# An adaptation of Relief for attribute estimation in regression

**Marko Robnik-Šikonja**
University of Ljubljana,
Faculty of Computer and Information Science,
Tržaška 25, 1001 Ljubljana, Slovenia,
Marko.Robnik@fri.uni-lj.si

**Igor Kononenko**
University of Ljubljana,
Faculty of Computer and Information Science,
Tržaška 25, 1001 Ljubljana, Slovenia,
Igor.Kononenko@fri.uni-lj.si

## Abstract

Heuristic measures for estimating the quality of attributes mostly assume the independence of attributes so in domains with strong dependencies between attributes their performance is poor. Relief and its extension ReliefF are capable of correctly estimating the quality of attributes in classification problems with strong dependencies between attributes. By exploiting local information provided by different contexts they provide a global view. We present the analysis of ReliefF which lead us to its adaptation to regression (continuous class) problems. The experiments on artificial and real-world data sets show that Regressional ReliefF correctly estimates the quality of attributes in various conditions, and can be used for non-myopic learning of the regression trees. Regressional ReliefF and ReliefF provide a unified view on estimating the attribute quality in regression and classification.

## 1 Introduction

The majority of current propositional inductive learning systems predict discrete class. They can also solve the regression (also called continuous class) problems by discretizing the prediction (class) in advance. This approach is often inappropriate. Regression learning systems (also called function learning systems), e.g., CART (Breiman et al., 1984), Retis (Karalič, 1992), M5 (Quinlan, 1993), directly predict continuous value.

The problem of estimating the quality of attributes seems to be an important issue in both classification and regression and in machine learning in general (e.g., feature selection, constructive induction). Heuristic measures for estimating the attribute's quality mostly assume the independence of attributes, e.g., information gain (Hunt et al., 1966), Gini index (Breiman et al., 1984), distance measure (Mantaras, 1989), and j-measure (Smyth and Goodman, 1990) for discrete class and the mean squared and the mean absolute error (Breiman et al., 1984) for regression. They are therefore less appropriate in domains with strong dependencies between attributes.

Relief (Kira and Rendell, 1992) and its extension ReliefF (Kononenko, 1994) are aware of the contextual information and can correctly estimate the quality of attributes in classification problems with strong dependencies between attributes. Similar approaches are the contextual merit (Hong, 1994) and the geometrical approach (Elomaa and Ukkonen, 1994). We present the analysis of ReliefF which lead us to adapt it to regression problems.

Several other researchers have investigated the use of local information and profited from being aware of it (Domingos, 1997; Atkeson et al., 1996; Friedman, 1994). The approach described in this paper is not directly comparable to theirs and provides a different perspective.

Relief has commonly been viewed as a feature selection method that is applied in a prepossessing step before the model is learned, however it has recently also been used during the learning process to select splits in the building phase of decision tree (Kononenko et al., 1997). We experimented also with similar use in regression trees.

In the next Section we present and analyze the novel RReliefF (Regressional ReliefF) algorithm for estimating the quality of attributes in regression problems. Section 3 describes experiments with estimation of attributes under various conditions, and Section 4 describes our use of RReliefF in learning of regression trees. The last Section summarizes and gives guidelines for further work.

*Algorithm* Relief
*Input:* for each training instance a vector of attribute values and the class value
*Output:* the vector W of estimations of the qualities of attributes

1. set all weights W[A] := 0.0;
2. **for** i := 1 **to** m **do begin**
3. randomly select an instance R;
4. find nearest hit H and nearest miss M;
5. **for** A := 1 **to** #all_attributes **do**
6. W[A] := W[A] - diff(A,R,H)/m + diff(A,R,M)/m;
7. **end**;

Figure 1: The basic Relief algorithm

## 2 RReliefF

### 2.1 Relief and ReliefF for classification

The key idea of the original Relief algorithm (Kira and Rendell, 1992), given in Figure 1, is to estimate the quality of attributes according to how well their values distinguish between the instances that are near to each other. For that purpose, given a randomly selected instance R (line 3), Relief searches for its two nearest neighbors: one from the same class, called *nearest hit* H, and the other from a different class, called *nearest miss* M (line 4). It updates the quality estimation W[A] for all the attributes A depending on their values for R, M, and H (lines 5 and 6). The process is repeated for $m$ times, where $m$ is a user-defined parameter.

Function $diff(Attribute, Instance1, Instance2)$ calculates the difference between the values of Attribute for two instances. For discrete attributes it is defined as:

$$diff(A, I_1, I_2) = \begin{cases} 0 & ; value(A, I_1) = value(A, I_2) \\ 1 & ; otherwise \end{cases} \quad (1)$$

and for continuous attributes as:

$$diff(A, I_1, I_2) = \frac{|value(A, I_1) - value(A, I_2)|}{max(A) - min(A)} \quad (2)$$

The function $diff$ is used also for calculating the distance between instances to find the nearest neighbors. The total distance is simply the sum of distances over all attributes. Relief's estimate $W[A]$ of the quality of attribute $A$ is an approximation of the following difference of probabilities (Kononenko, 1994):

$$W[A] = P(\text{diff. value of A}|\text{nearest inst. from diff. class}) - P(\text{diff. value of A}|\text{nearest inst. from same class}) \quad (3)$$

The complexity of Relief for $n$ training instances and $A$ attributes is $O(m \times n \times A)$. The original Relief can deal with discrete and continuous attributes. However, it can not deal with incomplete data and is limited to two-class problems. Kononenko (1994) has shown that Relief's estimates are strongly related to impurity functions and developed an extension called ReliefF that is more robust, can tolerate incomplete and noisy data and can manage multiclass problems. One difference from original Relief, interesting also for regression, is that, instead of one nearest hit and one nearest miss, ReliefF uses $k$ nearest hits and misses and averages their contribution to $W[A]$.

The power of Relief is its ability to exploit information locally, taking the context into account, but still to provide the global view.

### 2.2 RReliefF for regression

In regression problems the predicted value (class) is continuous, 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 the predicted values of two instances are different. This probability can be modeled with the relative distance between the predicted (class) values of the two instances.

Still, to estimate W[A] in (3), the information about the sign of each contributed term is missing. In the following derivation we reformulate (3), so that it can be directly evaluated using the probability of the predicted values of two instances being different. If we rewrite

$$P_{diffA} = P(\text{diff. value of A}|\text{nearest instances}) \quad (4)$$

$$P_{diffC} = P(\text{different prediction}|\text{nearest instances}) \quad (5)$$

and

$$P_{diffC|diffA} = P(\text{different prediction}|$$
$$|\text{different value of A and nearest instances}) \quad (6)$$

we obtain from (3) using Bayes rule:

$$W[A] = \frac{P_{diffC|diffA}P_{diffA}}{P_{diffC}} - \frac{(1-P_{diffC|diffA})P_{diffA}}{1-P_{diffC}} \quad (7)$$

Therefore, we can estimate $W[A]$ by approximating terms defined by Equations 4, 5 and 6. This can be done by the algorithm on Figure 2. The weights for different prediction (class), different attribute, and different prediction & different attribute are collected in $N_{dC}$, $N_{dA}[A]$, and $N_{dC\&dA}[A]$, respectively. The final estimation of each attribute $W[A]$ (Equation 7) is computed in lines 14 and 15.

Term $d(i,j)$ in Figure 2 (lines 6, 8 and 10) is used to take into account the distance between the two instances $R_i$ and $I_j$. Closer instances should have greater influence, so we exponentially decreased the influence of instance $I_j$ with the distance from the given instance $R_i$:

$$d(i,j) = \frac{d_1(i,j)}{\sum_{l=1}^{k} d_1(i,l)} \quad \text{and} \quad (8)$$
$$d_1(i,j) = e^{-\left(\frac{rank(R_i,I_j)}{\sigma}\right)^2}$$

where $rank(R_i, I_j)$ is the rank of instance $I_j$ in a sequence of instances ordered by the distance from $R_i$ and $\sigma$ is a user defined parameter. We also experimented using the constant influence of all $k$ nearest instances $I_j$ around instance $R_i$ by taking $d_1(i,j) = 1/k$, but the results did not differ significantly. Since we consider the former to be more general we have chosen it for this presentation.

Note that the time complexity of RReliefF is the same as that of original Relief, i.e., $O(m \times n \times A)$. The most complex operation within the main **for** loop is the selection of **k** nearest instances $I_j$. For it we have to compute the distances from $I_j$ to $R$, which can be done in $O(n \times A)$ steps for $n$ instances. This is the most complex operation. $O(n)$ is needed to build a heap, from which $k$ nearest instances are extracted in $O(k \log n)$ steps, but this is less than $O(n \times A)$.

## 3 Estimating the attributes

In this Section we examine the ability of RReliefF to recognize and rank important attributes and in the next Section we use these abilities in learning regression trees.

We compare the estimates of RReliefF with the mean squared error (MSE) as a measure of the attribute's quality (Breiman et al., 1984). This measure is standard in regression tree systems. By this criterion the best attribute is the one which minimizes the equation:

$$MSE(A) = p_L \cdot s(t_L) + p_R \cdot s(t_R), \quad (9)$$

where $t_L$ and $t_R$ are the subsets of cases that go left and right, respectively, by the split based on $A$, and $p_L$ and $p_R$ are the proportions of cases that go left and right. $s(t)$ is the standard deviation of the predicted values $c_i$ of cases in the subset $t$ :

$$s(t) = \sqrt{\frac{1}{N(t)} \sum_{i=1}^{N(t)} (c_i - \overline{c(t)})^2}. \quad (10)$$

The minimum of (9) among all possible splits for an attribute is considered its quality estimate and is given in the results below.

We have used several families of artificial data sets to check the behavior of RReliefF in different circumstances.

**FRACTION:** each domain contains continuous attributes with values from 0 to 1. The predicted value is the fractional part of the sum of $I$ important attributes: $C = \sum_{j=1}^{I} A_j - \lfloor \sum_{j=1}^{I} A_j \rfloor$. These domains are floating point generalizations of parity concept of order $I$, i.e., domains with highly dependent pure continuous attributes.

**MODULO-8:** domains are described by a set of attributes, value of each attribute is an integer value in the range 0-7. Half of the attributes are treated as discrete and half as continuous; each continuous attribute is exact match of one of the discrete attributes. The predicted value is the sum of the $I$ important attributes by modulo 8; $C = (\sum_{i=j}^{I} A_j) \bmod 8$. These domains are integer generalizations of parity concept (which is the sum by modulo 2) of order $I$. They shall show how well RReliefF recognizes highly dependent attributes and how it ranks discrete and continuous attributes of equal importance.

**PARITY:** each domain consists of discrete, Boolean attributes. The $I$ informative attributes define parity concept: if their parity bit is 0, the predicted value is set to a random number between 0 and 0.5, otherwise it is randomly chosen to be between 0.5 and 1.

$$C = \begin{cases} rand(0, 0.5) & ; \quad (\sum_{j=1}^{I} A_j) \bmod 2 = 0 \\ rand(0.5, 1) & ; \quad (\sum_{j=1}^{I} A_j) \bmod 2 = 1 \end{cases}$$

*Algorithm* RReliefF
*Input:* for each training instance a vector of attribute values **x** and the predicted value $\tau(\mathbf{x})$
*Output:* the vector W of estimations of the qualities of attributes

1. set all $N_{dC}$, $N_{dA}[A]$, $N_{dC\&dA}[A]$, $W[A]$ to 0;
2. **for** i := 1 **to** m **do begin**
3. randomly select instance $R_i$;
4. select **k** instances $I_j$ nearest to $R_i$;
5. **for** j := 1 **to** k **do begin**
6. $N_{dC} := N_{dC} + |f(R_i) - f(I_j)| \cdot d(i,j)$;
7. **for** A := 1 **to** #all_attributes **do begin**
8. $N_{dA}[A] := N_{dA}[A] + diff(A, R_i, I_j) \cdot d(i,j)$;
9. $N_{dC\&dA}[A] := N_{dC\&dA}[A] + |f(R_i) - f(I_j)| \cdot$
10. $diff(A, R_i, I_j) \cdot d(i,j)$;
11. **end**;
12. **end**;
13. **end**;
14. **for** A := 1 **to** #all_attributes **do**
15. $W[A] := N_{dC\&dA}[A]/N_{dC}$ - $(N_{dA}[A] - N_{dC\&dA}[A])/(m - N_{dC})$;

Figure 2: Pseudo code of RReliefF (Regressional ReliefF)

These concepts present blurred versions of the parity concept (of order $I$). They shall test the behavior of RReliefF on discrete attributes.

**LINEAR:** the domain is described by continuous attributes with values chosen randomly between 0 and 1; the predicted value is computed by the following linear formula: $C = A_1 - 2A_2 + 3A_3 - 3A_4$. We have included this domain to compare the performance of RReliefF with that of the MSE, which is known to recognize linear dependencies.

**COSINUS:** this domain has continuous attributes with values from 0 to 1; the prediction is computed as follows: $C = (-2A_2 + 3A_3)\cos(4\pi A_1)$. It shall show the ability of the heuristics to handle non-linear dependencies.

In experiments below we have used $I = \{2, 3, 4\}$ important attributes. Each of the domains has also some irrelevant (random) attributes with values in the same range as the important attributes.

For each domain we have generated $N$ examples and computed the estimates as the average of 10-fold cross validation. With this we collected enough data to eliminate any probabilistic effect caused by the random selection of instances in RReliefF. We also evaluated the significance of differences between the estimates with paired t-test (at 0.05 level).

In all experiments RReliefF was run with the same default set of parameters (constant $m$ in main loop = 250, k-nearest = 200, $\sigma = 20$ (see (8))).

### 3.1 Varying the number of examples

First we investigated how the number of the available examples influences the estimates. We have generated domains with $I = \{2, 3, 4\}$ important attributes and added them $R = 10 - I$ random attributes with values in the same range, so that the total number of the attributes was 10 (LINEAR and COSINUS have $I$ fixed to 4 and 3, respectively). We cross-validated the estimates of the attributes in altogether 11 data sets, varying the size of the data set from 10 to 1000 in steps of 10 examples.

Figure 3 shows the dependence for FRACTION domain with $I = 2$ important attributes. *Note that RReliefF gives higher scores to better attributes, while MSE does the opposite.* On the tope we can see that with small number of examples (below 50) a random attribute with the highest estimate (best random) is estimated as better than the two important attributes (I1 and I2). Increasing the number of the examples to 100 was enough that the estimates of the important attributes were significantly better than the estimate of the best random attribute. The bottom of Figure 3 shows that MSE is incapable of distinguishing between important and random attributes. Since there are 8 random attributes the one with the lowest estimate is mostly esti-

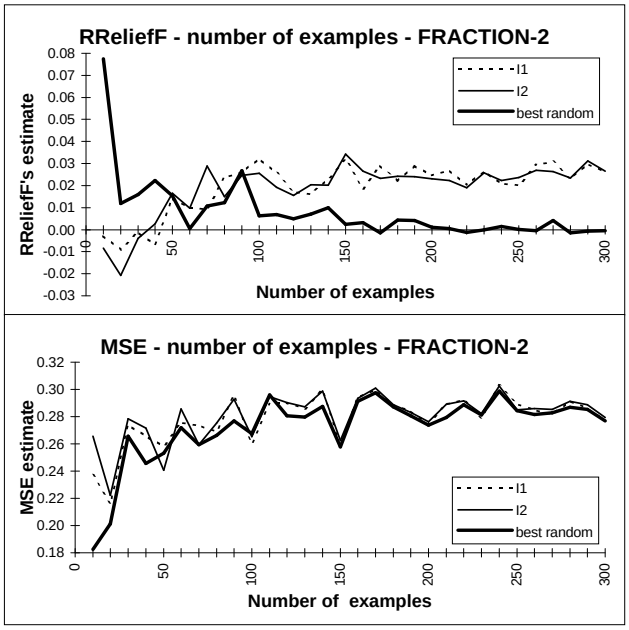


Figure 3: Varying the number of examples in FRACTION domain with 2 important and 8 random attributes. Note that RReliefF gives higher scores to better attributes, while the MSE does the opposite.

mated as better than I1 and I2.

The behaviors of RReliefF and MSE are similar on other FRACTION, MODULO and PARITY data sets. Due to the lack of space we will omit other graphs and will only comment the summary of the results presented in the Table 1. The two numbers given are the limiting number of examples that were needed that the estimates of the important attributes which were estimated as the worst ($I_w$) and the best ($I_b$) between important attributes were significantly better than the estimates of the attribute that was estimated as the best between the random attributes. The '-' sign means that the estimator did not succeed to significantly distinguish between the two groups of the attributes.

We can observe that the number of the examples needed is increasing with the increasing complexity (number of important attributes) of each problem. While the PARITY, FRACTION and MODULO-8 are solvable for RReliefF, MSE is completely lost there. MODULO-8-4 (with 4 important attributes) is too difficult even for RReliefF. It seems that 1000 examples is not enough for a problem of such complexity, namely the complexity grows exponentially here: the number of peaks in the instance space for MODULO-m-p domain is $m^p$. This was confirmed by an additional experiment with 8000 examples where RReliefF succeeded to separate the two groups of examples.

Table 1: Results of varying the number of the examples. Numbers present the number of examples required to ensure that relevant attributes are ranked significantly higher than irrelevant attributes.

| | | RReliefF | | MSE | |
|---|---|---|---|---|---|
| domain | I | $I_w$ | $I_b$ | $I_w$ | $I_b$ |
| FRACTION | 2 | 100 | 100 | - | - |
| | 3 | 300 | 220 | - | - |
| | 4 | 950 | 690 | - | - |
| MODULO-8 | 2+2 | 80 | 70 | - | - |
| | 3+3 | 370 | 230 | - | - |
| | 4+4 | - | - | - | - |
| PARITY | 2 | 50 | 50 | - | - |
| | 3 | 100 | 100 | - | - |
| | 4 | 280 | 220 | - | - |
| LINEAR | 4 | - | 10 | 340 | 20 |
| COSINUS | 3 | - | 50 | 490 | 90 |

Also interesting in the MODULO problem is that the important discrete attributes are considered better than their continuous counterparts. We can understand this if we consider the behavior of $diff$ function (see (1) and (2)). Let's take two cases with 2 and 5 being their values of attribute $A_i$, respectively. If $A_i$ is the discrete attribute, the value of $diff(A_i, 2, 5) = 1$, since the two categorical values are different. If $A_i$ is the continuous attribute, $diff(A_i, 2, 5) = \frac{|2-5|}{7} \approx 0.43$. So, with this form of $diff$ function continuous attributes are underestimated. We can overcome this problem with the ramp function as proposed by (Hong, 1994). It can be defined as a generalization of $diff$ function for the continuous attributes:

$$diff(A, I_1, I_2) = \begin{cases} 0 & ; d \leq t_{eq} \\ 1 & ; d > t_{diff} \\ \frac{d - t_{eq}}{t_{diff} - t_{eq}} & ; t_{eq} < d \leq t_{diff} \end{cases} \quad (11)$$

where $d = |value(A, I_1) - value(A, I_2)|$ presents the distance between the attribute values of the two instances, and $t_{eq}$ and $t_{diff}$ are two user definable threshold values; $t_{eq}$ is the maximum distance between the two attribute values to still consider them equal, and $t_{diff}$ is the minimum distance between attribute values to still consider them different. We have omitted the use of the ramp function from this presentation, as it complicates the basic idea, but the tests with sensible default thresholds have shown that continuous attributes are no longer underestimated.

The results for LINEAR and COSINUS domains show that separating the worst important attribute ($A_1$ and $A_2$, respectively) from the best random attribute is not easy neither for RReliefF nor for MSE, but MSE was better. RRe-

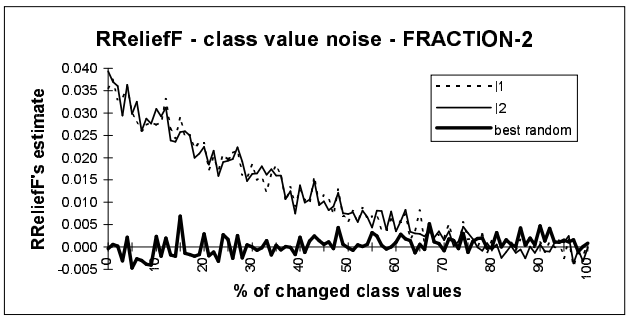


Figure 4: RReliefF's estimates when the noise is added to the predicted value. in FRACTION domain with 2 important attributes.

liefF could distinguish the two attributes significantly with more than 100 examples, but occasionally some peak in the estimation of the random attributes caused the t-value to fall below the significance threshold. MSE could also mostly distinguish the two groups, but lower variation gives it a slight advantage. When separating the best important attribute from random attributes, both RReliefF and MSE were successful but RReliefF needed less examples. This probably compensates the negative result for RReliefF with the worst important attribute (e.g., in the regression tree learning a single best attribute is selected in each node). The difference between the estimators is also in ranking the attributes by importance in COSINUS domain. The correct decreasing order replicated by RReliefF is $A_1$, $A_3$, $A_2$ , while MSE orders them: $A_3$, $A_2$, $A_1$, as it is unable to detect non-linear dependencies.

LINEAR and COSINUS domains show that on relatively simple relations the performance of RReliefF and MSE are comparable.

We have also tested other types of non-linear dependencies between the attributes (logarithmic, exponential, polynomial, trigonometric, ...) and RReliefF has been always superior to MSE.

### 3.2 Adding noise by changing the predicted value

We checked the robustness of RReliefF to noise by using the same setting as before (data sets with $I = \{2, 3, 4\}$ important attributes, altogether 10 attributes), but with the number of examples fixed to 1000. We added noise to the data sets by changing certain percent of predicted values to a random value in the same range as the correct values. We were varying the noise from 0 to 100%.

The Figure 4 shows the dependence for FRACTION domain with $I = 2$ important attributes.

We can see that RReliefF is robust, as it can significantly distinguish the worst important attribute from the best random even with 50 % of corrupted prediction values. The Table 2 summarizes the results for all the domains. The two columns give the maximal percentage of corrupted prediction values where the estimates of the worst important ($I_w$) and the best important attribute ($I_b$), respectively, were still significantly better than the estimates of the best random attribute. The '-' sign means that the estimator did not succeed to significantly distinguish between the two groups even without any noise.

Table 2: Results of adding noise to predicted values. Numbers tell us which percent of predicted values could we corrupt still to get significant differences in estimations.

| | | RReliefF | | MSE | |
|---|---|---|---|---|---|
| domain | I | $I_w$ | $I_b$ | $I_w$ | $I_b$ |
| FRACTION | 2 | 53 | 59 | - | - |
| | 3 | 16 | 35 | - | - |
| | 4 | 3 | 14 | - | - |
| MODULO-8 | 2+2 | 64 | 75 | - | - |
| | 3+3 | 52 | 70 | - | - |
| | 4+4 | - | - | - | - |
| PARITY | 2 | 66 | 70 | - | - |
| | 3 | 60 | 71 | - | - |
| | 4 | 50 | 67 | - | - |
| LINEAR | 4 | - | 66 | 50 | 85 |
| COSINUS | 3 | - | 46 | 36 | 63 |

### 3.3 Adding more random attributes

RReliefF is unlike MSE sensitive to random attributes. We tested this sensitivity with similar settings as before (data sets with $I = \{2, 3, 4\}$ important attributes, the number of examples fixed to 1000), and added from 1 to 200 random attributes.

The Figure 5 shows the dependence for FRACTION domain with $I = 2$ important attributes. We can see that RReliefF is quite tolerant to this kind of noise as even 70 random attributes does not prevent it to assign significantly different estimates to the worst informative and the best random attribute.

Table 3 summarizes the results. Since MSE estimates each attribute separately from the others it is not sensitive to this kind of noise and we did not include it into this experiment. The two columns give the maximal number of random attributes that could be added before the estimates of the worst ($I_w$) and the best important attribute ($I_b$), respectively, were no more significantly better than the estimates

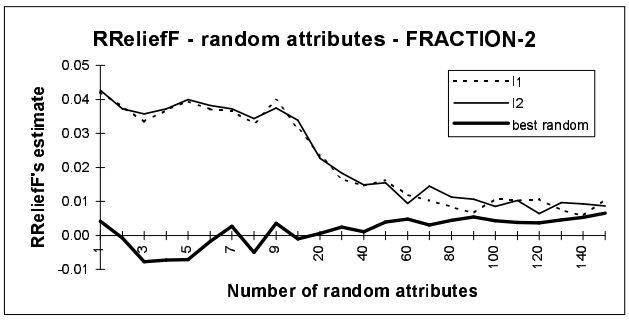


Figure 5: RReliefF's estimates when random attributes are added in FRACTION domain with 2 important attributes.

Table 3: Results of adding random attributes. Numbers tell us how many random attributes could we add still to significantly differentiate the worst and the best important attribute from the best random attribute

| domain | I | $I_{worst}$ | $I_{best}$ |
|---|---|---|---|
| FRACTION | 2 | 70 | 80 |
| | 3 | 10 | 20 |
| | 4 | 4 | 5 |
| MODULO-8 | 2+2 | 50+50 | 100+100 |
| | 3+3 | 7+7 | 20+20 |
| | 4+4 | - | - |
| PARITY | 2 | $>200$ | $>200$ |
| | 3 | 50 | 70 |
| | 4 | 10 | 20 |
| LINEAR | 4 | - | $>200$ |
| COSINUS | 3 | - | $>200$ |

of the best random attribute. The '-' sign means that the estimator did not succeed to significantly distinguish between the two groups even with a single random attribute.

## 4 Building regression trees

We have developed a learning system which builds binary regression and model trees (as named by (Quinlan, 1993)) by recursively splitting training examples based on the values of attributes. The attribute in each node is selected according to the estimates of the attributes' quality by RReliefF or by MSE (9). These estimates are computed on the subset of the examples that reach current node. Such use of ReliefF on classification problems was shown to be sensible and can significantly outperform impurity measures (Kononenko et al., 1997).

We have run our system with two sets of parameters and procedures. They are named according to the type of models used in the leaves.

**Point** mode is similar to CART (Breiman et al., 1984) and uses the average prediction (class) value of the examples in each leaf node as the predictor. Instead of cost complexity pruning of CART which demands the cross validation or separate set of examples for setting its complexity parameter we were using the Retis' pruning with $m$-estimate of probability (Karalič, 1992) which produces comparable or better results.

**Linear** mode is similar to M5 (Quinlan, 1993), and uses pruned linear models in each leaf node as the predictor. We were using the same procedures for pruning and smoothing of the trees as M5.

The trees with linear models offer greater expressive power and mostly perform better on real world problems which often contain some form of near linear dependency (Karalič, 1992; Quinlan, 1993). The problem of overfitting the data is relieved by the pruning procedure employed in M5 which can reduce the linear formula to the constant term if there is not enough evidential support for it.

Each of the modes uses the same default set of parameters for growing and pruning of the tree. We used two stopping criteria, namely the minimal number of cases in the leaf (5) or the minimal purity of a leaf (proportion of the root's standard deviation ($s$) in the leaf = 8%).

We ran our system on the artificial data sets and on domains with continuous prediction value from UCI (Murphy and Aha, 1995). Artificial data sets were used as described above (11 data sets, each consisting of 10 attributes - 2, 3 or 4 important, the rest are random, and containing 1000 examples). For each domain we collected the results as the average of 10 fold cross-validation. We present results in Table 4.

We compare the relative mean squared error of the predictors $\phi$:

$$RE_t(\phi) = \frac{R_t(\phi)}{R_t(\mu)}, \text{ where } R_t(\tau) = \frac{1}{N_t}\sum_{i=1}^{N_t}(c_i - \tau(x_i))^2. \quad (12)$$

The $i^{th}$ example is written as the ordered pair $(c_i, x_i)$, where $x_i$ is the vector of attribute values. $\tau(x_i)$ is the value predicted by $\tau$, and $\mu$ is a predictor which always returns the mean value of the prediction values. Sensible predictors have $RE(\phi) < 1$.

Besides the error we included also the measure of complexity $C$ of the tree. C is the number of all the occurrences of all the attributes anywhere in the tree plus the

Table 4: Relative error and complexity of the regression and model trees with RReliefF and MSE as the estimators of the quality of the attributes.

| | linear | | | | | point | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | RReliefF | | MSE | | | RReliefF | | MSE | | |
| domain | $RE$ | $C$ | $RE$ | $C$ | S | $RE$ | $C$ | $RE$ | $C$ | S |
| Fraction-2 | .34 | 112 | .86 | 268 | + | .52 | 87 | 1.17 | 160 | + |
| Fraction-3 | .73 | 285 | 1.08 | 440 | + | 1.05 | 174 | 1.62 | 240 | + |
| Fraction-4 | 1.05 | 387 | 1.10 | 392 | 0 | 1.51 | 250 | 1.65 | 219 | 0 |
| Modulo-8-2 | .22 | 98 | .77 | 329 | + | .06 | 58 | .81 | 195 | + |
| Modulo-8-3 | .58 | 345 | 1.08 | 436 | + | .59 | 166 | 1.58 | 251 | + |
| Modulo-8-4 | 1.05 | 380 | 1.07 | 439 | 0 | 1.52 | 253 | 1.52 | 259 | 0 |
| Parity-2 | .28 | 125 | .55 | 208 | + | .27 | 7 | .38 | 103 | 0 |
| Parity-3 | .31 | 94 | .82 | 236 | + | .27 | 15 | .61 | 213 | + |
| Parity-4 | .35 | 138 | .96 | 283 | + | .25 | 31 | .88 | 284 | + |
| Linear | .02 | 4 | .02 | 4 | 0 | .19 | 59 | .19 | 55 | 0 |
| Cosinus | .27 | 334 | .41 | 364 | + | .36 | 105 | .29 | 91 | 0 |
| Auto-mpg | .13 | 97 | .14 | 102 | 0 | .21 | 27 | .21 | 19 | 0 |
| Auto-price | .14 | 48 | .12 | 53 | 0 | .28 | 16 | .17 | 10 | 0 |
| CPU | .12 | 33 | .15 | 47 | 0 | .42 | 16 | .31 | 10 | 0 |
| Housing | .17 | 129 | .15 | 177 | 0 | .31 | 33 | .23 | 23 | 0 |
| Servo | .25 | 53 | .28 | 55 | 0 | .24 | 7 | .33 | 9 | 0 |

constant term in the leaves. The column labeled S presents the significance of the differences between RReliefF and MSE computed with paired t-test at 0.05 level of significance. 0 indicates that the differences are not significant at 0.05 level, '+' means that predictor with RReliefF is significantly better, and '-' implies the significantly better score by MSE.

In the linear mode (left side of Table 4) the predictor generated with RReliefF is on artificial data sets mostly significantly better than the predictor generated with MSE. On UCI databases the predictors are comparable, RReliefF being better on 3 data sets, and MSE on 2, but with insignificant differences. The complexity of the models induced by RReliefF is considerably smaller in most cases on both artificial and UCI data sets which indicates that RReliefF was more successful detecting the dependencies. With RReliefF the strong dependencies were mostly detected and expressed with the selection of the appropriate attributes in the upper part of the tree; the remaining dependencies were incorporated in the models used in the leaves of the tree. MSE was blind for strong dependencies and was splitting the examples solely to minimize their impurity (mean squared error) which prevented it to successfully model weaker dependencies with linear formulas in the leaves of the tree.

In the point mode RReliefF produces smaller and more accurate trees on problems with strong dependencies (FRACTION, MODULO-8 and PARITY), while on UCI databases MSE was better on 3 data sets and RReliefF on one data set, but the differences are not significant. MSE produced less complex trees on LINEAR, COSINUS and UCI datasets. The reason for this is similar to the opposite effect observed in the linear mode. Since there are probably no strong dependencies in these domains the ability to detect them did not bring any advantage to RReliefF. MSE which minimizes the squared error was favored by the stopping criterion (percentage of the standard deviation) and the pruning procedure which both rely on the estimation of the error. Similar effect was observed in classification (Brodley, 1995; Lubinsky, 1995), namely it was empirically shown that the classification error is the most appropriate for selecting the splits near the leaves of the decision tree. This effect is hidden in the linear mode where the trees are smaller (but not less complex as can be seen from Table 4) and the linear models in the leaves play the role of minimizing the accuracy.

## 5 Conclusions

Our experiments show that RReliefF can discover strong dependencies between attributes, while in domains without such dependencies it performs the same as the mean squared error. It is also robust and noise tolerant. Its intrinsic contextual nature allows it to recognize contextual

attributes. From our experimental results we can conclude that learning regression trees with RReliefF is promising especially in combination with linear models in the leaves of the tree.

Both, RReliefF in regression and ReliefF in classification (Kononenko, 1994) are estimators of (7), which gives a unified view on the estimation of the quality of attributes for classification and regression. RReliefF's good performance and robustness indicate its appropriateness for feature selection.

As further work we are planning to use RReliefF in detection of context switch in incremental learning, and to guide the constructive induction process.

## References


Atkeson, C. G., Moore, A. W., and Schall, S. (1996). Locally weighted learning. Technical report, Georga Institute of Technology.

Breiman, L., Friedman, L., Olshen, R., and Stone, C. (1984). *Classification and regression trees*. Wadsworth Inc., Belmont, California.

Brodley, C. E. (1995). Automatic selection of split criterion during tree growing based on node location. In *Proceedings of the XII International Conference on Machine Learning*. Morgan Kaufmann.

Domingos, P. (1997). Context-sensitive feature selection for lazy learners. *Artificial Intelligence Review*. (to appear).

Elomaa, T. and Ukkonen, E. (1994). A geometric approach to feature selection. In De Raedt, L. and Bergadano, F., editors, *Proceedings of European Conference on Machine Learning*, pages 351–354. Springer Verlag.

Friedman, J. H. (1994). Flexible metric nearest neighbor classification. Technical report, Stanford University. available by anonomous ftp from playfair.stanford.edu pub/friedman.

Hong, S. J. (1994). Use of contextual information for feature ranking and discretization. Technical Report RC19664, IBM.

Hunt, E., Martin, J., and Stone, P. (1966). *Experiments in Induction*. Academic Press, New York.

Karalič, A. (1992). Employing linear regression in regression tree leaves. In Neumann, B., editor, *Proceedings of ECAI'92*, pages 440–441. John Wiley & Sons.

Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In D.Sleeman and P.Edwards, editors, *Proceedings of International Conference on Machine Learning*, pages 249–256. Morgan Kaufmann.

Kononenko, I. (1994). Estimating attributes: analysis and extensions of Relief. In De Raedt, L. and Bergadano, F., editors, *Machine Learning: ECML-94*, pages 171–182. Springer Verlag.

Kononenko, I., Šimec, E., and Robnik-Šikonja, M. (1997). Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence*, 7:39–55.

Lubinsky, D. J. (1995). Increasing the performance and consistency of classification trees by using the accuracy criterion at the leaves. In *Proceedings of the XII International Conference on Machine Learning*. Morgan Kaufmann.

Mantaras, R. (1989). ID3 revisited: A distance based criterion for attribute selection. In *Proceedings of Int. Symp. Methodologies for Intelligent Systems*, Charlotte, North Carolina, USA.

Murphy, P. and Aha, D. (1995). UCI repository of machine learning databases. (http://www.ics.uci.edu/ mlearn/MLRepository.html).

Quinlan, J. R. (1993). Combining instance-based and model-based learning. In *Proceedings of the X. International Conference on Machine Learning*, pages 236–243. Morgan Kaufmann.

Smyth, P. and Goodman, R. (1990). Rule induction using information theory. In Piatetsky-Shapiro, G. and Frawley, W., editors, *Knowledge Discovery in Databases*. MIT Press.