Experiments with Cost-Sensitive Feature Evaluation

Marko Robnik-Šikonja

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

Abstract. Many machine learning tasks contain feature evaluation as one of its important components. This work is concerned with attribute estimation in the problems where class distribution is unbalanced or the misclassification costs are unequal. We test some common attribute evaluation heuristics and propose their cost-sensitive adaptations. The new measures are tested on problems which can reveal their strengths and weaknesses.

1 Introduction

Feature (attribute) evaluation is an important component of many machine learning tasks, e.g., feature subset selection, constructive induction, decision and regression tree learning. In feature subset selection we need a reliable and practically efficient method for estimating the relevance of the features to the target concept, so that we can tackle learning problems where hundreds or thousands of potentially useful features describe each input object. In the constructive induction we try to enhance the power of the representation language and therefore introduce new features. Typically many candidate features are generated and again we have to evaluate them in order to decide which to retain and which to discard. While constructing a decision or regression tree the learning algorithm at each interior node selects the splitting rule (feature) which divides the problem space into subspaces. To select an appropriate splitting rule the learning algorithm has to evaluate several possibilities and decide which would partition the given problem most appropriately. Feature rankings and numerical estimates provided by evaluation algorithms are also an important source of information for a human understanding of certain tasks.

While historically the majority of machine learning research have been focused on reducing the classification error, there also exists a corpus of work on cost-sensitive classification where all errors are not equally important (see online bibliography [13]). In general, differences in importance of errors are handled through the cost of misclassification.

This work is concerned with the cost-sensitive attribute estimation and we assume that costs can be presented with the cost matrix C, where $C(i, j)$ is the

N. Lavrač et al. (Eds.): ECML 2003, LNAI 2837, pp. 325–336, 2003.

cost (could also be benefit) associated with prediction that an example belongs to the class τ_j where in fact it belongs to the class τ_i. The optimal prediction for an example $\mathbf{x}$ is the class τ_i that minimizes the expected loss:

$$L(\mathbf{x}, \tau_i) = \sum_{j=1}^{c} P(\tau_j|\mathbf{x})C(j,i),$$

where $P(\tau_j|\mathbf{x})$ is the probability of the class τ_j given example $\mathbf{x}$. The task of a learner is therefore to estimate these conditional probabilities. Feature evaluation measure need not be cost-sensitive for decision tree building, as shown by [1,3,4]. However, cost-sensitivity is a desired property of an algorithm which tries to rank or weight features according to their importance. Such ranking can be used for feature selection and feature weighting or shown to human experts to confirm/expand their domain knowledge. This is especially important in the fields like medicine where experts posses great deal of intuitive knowledge.

We will investigate some properties of attribute evaluation measures, like how do they behave on imbalanced data sets, scale with increasing number of classes, whether they detect (conditional) dependencies between attributes and to what extent they are cost-sensitive. We propose several cost-sensitive variants of common attribute evaluation measures and test them on artificial data sets which can reveal their properties.

Throughout the paper we use the notation where each learning instance $I_1, I_2, ..., I_n$ is represented by an ordered pair $(\mathbf{x}_k, \tau)$, where each vector of attributes $\mathbf{x}_k$ consists of individual attributes A_i, $i = 1, ..., a$, (a is the number of attributes) and is labeled with the target value τ_j, $j = 1, ..., c$ (c is the number of class values). Each discrete attribute A_i has values a_1 through a_{m_i}. Notation $I_{i,j}$ presents the value of j-th attribute for the instance I_i, and $I_{i,\tau}$ presents its class value. We write $p(a_{i,k})$ for the probability that the attribute A_i has value a_k, $p(\tau_k)$ is the probability of the class τ_k, and $p(\tau_j|a_{i,k})$ is the probability of the class τ_j conditioned by the attribute A_i having the value a_k.

The paper is organized into 5 sections. In Section 2 we review some selected attribute evaluation measures and in Section 3 we test how imbalanced class distribution affects their performance. In Section 4 we describe how to extend these measures to use the information from the cost matrix and in Section 5 we evaluate the proposed extensions. Section 6 concludes the work.

2 Attribute Evaluation Measures

The problem of attribute estimation has received much attention in the literature. There are several measures for estimating attributes' quality. In classification problems these are e.g., Gini index [1], Gain ratio [11], Relief [5], ReliefF [6], MDL [7], and DKM [2].

Except Relief and ReliefF all these attribute evaluation measures are impurity based, meaning that they measure impurity of the class value distribution. They assume the conditional (upon the class) independence of the attributes,

evaluate each attribute separately and not take the context of other attributes into account. In problems which possibly involve much feature interactions these measures are not appropriate. Relief and ReliefF do not make this assumption and can correctly evaluate attributes in problems with strong dependencies between the attributes. We will first present measures based on impurity followed by ReliefF.

2.1 Impurity Based Measures

These measures evaluate each attribute separately by measuring impurity of the splits resulting from partition of the learning instances according to the values of the evaluated attribute. The general form of all impurity based measures is:

$$M(A_i) = i(\tau) - \sum_{j=1}^{a_{m_i}} p(a_{i,j}) i(\tau|a_{i,j}) \ ,$$

where $i(\tau)$ is the impurity of class values before the split, and $i(\tau|a_{i,k})$ is the impurity of class values after the split on $A_i = a_{k,j}$. By subtracting weighted impurity of the splits from the impurity of unpartitioned instances we measure gain in the purity of class values resulting from the split. Larger values of $M(A_i)$ imply pure splits and therefore good attributes. We cannot directly apply these measures to numerical attributes, but we can use any of the number of discretization techniques first and then evaluate discretized attributes. We consider three measures as examples of impurity based attribute evaluation.

Gain Ratio [11] is implemented in C4.5 program and is the most often used impurity based measure. It is defined as

$$GR(A_i) = \frac{\sum_{i=1}^{c} p(\tau_i) \log p(\tau_i) - \sum_{j=1}^{a_{m_i}} \sum_{i=1}^{c} p(\tau_i|a_{i,j}) \log p(\tau_i|a_{i,j})}{\sum_{j=1}^{a_{m_i}} p(a_{i,j}) \log p(a_{i,j})} \ . \qquad (1)$$

Its gain part tries to maximize the difference of entropy (which serves as impurity function) before and after the split. To prevent excessive bias towards multiple small splits the gain is normalized with the attribute's entropy.

DKM [2] has the following form of impurity function:

$$i(\tau) = 2\sqrt{p(\tau_{max})(1 - p(\tau_{max}))} \ , \text{ where } \ p(\tau_{max}) = \max_{i=1}^{c} p(\tau_i) \qquad (2)$$

is the most probable class value (the one which labels the split). Drummond and Holte [3] have shown that for binary attributes this function is invariant to changes in the proportion of different classes, i.e. it is cost-insensitive.

MDL is based on Minimum Description Length principle and measures the quality of attributes as their ability to compress the data. The difference in coding length before and after the value of the attribute is revealed corresponds to the difference in impurity. Kononenko [7] has shown empirically that this criterion has the most appropriate bias concerning multi-valued attributes among a number of other impurity-based measures. It is defined as:

$$MDL(A_i) = \frac{1}{n}\left(\log_2 \binom{n}{n_{1.},...,n_{c.}} - \sum_{j=1}^{a_{m_i}} \log_2 \binom{n_{.j}}{n_{1j},...,n_{cj}} + \log_2 \binom{n+c+1}{c-1} - \sum_{j=1}^{a_{m_i}} \log_2 \binom{n_{.j}+c-1}{c-1}\right) \quad (3)$$

Here n is the number of training instances, $n_{i.}$ the number of training instances from class i, $n_{.j}$ the number of instances with j-th value of given attribute, and n_{ij} the number of instances from class i with j-th value of the attribute.

2.2 ReliefF

ReliefF algorithm [6,12] is an extension of Relief [5]. Unlike Relief it is not limited to two class problems, is more robust, and can deal with incomplete and noisy data. The idea of Relief and ReliefF is to evaluate partitioning power of attributes according to how well their values distinguish between similar instances. An attribute is given a high score if its values separate similar observations with different class and do not separate similar instances with the same class values. ReliefF samples the instance space, computes the differences between predictions and values of the attributes and forms a statistical measure for the proximity of the probability densities of the attribute and the class. Assigned quality evaluations are in the range $[-1, 1]$.

Pseudo code of the algorithm is given on Figure 1. ReliefF randomly selects an instance R_i (line 3), and then searches for k of its nearest neighbors from the same class, called nearest hits H (line 4), and also k nearest neighbors from each of the different classes, called nearest misses $M(t)$ (lines 5 and 6). It updates the quality estimation W_v for all attributes depending on their values for R_i, hits H and misses $M(t)$ (lines 7 and 8). The process is repeated for m times.

The update formula balances the contribution of hits and all the misses, and averages the result of m iterations:

$$W_v = W_v - \frac{1}{m}\mathrm{con}(A_v, R_i, H) + \frac{1}{m}\sum_{\substack{t=1 \\ t\neq R_{i,\tau}}}^{c} \frac{p(\tau_t)\mathrm{con}(A_v, R_i, M(C))}{1 - p(R_{i,\tau})} \quad (4)$$

where $\mathrm{con}(A_v, R_i, S)$ is the contribution of k nearest instances from the set S (hits or misses). In the simplest case it can be an average difference of attribute's

Algorithm ReliefF
Input: for each training instance a vector of attribute values and the class value
Output: the vector W with the evaluation for each attribute

1. **for** v = 1 **to** a **do** $W_v = 0$
2. **for** i = 1 **to** m **do begin**
3. randomly select an instance R_i
4. find k nearest hits H
5. **for** each class $t \neq R_{i,\tau}$ **do**
6. from class t find k nearest misses $M(t)$
7. **for** $v = 1$ **to** a **do**
8. update W_v according to Eq. (4)
9. **end**;

Fig. 1. Pseudo code of ReliefF algorithm.

values for k instances:

$$\mathrm{con}(A_v, R_i, S) = \frac{1}{k}\sum_{j=1}^{k} \mathrm{diff}(A_v, R_i, S_j) \ .$$

Here $\mathrm{diff}(A_v, I_t, I_u)$ denotes the difference between the values of the attribute A_v for two instances I_t and I_u. For nominal and numerical attributes, respectively, it can be defined as:

$$\mathrm{diff}(A_v, I_t, I_u) = \begin{Bmatrix} 0; & I_{t,v} = I_{u,v} \\ 1; & \text{otherwise} \end{Bmatrix}, \quad \mathrm{diff}(A_v, I_t, I_u) = \frac{|I_{t,v} - I_{u,v}|}{\max\limits_{l=1}^{n} I_{l,v} - \min\limits_{l=1}^{n} I_{l,v}} \ . \quad (5)$$

In this work we use exponentially decreasing weighted contribution of instances ranked by distance ($k = 70$, $\sigma = 20$ as recommended by [12]):

$$\mathrm{con}(A_v, R_i, S) = \frac{\sum_{j=1}^{k} \mathrm{diff}(A_v, R_i, S_j) e^{-\left(\frac{\mathrm{rank}(R_i, S_j)}{\sigma}\right)^2}}{\sum_{l=1}^{k} e^{-\left(\frac{\mathrm{rank}(R_i, S_l)}{\sigma}\right)^2}} \ .$$

In (4) the contribution of each misses' class is weighted with the prior probability of that class $p(\tau_t)$. Since the contributions of hits and misses in each step should be in $[0, 1]$ and also symmetric, the misses' probabilities have to sum to 1. As the class of hits is missing in the sum we have to divide each probability weight with factor $1 - p(R_{i,\tau})$.

Selection of k hits and k misses from each class instead of just one hit and miss and weighted update of misses is the basic difference to Relief. It ensures greater robustness of the algorithm concerning noise and favorable bias concerning multi-valued attributes and multi-class problems.

Table 1. Characteristics of the problems.

name	c	class distribution	a	#inf	#rnd	n	ε distribution by (7)
C2u	2	0.5, 0.5	9	4	5	1000	0.05 0.95
C2i	2	0.9, 0.1	9	4	5	1000	0.31 0.69
C3u	3	0.33, 0.33, 0.33	11	6	5	1000	0.06 0.18 0.76
C3i	3	0.8, 0.15, 0.05	11	6	5	1000	0.33 0.30 0.37
C5u	5	0.2, 0.2, 0.2, 0.2, 0.2	15	10	5	1000	0.01 0.01 0.03 0.06 0.89
C5i	5	0.5, 0.3, 0.15, 0.04, 0.01	15	10	5	1000	0.16 0.17 0.22 0.12 0.33
C2xu	2	0.5, 0.5	13	8	5	1000	0.05 0.95
C2xi	2	0.9, 0.1	13	8	5	1000	0.31 0.69
C3xu	3	0.33, 0.33, 0.33	17	12	5	1000	0.06 0.18 0.76
C3xi	3	0.8, 0.15, 0.05	17	12	5	1000	0.33 0.30 0.37
C5xu	5	0.2, 0.2, 0.2, 0.2, 0.2	25	20	5	1000	0.01 0.01 0.03 0.06 0.89
C5xi	5	0.5, 0.3, 0.15, 0.04, 0.01	25	20	5	1000	0.16 0.17 0.22 0.12 0.33

3 Imbalanced Data Sets

Misclassification costs are often closely related with imbalanced distribution of class values in the data set (rare classes usually being of higher interest). We first test an ability of described measures to detect attributes which identify minority class values and, for now, we do not assume any knowledge of costs. For that matter we constructed three problems, C2, C3 and C5 with 2, 3, and 5 class values (available labels are c1, c2, c3, c4, or c5). For each class value (2, 3, or 5) we construct two binary attributes A-c?-90 and A-c?-70 (with values 0 and 1). Each binary attribute identifies one class value in 90% or 70% of the cases (e.g., the value of attribute A-c2-90 is 1 in 90% of the cases where the instance is labeled with c2; if label is different from c2, the attribute's value is randomly assigned). In each problem we also have 5 binary random attributes (R-50, R-60, R-70, R-80, and R-90), with 50%, 60%, 70%, 80%, and 90% of 0 values.

To test detection of conditional dependencies we transformed C2, C3 and C5 in such a way, that we replaced each of the informative binary attributes with two attributes, which are XOR of the original attribute (e.g., A-c2-90 is replaced with X1-c2-90 and X2-c2-90, where their values are assigned in such a way that the parity bit of the two attributes equals the value of A-c2-90). We call the transformed problems C2x, C3x, and C5x, respectively.

To observe how the distribution of class values influences the evaluation measures we formed two versions of each problem, one with uniform distribution of class values (data sets with suffix 'u') and one with imbalanced distribution of class values (data sets with suffix 'i'), so altogether 12 data sets. Distribution of class values and characteristics of the problems are given in Table 1.

We begin our analysis with two class problems. Note that for all measures higher score means better attribute, but the scores are not comparable between measures or across problems.

Left-hand side of Table 2 gives evaluations for the problem where class values are uniformly distributed (C2u problem). All the measures give expected

rankings, i.e, attributes identifying values in 90% of the cases have higher scores than 70% attributes. All informative attributes were assigned higher scores than R_{max}, which is the highest score assigned to one of the five random attributes. If its value is larger than the value of some informative attribute that attribute is indistinguishable from random attributes for the respective measure.

Right-hand side of Table 2 contains evaluations for the two class, imbalanced problem (C2i). As before impurity-based measures rank 90% attributes higher than 70% attributes, and they also rank higher the attributes identifying more probable class. ReliefF, on the contrary ranks the minority class higher. The reason for this as well as for the high score of random attribute (R-50) becomes evident if we consider the space of attributes and its role in (4). The negative update of nearest hits in this two cases is likely to be zero (nearest instances have the same values of attributes), and so the positive update of nearest misses is not canceled for random attributes and the attributes identifying minority class.

Table 2. Feature evaluations for C2u and C2i.

	C2u, uniform					C2i, imbalanced				
measure	A-c1-90	A-c1-70	A-c2-90	A-c2-70	R_{max}	A-c1-90	A-c1-70	A-c2-90	A-c2-70	R_{max}
Gain ratio	0.171	0.022	0.193	0.027	0.001	0.078	0.007	0.031	0.017	0.002
DKM	0.110	0.015	0.122	0.018	0.001	0.045	0.007	0.039	0.020	0.003
MDL	0.149	0.018	0.164	0.022	-0.003	0.041	0.002	0.026	0.012	-0.002
ReliefF	0.156	0.033	0.130	0.029	0.008	0.185	0.137	0.301	0.183	0.141

Similar results for three class problems are collected in Table 3. Due to space constraints we omit results for A-c1-70 and A-c2-70, as they show similar trend than A-c3-70, but are always assigned higher scores. With uniform class distribution (left-hand side of the table) all measures except DKM separate informative from random attributes and rank 90% attributes higher than 70% attributes. The values of DKM are completely uninformative (after the split the probability of the majority class is around 0.5, giving high impurity impression). With imbalanced class distribution (p(0)=0.8, p(1)=0.15, p(2)=0.05; right-hand side of the table), all measures rank attributes identifying more frequent classes higher than attributes identifying less frequent classes, 90% attributes higher than 70% attributes, and do not distinguish between A-c3-70 and random attribute with maximal score. ReliefF improves its behavior compared to two class problems, because of more attributes (distances are larger and hits start to normalize the excessive contributions of the misses) and because of its normalizing factor for misses in (4). We get similar results and trends for 5 class problems so we skip the details.

In all problems where informative attributes are replaced with two XOR-ed attributes (C2xi, C2xu, C3xi, C3xu, C5xi, C5xu) the impurity functions do not differentiate between informative and random attributes, while ReliefF does, except for 70% attributes and the best random attribute (R-50). As it is well established fact that ReliefF can detect attributes with strong interactions and

Table 3. Feature evaluations for C3u and C3i.

	C3u, uniform					C3i, imbalanced				
measure	A-c1-90	A-c2-90	A-c3-90	A-c2-70	R_{max}	A-c1-90	A-c2-90	A-c3-90	A-c3-70	R_{max}
Gain ratio	0.138	0.118	0.121	0.029	0.002	0.223	0.066	0.028	0.002	0.002
DKM	-0.055	-0.053	-0.054	-0.038	-0.014	0.121	0.040	0.002	0.001	0.004
MDL	0.123	0.103	0.109	0.021	-0.001	0.149	0.057	0.019	-0.006	-0.000
ReliefF	0.108	0.093	0.086	0.027	0.006	0.262	0.191	0.127	0.094	0.096

impurity based measures cannot this is an expected result but shows that this ability exists in the imbalanced data sets as well. We skip the details.

The attribute evaluation measures we described so far did not take cost information into account. Surely, if such information is available we want that measures take it into account and give higher scores to attributes identifying classes whose misclassification cost is higher. We present such measures in the next section.

4 Implanting Cost-Sensitivity

There are different techniques how to incorporate cost information into learning. The key idea is to use expected cost of misclassification [1,13]. Following [8], we define expected cost of misclassifying an example that belongs to the i-th class as

$$\varepsilon_i = \frac{1}{1-p(\tau_i)} \sum_{\substack{j=1 \\ j \neq i}}^{c} p(\tau_j) C(i,j) \tag{6}$$

and than change the probability estimates for class values:

$$p'(\tau_i) = \frac{p(\tau_i)\varepsilon_i}{\sum_{j=1}^{c} p(\tau_j)\varepsilon_j} \quad . \tag{7}$$

We use (7) in (1) and (2) to make Gain ratio and DKM cost sensitive. In (1) conditional probabilities $p(\tau_i|a_{i,j})$ are also computed in the spirit of (7). We call the respective measures GRatioC and DKMc. This adaptation has the same effect as sampling the data proportionally to (7). MDL uses length of the code instead of probabilities, so we cannot use this approach, but we can sample the data according to (7) and run MDL (3) on the resulting data set. The resulting measure is referred to as MDLs.

For two class problems [8] have adapted Relief[1] to use cost by changing its update formula[2]:

$$W_v = W_v - \mathrm{diff}(A_v, R_i, H)/m + \frac{\varepsilon_{R_{i,\tau}}}{\sum_{j=1}^{c} p(\tau_j)\varepsilon_j} \mathrm{diff}(A_v, R_i, M))/m \quad . \tag{8}$$

[1] Relief uses one nearest hit H and one nearest miss M, so we use diff instead of con.

[2] This formula was typeset incorrectly in [8] (confirmed by M. Kukar, personal communication). Eq. (8) is the correct version which was actually implemented.

This adaptation (called ReliefK in results below) is tailored for two class problems. As we were not satisfied with its performance on multi-class problems we tried different multi-class extensions and used $p'(\tau_i)$ instead of $p(\tau_i)$ in (4). We denote this extension with ReliefFp'. If we use just the information from cost matrix and do not take prior probabilities into account, similarly to (6) and (7), we compute average cost of misclassifying an example that belongs to the i-th class as

$$\alpha_i = \frac{1}{1-c} \sum_{\substack{j=1 \\ j \neq i}}^{c} C(i,j) \ . \tag{9}$$

The prior probability of class value becomes

$$\bar{p}(\tau_i) = \frac{\alpha_i}{\sum_{j=1}^{c} \alpha_i} \ . \tag{10}$$

We use $\bar{p}(\tau_i)$ instead of $p(\tau_i)$ in (4) and call this version ReliefF$\bar{p}$. For two class problems ReliefF, ReliefFp', and ReliefF$\bar{p}$ are identical.

Another idea how to use the cost information stems from the generalized form of ReliefF [12]:

$$W_v^G = \sum_{I_t, I_u \in \mathcal{I}} \text{similarity}(\tau, I_t, I_u) \cdot \text{similarity}(A_v, I_t, I_u) \ ,$$

where I_t and I_u are appropriate samples drawn from the instance population $\mathcal{I}$. For attribute similarity ReliefF uses negative diff function (5) and for class similarity it uses

$$\text{similarity}(\tau, I_t, I_u) = \left\{ \begin{array}{r} 1 \ ; \ I_{t,\tau} = I_{u,\tau} \\ -1 \ ; \ I_{t,\tau} \neq I_{u,\tau} \end{array} \right\} \ , \tag{11}$$

which together gives exactly updates for hits and misses in original Relief. The obvious place to use cost information is therefore (11), which affects the update formula (4). We used cost information in the form of expected and average cost. Using the expected cost, the contribution of class differences in hits costs $\varepsilon_{R_{i,\tau}}$, and different class of miss prevents the actual cost, so (4) changes to

$$\begin{aligned} W_v = W_v &- \varepsilon_{R_{i,\tau}} \text{con}(A_v, R_i, H)/m \\ &+ \sum_{\substack{t=1 \\ t \neq R_{i,\tau}}}^{c} \frac{p(\tau_t) C(R_{i,\tau}, \tau_t) \text{con}(A_v, R_i, M(t))/m}{1 - p(R_{i,\tau})} \ . \end{aligned} \tag{12}$$

We call this measure ReliefFeC. While its updates are symmetric for hits and misses, note that they are not normalized to [0,1], so the scores of the attributes are not necessary normalized to [-1,1] . If we use just cost information (no priors) then we can use average cost of misclassification (ReliefFaC variant)

$$\begin{aligned} W_v = W_v &- \alpha_{R_{i,\tau}} \text{con}(A_v, R_i, H)/m \\ &+ \sum_{\substack{t=1 \\ t \neq R_{i,\tau}}}^{c} \frac{C(R_{i,\tau}, \tau_t) \text{con}(A_v, R_i, M(t))/m}{c-1} \ . \end{aligned} \tag{13}$$

For two class problems ReliefFec and ReliefFac are identical.

We assumed that $C(i,i) = 0$, i.e., that predicting correct class implies no cost. If we are using benefit matrix instead of cost matrix, this is usually not the case, and we suggest using actual $C(i,i)$ instead of expected and average cost as normalizing factor for hits in (12) and (13).

Alternatively, instead of using costs directly, we can change the sampling to reflect the cost matrix as in [1,10]. While this approach may not reflect all the details of cost matrix, it may still work well in practice. We made sampling of random instances of class j in ReliefF (line 3 on Figure 1) proportional to (7). The resulting measure is called ReliefFs. In the next section we test how these measures exploit cost information.

5 Using Cost Information

Following the arguments of [4] and [9] not all cost matrixes are sensible and realistic. We try to test our measures with realistic cost matrixes, e.g., detecting exception for C2, progressive health risk for C3 and financial loss for C5 problems:

$$C2: \begin{bmatrix} 0 & 1 \\ 20 & 0 \end{bmatrix} \qquad C3: \begin{bmatrix} 0 & 1 & 1 \\ 5 & 0 & 1 \\ 20 & 5 & 0 \end{bmatrix} \qquad C5: \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 2 & 0 & 1 & 1 & 1 \\ 5 & 4 & 0 & 1 & 1 \\ 10 & 9 & 6 & 0 & 1 \\ 100 & 99 & 96 & 91 & 0 \end{bmatrix}$$

The right-most column of Table 1 presents the probability distributions by (7) computed from the given class distributions and cost matrixes.

In Table 4 we give results for two and three class problems with imbalanced distribution (C2i and C3i). Uniform distribution is nonrealistic with cost matrix information, so we skip these results. A-70$_{min}$ denotes 70% attribute with minimal score and R_{max} random attribute with maximal score.

For two class problem C2i (left-hand side of Table 4) MDLc and all variants of ReliefF reflect cost-sensitivity, i.e., they evaluate A-c2-90 as better than A-c1-90 (c2 has higher cost assigned, so attribute identifying it is more useful). These measures also separate 90% attributes from the random attributes. Only MDLs separates 70% attribute with minimal score from random attributes for 2 class problems, while none of the measures cannot do that for 3 and 5 class problems, which means, that random attributes are more difficult to detect in the cost-sensitive context. GRatioC and DKMc are also cost-sensitive but fail to separate A-c1-90 from the random attributes.

For three class problem C3i (right-hand side of Table 4) MDLs, ReliefF$\bar{p}$, ReliefFp' and GRatioC are the most cost-sensitive (which can be seen by comparing results with Table 3), followed by ReliefFeC and ReliefFac. ReliefFs and ReliefK are cost-sensitive to a lesser extent. DKMc once again fails completely for multi-class problems as the changed probability distribution moved towards the uniform distribution.

Table 4. Cost-sensitive feature evaluations for C2i and C3i.

	C2i				C3i				
measure	A-c1-90	A-c2-90	A-70$_{min}$	R_{max}	A-c1-90	A-c2-90	A-c3-90	A-70$_{min}$	R_{max}
GRatioC	-0.029	0.114	0.000	0.000	0.177	0.132	0.169	0.009	0.020
DKMc	-0.007	0.083	0.000	0.000	-0.035	-0.032	-0.032	-0.029	0.000
MDLs	0.095	0.189	0.021	-0.000	0.185	0.106	0.144	0.003	0.007
ReliefK	0.078	0.107	0.000	0.034	0.125	0.034	0.044	0.008	0.018
ReliefFp'	0.185	0.306	0.137	0.141	0.286	0.200	0.195	0.136	0.133
ReliefF$\bar{p}$	0.185	0.306	0.137	0.141	0.306	0.208	0.252	0.171	0.166
ReliefFeC	0.236	0.335	-0.125	-0.072	0.405	0.029	0.092	-0.132	-0.020
ReliefFaC	0.236	0.335	-0.125	-0.072	0.352	0.105	0.182	-0.001	0.000
ReliefFs	0.083	0.123	-0.045	-0.025	0.167	0.025	0.050	-0.049	-0.006

These findings are even more radical for the five class problem C5i, where only ReliefF$\bar{p}$, ReliefFp', MDLs and GRatioC can separate 90% attributes from random ones. ReliefFeC and ReliefFaC use (6) and (9) to normalize its hits so they are less stable when large differences between entries in cost matrix are not reflected by sufficiently large number of instances. ReliefFs also suffers from insufficient number of instances, while ReliefK is not properly normalized for multi-class problems.

In problems with XOR-ed attributes ReliefF based measures are cost-sensitive and can differentiate between informative and random attributes, while impurity based measures cannot.

6 Conclusions

We have investigated the performance of common attribute evaluation measures in problems where the class distribution is imbalanced and in problems with unequal misclassification costs. For that matter we constructed several data sets and adapted existing measures. Impurity based measures were adapted by including expected misclassification costs into class probabilities or through sampling. Adaptations of ReliefF stemmed from the expected misclassification cost, average misclassification cost, general form of ReliefF, and cost stratified sampling.

Imbalanced data sets cause no problems to Gain ratio, MDL and ReliefF, while DKM works only for two class problems. Only ReliefF detects highly dependent attributes.

In problems with unequal misclassification costs only MDLs and two variants of ReliefF, which use probability estimates (7) and (10) in the update formula (4), reliably exploit information from cost matrix. Cost-sensitive adaptation of Gain ratio fails to detect all important attributes in two class problem, while DKM is useless for multi-class problems. ReliefF variants retain its ability to detect highly dependent attributes.

While feature evaluation measures need not be cost-sensitive for decision tree building, in further work we want to test this hypothesis the presented measures. We will also investigate feature selection and weighting in the cost-sensitive context.

References

[1] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and regression trees.* Wadsworth Inc., Belmont, California, 1984.

[2] Thomas G. Dietterich, Michael Kerns, and Yishay Mansour. Applying the weak learning framework to understand and improve C4.5. In Lorenza Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference (ICML'96)*, pages 96–103. Morgan Kaufmann, San Francisco, 1996.

[3] Chris Drummond and Robert C. Holte. Exploiting the cost (in)sensitivity of decision tree splitting criteria. In *Proceedings of the Seventeenth International Conference on Machine Learnintg (ICML-2000)*, pages 239–246, 2000.

[4] Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificaial Intelligence (IJCAI'01)*, 2001.

[5] Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In D. Sleeman and P. Edwards, editors, *Machine Learning: Proceedings of International Conference (ICML'92)*, pages 249–256. Morgan Kaufmann, San Francisco, 1992.

[6] Igor Kononenko. Estimating attributes: analysis and extensions of Relief. In Luc De Raedt and Francesco Bergadano, editors, *Machine Learning: ECML-94*, pages 171–182. Springer Verlag, Berlin, 1994.

[7] Igor Kononenko. On biases in estimating multi-valued attributes. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1034–1040. Morgan Kaufmann, 1995.

[8] Matjaž Kukar, Igor Kononenko, Ciril Grošelj, Katarina Kralj, and Jure Fettich. Analysing and improving the diagnosis of ischaemic heart disease with machine learning. *Artificial Intelligence in Medicine*, 16:25–50, 1999.

[9] Dragos D. Margineantu. On class-probability estimates and cost-sensitive evaluation of classifiers. In *Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning (WCSL at ICML-2000)*, 2000.

[10] Dragos D. Margineantu and Thomas G. Dietterich. Bootstrap methods for the cost-sensitive evaluation of classifiers. In *Machine Learning: Proceedings of Seventeenth International Conference on Machine Learning (ICML'2000)*, pages 583–590. Morgan Kaufmann, San Francisco, 2000.

[11] J. Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Francisco, 1993.

[12] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning Journal*, 2003. URL http://www.kluweronline.com/issn/0885-6125/. (forthcoming, available also as technical report at http://lkm.fri.uni-lj.si/rmarko/).

[13] Peter D. Turney and Olcay Boz. On-line cost-sensitive learning bibliography, 1996-2001. URL http://home.ptd.net/ olcay/cost-sensitive.html.