation (10) is, of course, the

probability that X has the same values in an arbitrary neighboring instance pair. This illustrates why the RELIEF family of measures has a self-normalizing effect against variety bias. And the above equations allow an alternate interpretation for RELIEF, namely, as a modified gini-gain in the locality context, which tends to supress the variety bias.

2.5.2 Approaches to normalization

The usual way to counter bias is to find another measure that has the opposit tendency, and use it to normalize the original measure, say by dividing. Quinlan [1] makes use of the fact that the more the variety of values attribute X has, the larger the value of its own entropy $H(X)$. So the entropy based purity gain is normalized by $H(X)$ in C4.5 tree induction. This certainly helps in many cases, although Kononenko [10] showed that even this *information-gain-ratio* exhibits some variety bias.

Recently Hong et al. [11] proposed using randomization as a basis for normalization. The idea is to compute the expected quality measure (of your choice) for a random attribute that has the same value distribution in place of a given attribute. It has been shown that such an expected quality measure is easily computed for purity-gain measures as well as contextual merit measure. Once the expected random merit is computed, one can compare it to the original merit of the attribute, either by dividing or subtracting. Since the expected measure for a randomized attribute is computed by the same method with the same biases, such normalization tends to cancel the effect of biases, especially the variety bias. It has been shown that the expected gini-gain is a function of only the number of distinct values the attribute takes and not of the actual distribution, and hence, for binarized attributes (equal tests or greater than tests usually used in purity-gain based tree induction) it has no relative effect. Information based measures also behave similarly. However, this approach was shown to be highly effective for the contextual merit. In particular, this approach normalizes the contextual merits of numeric attributes (which tend to be more inflated) so that they can be compared fairly to the merits of categorical attributes.

The handling of variety bias is not a simple problem. Such bias is as it should be in certain applications, i.e. it may not be a bias. Even when a measure is normalized by its randomized counterpart, it is not clear if an attribute, that has tiny quality measure that happens to be far greater than even tinier randomized measure, should be preferred. So, although simple division or subtraction may work for most cases, a better understanding is necessary to properly "normalize" the attribute quality measures for a given application domain.

3 Attribute selection

We now come to the actual selection problem. We have so far discussed some key approaches and issues regarding how to measure the relative quality of an attribute. These measures are used for actual selection depending on where in the modelling process the selection takes place. We will first address the problem of relevant subset selection, followed by the problem of attribute selection in the decision tree induction process.

3.1 Relevant subset selection

Ideally, an induction algorithm should perform better with more attributes. However, irrelevant or redundant attributes often fool most algorithms to induce inferior models as mentioned earlier. This problem is perhaps more acute than the fact that at least a linear increase in computation is required to handle the "extra" nonessential attributes in the modelling process. Therefore, the selection of a subset of essential attributes is an important part of data mining. Our interest here is in obtaining the subset that will result in the "best model" and not in obtaining the subset that includes attributes individually most correlated to the class.

The subset can be arrived at by successively adding an attribute, successively discarding an attribute, or both. Various quality measures we have discussed can be used for deciding which attribute to add or discard. For many practical modelling problems the variety bias of the measure used should be either avoided as in ReliefF or neutralized by normalization. One of the most important questions, to which there is no satisfactory technical solution yet, is how many attributes should be in the selected subset. This is a matter decided by experiments in today's practice: "Try decreasing (increasing) the number of attributes until the best performing model results." (The usual number of attributes for large real applications ranges from 10 to 40).

The reason "successive" adding or discarding is prudent has to do with the fact that all of the quality measures assign either all relatively high values or all relatively low values to strong attributes that happen to be highly correlated to each other (a pedagogical case would be multiple replication of an attribute). Therefore, computing the quality measure once and selecting the top so many attributes, or selecting those above (discarding those below) a certain threshold value would be unwise. Nevertheless, Kira and Rendell reported initial successful application of RELIEF in just such filtering mode.

The quality measures that are related to the cost of exclusion approach assign relatively low values to a set of attributes that happen to be slightly different from each other, for in the presence

of the others excluding one is not very significant even when they are individually very telling. Therefore, these measures are best used in successively discarding mode. Once an attribute is discarded, the measure is recomputed for the remaining attributes, iteratively. Of course, when there are hundreds of attributes initially, one would try to discard some reasonable number of low ranking attributes in each iteration. One must then check to see if some of the discard candidates are highly correlated to each other and, if so, return the best among them before the next iteration.

John et al. [12] defined a disciplined trial approach of arriving at a subset called *wrapper approach*. Kohavi and Sommerfield [13] further refine the wrapper based subset selection by trying for the best addition/deletion as well as some combination of these as a search step. At each step, the choice is determined by comparing the outcome (cross validation of accuracy performance) of all such possibilities. This method is coupled with the chosen modelling technique (e.g. CART, C4.5 like trees, artificial neural networks, DNF rule induction methods, etc.). Since the wrapper method terminates when there is no improvement, the size of the subset is automatically determined. This optimizes in a greedy manner the subset matched to the modelling technique, at the cost of vastly increased computation. See their papers for further references on the subset selection problem and successful applications of this approach. A less greedy wrapper style search can be made by employing a genetic algorithm [14].

When there are too many instances in the training set, the subset selection process (including the computation for the quality measures, often repeatedly) may require too much time. One can use sampling of the training instances to alleviate this problem. In fact, RELIEF explicitly samples in its outer loop (see Figure 1). For non-myopic measures that take $O(|T|^2 \times n)$ computation, one can iteratively double the sample size until the relative magnitude of the measures are stable, which is quite effective in general.

3.2 Attribute selection for decision tree induction

Decision trees constitute a popular model family. A node test partitions the (sub)universe, iteratively, until the instances wihtin each part become "reasonably" pure. The problem is how to find the "best" attribute to split the current node of the tree usually into two branches. The entire instance space is the root node, initially. A node test is usually a modified attribute and this tends to increase the effective number of attributes (= tests and subset tests for categorical attributes and $\leq$ tests for numeric attributes). Traditionally, some chosen quality measure is computed on these modified attributes to determine which is the "best". These include the Gini-gain for CART, information-gain-ratio for C4.5. Kononenko et al. [6] have shown that in many cases using the non-myopic ReliefF measure produced superior tree performances.

When a $\leq$ test for a numeric attribute is selected, it effectively discretizes the attribute by introducing a cut point in its value range (see the section on discretization for more discussion). Quinlan [15] recently proposed modifying C4.5 so that the information-gain of a $\leq$ test be reduced by $\log_2(N-1)/|T|$ where N is the number of distinct values the test's numeric attribute takes on within the part being split. This modification is an MDL-inspired penalty recognizing the fact that $\log_2(N-1)$ bits are required to specify the threshold value for the $\leq$ test, which is amortized for the number of instances in the current part being split. This also has a normalization effect against variety bias, in addition to the use of gain-ratio, for numeric attributes. The proposed new method also discards the numeric attribute from being considered for the best gain-ratio, if all of its possible $\leq$ tests have negative reduced information-gains.

The quality measures related to the cost of exclusion approach can not be directly used for node attribute selection because the measure is in the context of all other attributes present. One can use them, however, if there are no strong mutual correlation among the attributes present.

One of the problems with choosing the attributes for trees, especially near the root node, is that once the choice is made and partitioning is done, a permanent decomposition of the problem has been made, i.e. each branch part is modelled as a separate problem. Questions as to whether each part is statistically significant to support finer models, or whether each part should be different in some other ways than just the fact that the split is made to obtain the best purity-gain, are not addressed by traditional approaches. These issues stem from much practical experience on problems that have many class values, probabilistic data, or a highly skewed class distribution where it has been observed that purity-gain approaches are not consistent performers. We now examine a few approaches that are motivated by these concerns.

Zhou and Dillon [16] proposed the symmetric τ measure which is designed to be relatively free of variety bias and handles the many-valued class attribute problem better. Their τ measure can be expressed for a generic attribute X in the same manner as in equations (10-12) as follows:

$$\tau(X) = \frac{\sum_c \sum_v (P(c,v)^2/P(c)) + \sum_v \sum_c (P(c,v)^2/P(v)) - \sum_c P(c)^2 - \sum_v P(v)^2}{2 - \sum_c P(c)^2 - \sum_v P(v)^2}$$

This can be interpreted as "the reduction in probability of prediction error resulting from knowledge of attribute X values" relative to having no such knowledge.

Smyth and Goodman [17] used the J-measure for measuring the quality of a DNF rule. When it is adopted for a single attribute-value situation, it can be expressed as:

$$J(X,v) = P(v) \sum_c \Big(P(c|v) \log \left(P(c|v)/P(c) \right) \Big)$$

This measure is a kind of reformulation of the conventional information-gain. It is non-negative and it combines a measure of simplicity and goodness-of-fit of a given rule. It can be easily verified that

$$\sum_v J(X, v) = J(X) = \textit{information-gain}$$

These measures may improve on the simple purity-gain measures in terms of being statistically more robust, but they still have the same myopia problem near the root node, i.e. neither the locality context nor the context of other attributes present is accounted for.

Choosing an attribute near the root node is a rather serious commitment. That is why using one of the non-myopic RELIEF family of measures tends to do better. There are several approaches that attempt to be more strategic about the choice. (These are most effective near the root node.) The so called TWOing strategy [2] was developed for many-valued class attribute problems. Here one computes the attribute quality measure (originally gini-gain in CART, but can be any reasonable measure) of attributes, not for the class itself, but for each of the two-way partitioning of the class values into two super-class groups. The overall winning attribute tends to split the class values into most dissimilar groups while splitting the instance space.

Fayyad and Irani [18] proposed another measure based on the idea that the class value distribution in each part should be made as different as possible. For binary candidate attribute tests, each resulting part's class value distribution (a frequency list for each class value) can be treated as a vector, and the cosine of the angle between the two part's vectors can be computed. The greater the angle the more different the two parts are. Apte et al. [19] pointed out that the early splitting attribute should strategically partition the instance space according to how different the decision characteristics are in the parts. This involves computing, for each candidate attribute, the quality measures for all other attributes in each of the parts. These are taken as vectors (measure-profiles) for each part and the angle between these measure-profile vectors are computed. This approach is most effective on problems that are quite heterogeneous in different regions of the instance space, such as the multiplexor problem.

4 Discretization of attributes

Discretization divides the interval $[\min_i \ .. \ \max_i]$ of the values of the numeric attribute X_i into a number of intervals. Each interval can then be treated as one value of a categorical attribute. Discretization is often considered a task for a domain expert and is then used as the background knowledge for the specific problem. However, often the expert is not available or unable to specify

exact intervals. With the help of an automatic procedure the task becomes easier. All non-analytic models are essentially expressed in terms of these discretized intervals.

Discretization of attributes can reduce the learning complexity and help to understand the dependency between the attributes and the target concept. Dougherty et al. [20] compare several techniques and observe from the many experiments that pre-discretization of numeric attributes often leads to a better decision tree. Quinlan [15] argues for the contrary, namely, by using the reduced information-gain approach, traditional binarization of numeric attributes within the process of node attribute selection is better. Aside from this debate (perhaps problem dependent), determining what intervals matter is in itself a practically useful exercise.

Some use simple minded discretization (equal interval size, equal instance population, one-dimensional clustering, etc.) which are easy to implement, but require the user to specify the number of intervals. As they ignore the class attribute, such quick discretization methods perform poorly in many situations. The ultimate goal is to discretize in such a way that the resulting model would perform best. This could be achieved by some form of wrapper style search, but the complexity would be compounded by the fact that the combination of possible discretization of all the numeric attributes is vast. The next best approach, which is generally practiced, is to discretize in such a way that the quality measure of the resultant discretized attribute is optimized.

Discretization is represented by a set of its interval boundary values called cut points. Each cut point is usually a mid-point value of some consecutive pair (magnitutide-wise) of distinct values of the attribute seen in the data. For common binary tree induction, the winning test for a numeric attribute is the $\leq$ test against one winning cut point value out of all possible such candidates (winning in the sense of the quality measure utilized). The number of candidates is therefore one less than the number of distinct values.

To arrive at an optimal cut point set, one can employ a greedy search strategy of successively adding an additional cut point which maximizes the quality measure of the resulting discretized attribute, until some stopping criteria (user given or determined by some other method) is met. Many methods compared in [20] use myopic measures. Robnik and Kononenko [21] have shown good results by using the ReliefF measure for this purpose.

The contextual merit measure can be used to obtain an optimal set of cut points as shown in [3]. Here, as the near neighbors are selected for the merit contribution, the actual value pairs and corresponding $1/D_{rs}^2$ value triplet, $\{x_{r,i}, x_{s,i}, 1/D_{rs}^2\}$, is collected for each (r,s) pair. These triplets are interpreted as spans which would contribute to the merit score by $1/D_{rs}^2$ if a cut point is placed between the two end values. A straightforward dynamic program then develops the best cut point

sets in increasing sizes. The score attained by the cuts closely approximates the contextual merit of the resulting discretized attribute. As the number of cut points increases, the total attained score increases in a manner typical of an efficiency curve. A heuristic procedure is then used to pick the number of cut points around the knee of this curve. After the initial round of such discretization for all numeric attributes, a few refining rounds are taken, where a single discretized attribute is freed back to its numeric value, one at a time, and re-discretized keeping all other numeric attributes with discretized values.

A challenging problem is to obtain a better understanding for how many discretized intervals are "optimal". As was pointed out above, it is impractical to actually try out different numbers of cut points in models and measure their performances, for even the combinations of different numbers of cuts for all numeric attributes are too numerous, let alone the combinations of actual cut point sets.

5 Conclusion

We presented principal issues and techniques in determining which attribute(s) is important for the purpose of modelling from the machine learning perspective. It has been shown that various approaches approximate the solution either by a search or by computing some quality measure for an attribute. Depending on how the quality measure is defined, a trade-off between computational complexity and degree of robustness is made. For some applications a fast myopic measure suffices, while for others non-myopic contextual quality measures are worth the increased computation.

All the quality measures that have been proposed represent an average or overall "goodness" measure over the entire instance space, regardless of whether the measure is myopic or not. If some attributes are "good" or important in only some region of the instance space, they may not achieve an overall measure high enough to be selected. However, a good set of attributes may be a set of just such attributes which together contribute to a superior model. Even a wrapper style search would not find such combinations for the search is greedy by practical necessity.

Since the attribute selection ultimately determines the quality of the ensuing model as well as the computational burden to develop it, we expect much research activity in this area.

References

[1] Quinlan J.R., *C4.5 programs for machine learning*, Morgan Kaufmann, 1993.

[2] Breiman L., Friedman J.H., Olshen R.A. & Stone C.J., *Classification and Regression Trees*, Wadsworth International Group, 1984.

[3] Hong S.J., "Use of contextual information for feature ranking and discretization", *IBM Research Report, RC-19664, July, 1994*, a revised version to appear in *IEEE-TKDE*, Vol. 9, No. 4, 1997.

[4] Kira K. & Rendell L., "A practical approach to feature selection", *Proc. Int. Conf. on Machine Learning, ICML-92*, pp. 249-256, 1992.

[5] Kononenko I., "Estimating attributes: Analysis and extensions of RELIEF", *Proc. European Conf. on Machine Learning*, pp. 171-182, 1994.

[6] Kononenko I., Šimec E. & Robnik M., "Overcoming the Myopia of inductive learning algorithms with ReliefF", *Applied Intelligence*, Vol. 7, pp. 39-55, 1997.

[7] Weiss S. and Indurkhya N., "Rule-based regression", *Proc. Intern. Joint Conf. on Artificial Intelligence, IJCAI-93*, pp. 1072-1078, 1993.

[8] Robnik-Šikonja M. & Kononenko I., "An adaptation of Relief for attribute estimation in regression", *Proc. Int. Conf. on Machine Learning, ICML-97*, Nashville, July 1997.

[9] White A.P. and Liu W.Z., "Bias in information-based measures in decision tree induction", *Machine Learning*, Vol 15., pp. 321-329, 1994.

[10] Kononenko I., "On biases when estimating multivalued attributes", *Proc. Intern. Joint Conf. on Artificial Intelligence, IJCAI-95*, pp. 1034-1040, 1995.

[11] Hong S.J., Hosking J.R.M. & Winograd S., "Use of randomization to normalize feature merits", *Information, Statistics and Induction Science, Proc. of ISIS-96*, pp. 10-19, 1996.

[12] John G., Kohavi R. & Pfleger K., "Irrelevant features and the subset selection problem", *Proc. 11th International Conf. on Machine Learning, ICML-94*, pp. 121-129, 1994.

[13] Kohavi R. & Sommerfield D., "Feature subset selection using the wrapper method: overfitting and dynamic search space topology", *Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining, KDD-95*, pp. 192-197, 1995.

[14] Cherkauer K.J., Shavlik J.W., "Growing simpler decision trees to facilitate knowledge discovery", *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, KDD-96*, pp. 315-318, 1996.

[15] Quinlan J.R., "Improved use of continuous attributes in C4.5", *J. of Artificial Intelligence Research*, Vol. 4, pp. 79-90, 1996.

[16] Zhou X.M., & Dillon T.S., "Statistical-heuristic feature selection criterion for decision tree induction", *IEEE-PAMI*, Vol. 13, No. 8, 1991.

[17] Smyth P. & Goodman R.M., "An information Theoretic approach to rule induction from databases", *IEEE-TKDE*, Vol. 4, No. 4, pp. 301-316, 1992.

[18] Fayyad U. & Irani K., "The attribute selection problem in decision tree generation", *Proc. 10th Nat. Conf. on Artificial Intelligence, AAAI-92*, pp. 104-110, 1992.

[19] Apte C., Hong S.J., Hosking J.R.M., Lepre J., Pednault E.& Rosen B., "Decomposition of heterogeneous classification problems", *IBM Research Report, RC-20514*, March 1996, To apear in *Proc. 2nd Int. Conf. on Intelligent Data Analysis, IDA-97*.

[20] Dougherty J., Kohavi R. & Sahami M., "Supervised and unsupervised discretization of continuous features", *Proc. of Twelfth Int. Conf. on Machine Learning, ICML-95*, pp. 194-202, 1995.

[21] Robnik M. & Kononenko I., "Discretization of continuous attributes using ReliefF", *Proc. Electr. and Comp. Sci. Conf., ERK-95*, pp. 149-152, 1995.