

# Relevancy constraints revisited in ROC space

Nada Lavrač<sup>1</sup>, Dragan Gamberger<sup>2</sup>

<sup>1</sup> Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia, and  
Nova Gorica Polytechnic, Vipavska 13, 5100 Nova Gorica, Slovenia  
nada.lavrac@ijs.si

<sup>2</sup> Rudjer Bošković Institute, Bijenička 54, 10000 Zagreb, Croatia  
dragan.gamberger@irb.hr

**Abstract.** This paper presents relevancy constraints used in subgroup discovery and a novel interpretation of the concept of relevancy in the ROC space context. It provides definitions of feature relevancy and constraints for feature filtering, introduces relevancy based mechanisms for handling of missing values in the examples, and discusses the concept of relevancy as an approach that can help to avoid overfitting. It is argued that logical combinations of features (rules) can be also treated as features and that the same relevancy relations and constraints can be applied for them as well. The paper includes an experimental evaluation of the discussed concepts on a descriptive induction task of gene expression data analysis.

## 1 Introduction

Inductive concept learning can be viewed as a process of searching a space of concept descriptions (hypotheses). The language bias determines the space of hypotheses to be searched. The language bias is determined by the syntactic restrictions of the hypothesis language and the vocabulary of terms in the language. In this paper the hypothesis language is restricted to simple if-then rules of the form  $Class \leftarrow Cond$ , where  $Class$  is the target class and  $Cond$  is a conjunction of features (attribute value pairs). For discrete (categorical) attributes, features have the form  $Attribute = value$  or  $Attribute \neq value$ , for continuous (numerical) attributes they have the form  $Attribute > value$  or  $Attribute \leq value$ .

This work discusses the feature and rule relevancy in the context of subgroup discovery [20, 2, 11] where the goal is to uncover characteristic properties of population subgroups by building short rules which are highly significant (assuring that the distribution of classes of covered instances are statistically significantly different from the distribution in the training set) and have a large coverage (covering many target class instances). The goal is the construction of rules with optimal covering properties on the available example set. A rule with ideal covering properties would be *true* for all target class (positive) examples and *not true* for all non-target class (negative) examples. Target class examples covered by a rule are called *true positives*,  $TP$ , while non-target class examples covered by the rule are called *false positives*,  $FP$ . All remaining non-target class examples

not covered by the rule are called *true negatives*,  $TN$ . An ideal rule has  $TP = P$  and  $TN = N$ , where  $P$  is the set of positive and  $N$  the set of negative examples,  $E = P \cup N$ .

The concept of relevancy is aimed at increasing the quality of induced rules by trying to avoid overfitting the training set. The concept introduces different constraints enabling restrictions of the hypothesis search space and elimination of features and their combinations with low covering properties. The methods based on absolute and relative relevancy are universally applicable to any domain and their use is suggested in all feature based inductive learning tasks. The other restriction methods are related to the form of rules and to the properties of the domain.

This paper provides a novel interpretation of relevancy in the ROC space context. In Section 2 the background is presented: an outline of the subgroup discovery algorithm used in our experiments, the notion of ROC space, relevancy as an approach to feature filtering and the related work. Section 3 provides new definitions of relevancy and reinterprets relevancy in terms of ROC space. Section 4 introduces a functional genomics domain aimed at distinguishing between different cancer types. This section discusses relevancy as constraints for avoiding overfitting. Experimental results of feature filtering show the benefits of relevancy in scientific discovery tasks.

## 2 Background

This section provides the background for this research: the subgroup discovery approach to rule learning, the notion of ROC and  $TP/FP$  space, the concept of feature relevancy based on  $p/n$  pairs of examples and the related work.

### 2.1 Rule learning for subgroup discovery

Subgroup discovery is a form of supervised inductive learning of subgroup descriptions for the target class in a two class domain. The descriptions have the form of rules built as logical conjunctions of features. Features are logical conditions that have values true or false, depending on the values of attributes which describe the examples in the problem domain. Subgroup discovery rule learning is therefore a form of two-class propositional inductive rule learning. Multi-class problems can be solved as a series of two-class learning problems, so that each class is once selected as the target class while examples of all other classes are treated as non-target class examples.

In this work, subgroup discovery is performed by the SD algorithm, a relatively simple iterative beam search rule learning algorithm [2]. The input to SD consists of a set of examples  $E$  and a set of features  $F$  that can be constructed for the given example set. The output of the SD algorithm is a set of rules with optimal covering properties on the given example set. The approach has been implemented in the on-line Data Mining Server (DMS), publicly available at <http://dms.irb.hr>. DMS and its constituting subgroup discovery algorithm

SD can be tested on user submitted domains with up to 250 examples and 50 attributes.

The goal of the subgroup discovery algorithm SD, outlined in Figure 1, is to search for rules that maximize rule quality measure  $q_g = \frac{|TP|}{|FP|+g}$ . High quality rules cover many target class examples and a low number of non-target examples. The user can express his preferences about rule generality (how many target class cases are covered by the rule description) in respect to the rule specificity (how many non-target class cases are covered by the rule) by selecting the parameter  $g$ . For low  $g$  values ( $g \leq 1$ ), induced rules will have high specificity since every false positive classification is made relatively very ‘expensive’. On the other hand, by selecting a high  $g$  value ( $g > 10$  for small domains), more general rules will be generated which can have also many false positive predictions. Suggested  $g$  values in the SD algorithm in the Data Mining Server are in the range between 0.1 and 100, for analyzing data sets of up to 250 examples.

**Algorithm SD: Subgroup Discovery**

**Input:**  $E = P \cup N$  ( $E$  training set,  $|E|$  training set size,  
 $P$  positive (target class) examples,  $N$  negative (non-target class) examples)  
 $F$  set of all defined features,  $f \in F$

**Parameter:**  $g$  (generalization parameter,  $0.1 < g$ , default value 1)  
 $min\_support$  (minimal support for rule acceptance)  
 $beam\_width$  (maximal number of rules in *Beam* and *New\_Beam*)

**Output:**  $S = \{TargetClass \leftarrow Cond\}$  (set of rules formed of  $beam\_width$  best conditions  $Cond$ )

- (1) **for** all rules in *Beam* and *New\_Beam* ( $i = 1$  to  $beam\_width$ ) **do**  
     initialize condition part of the rule to be empty,  $Cond(i) \leftarrow \{\}$   
     initialize rule quality,  $q_g(i) \leftarrow 0$
- (2) **while** there are improvements in *Beam* **do**
- (3)     **for** all rules in *Beam* ( $i = 1$  to  $beam\_width$ ) **do**
- (4)         **for** all  $f \in F$  **do**
- (5)             form a new rule by forming a new condition as a conjunction of the  
                 condition from *Beam* and feature  $f$ ,  $Cond(i) \leftarrow Cond(i) \wedge f$
- (6)             compute the quality of a new rule as  $q_g = \frac{|TP|}{|FP|+g}$
- (7)             **if**  $\frac{|TP|}{|E|} \geq min\_support$  **and if**  $q_g$  is larger than any  $q_g(i)$  in *New\_Beam*  
                 **and if** the new rule is relatively relevant **do**
- (8)                 replace the worst rule in *New\_Beam* with the new rule and  
                     reorder the rules in *New\_Beam* with respect to their quality
- (9)             **end for** features
- (10)     **end for** rules from *Beam*
- (11)      $Beam \leftarrow New\_Beam$
- (12) **end while**

**Fig. 1.** Heuristic beam search rule construction algorithm for subgroup discovery.

In addition to parameters  $g$  and  $min\_support$ , the SD algorithm has an additional parameter which is defined by the user, but which does not need to be adjusted frequently. The  $beam\_width$  parameter (default value is 100 for gene expression domains) defines the number of solutions kept in the beam in each iteration. The output of the algorithm is set  $S$  of  $beam\_width$  different rules

with highest  $q_g$  values. In the described experiments we have used only the first (best) solution although there is a possibility to select a few relatively different solutions using the algorithm described in [2], or to enter the expert evaluation process with a set of a few best rules, letting the experts select the optimal solution(s). Moreover, the rules from set  $S$  could be used as an input to a redundant voting classifier, but this variant is out of the scope of this work.

The algorithm initializes all the rules in *Beam* and *New\_beam* by empty rule conditions. Their quality values  $q_g(i)$  are set to zero (step 1). Rule initialization is followed by an infinite loop (steps 2–12) that stops when, for all rules in the beam, it is no longer possible to further improve their quality. Rules can be improved by conjunctively adding features from  $F$ . After the first iteration, a rule condition consists of a single feature, after the second iteration up to two features, and so forth. The search is systematic in the sense that for all rules in the beam (step 3) all features from  $F$  (step 4) are tested in each iteration. For every new rule, constructed by conjunctively adding a feature to rule body (step 5), quality  $q_g$  is computed (step 6). If the support of the new rule is greater than *min\_support* and if its quality  $q_g$  is greater than the quality of any rule in *New\_beam*, the worst rule in *New\_beam* is replaced by the new rule. The rules are reordered in *New\_beam* according to their quality  $q_g$ . At the end of each iteration, *New\_beam* is copied into *Beam* (step 11). When the algorithm terminates, the first rule in *Beam* is the rule with maximum  $q_g$ .

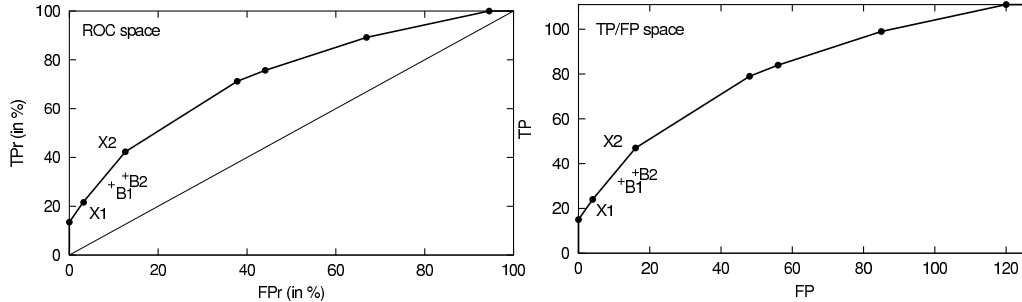
A necessary condition (in step 7) for a rule to be included in *New\_beam* is its relative relevancy. This is the concept described in Section 3.1 implemented for rules. A new rule is irrelevant if there already exists a rule  $R$  in *New\_beam* such that true positives of the new rule are a subset of true positives of  $R$  and true negatives of the new rule are a subset of true negatives of  $R$ . After the new rule is included in *New\_beam* it may happen that some of the existing rules in *New\_beam* become relatively irrelevant with respect to this new rule. Such rules are eliminated from *New\_beam* during its reordering (in step 8). The testing of relevancy ensures that *New\_beam* contains only different and relatively relevant rules.

## 2.2 ROC space

A point in ROC space (ROC: Receiver Operating Characteristic) [15] shows classifier performance in terms of false alarm or *false positive rate*  $FPr = \frac{|FP|}{|TN|+|FP|} = \frac{|FP|}{|N|}$  (plotted on the  $X$ -axis), and sensitivity or *true positive rate*  $TPr = \frac{|TP|}{|TP|+|FN|} = \frac{|TP|}{|P|}$  (plotted on the  $Y$ -axis).

In the context of subgroup discovery, each point in ROC space represents covering properties of a rule. ROC space is appropriate for measuring the success of subgroup discovery, since rules/subgroups whose  $TPr/FPr$  tradeoff is close to the diagonal can be discarded as insignificant; the reason is that the rules with  $TPr/FPr$  on the diagonal have the same distribution of covered positives and negatives as the distribution in the entire data set. Conversely, significant rules/subgroups are those sufficiently distant from the diagonal. Subgroups that

may be optimal under varying  $TPr/FPr$  tradeoffs form a convex hull called the ROC curve. Figure 2 presents seven rules on the convex hull (marked by circles), including  $X1$  and  $X2$ , while two rules  $B1$  and  $B2$  below the convex hull (marked by +) are of lower quality in terms of their  $TPr/FPr$  tradeoff.



**Fig. 2.** The left-hand side figure shows ROC space with a convex hull formed of seven rules that are optimal under varying  $TPr/FPr$  tradeoffs, and two suboptimal rules  $B1$  and  $B2$ . The right-hand side presents the positions of the same rules in the corresponding  $TP/FP$  space.

It was shown in [11] that the distance of a point to the ROC diagonal is proportional to the significance of the rule. Hence, the goal of a subgroup discovery algorithm is to find subgroups in the upper-left corner area of ROC space. The most significant rule would lie in point  $(0, 1)$  representing a rule covering only positive and none of the negative examples ( $FPr = 0\%$  and  $TPr = 100\%$ ).

An alternative to ROC space is the so-called  $TP/FP$  space (see the right-hand side of Figure 2), where  $FPr$  on the  $X$ -axis is replaced by  $|FP|$  and  $TPr$  on the  $Y$ -axis by  $|TP|$ .<sup>3</sup>  $TP/FP$  space is equivalent to ROC space when comparing the quality of subgroups induced in a single domain. For simplicity, the remainder of this paper considers only this simpler  $TP/FP$  space representation.

### 2.3 Theory of relevancy based on p/n pairs of examples

The main aim of the theory of relevancy, described in [9, 10], is to reduce the hypothesis space by the elimination of irrelevant features. Consider a two-class learning problem in which examples  $e \in E$  are tuples of truth-values of features  $F$ . Training set  $E$  is represented as a table where rows correspond to training examples and columns correspond to features. A sample table is shown in Table 1. An element in the table has the value *true* when the example satisfies the condition (feature) in the column of the table, otherwise its value is *false*.

**Definition 1: p/n pairs.**

A *p/n pair* is a pair of training examples where  $p \in P$  and  $n \in N$ .

<sup>3</sup>  $TP/FP$  space can be turned into ROC space by simply normalizing the  $TP$  and  $FP$  axes to scale  $[0,1] \times [0,1]$ .

**Definition 2: Coverage of p/n pairs.**

Let  $F$  denote a set of features. Feature  $f \in F$  covers a  $p/n$  pair iff feature  $f$  has value *true* for  $p$  and value *false* for  $n$ .

Notice that in the standard machine learning terminology we may reformulate the definition of coverage of  $p/n$  pairs as follows: feature  $f$  covers a  $p/n$  pair iff  $f$  covers (has value *true* for) the positive example  $p$  and does not cover (has value *false* for) the negative example  $n$ .

The notion of  $p/n$  pairs can be used to prove important properties of features for building complete and consistent concept descriptions. The following theorem assumes that the hypothesis language  $\mathcal{L}$  is rich enough to allow for a complete and consistent hypothesis  $H$  to be induced from the set of training examples  $E$ .

**Theorem 1.**

Assume a training set  $E$  and a set of features  $F$  such that a complete and consistent hypothesis  $H$  can be found. Let  $F' \subseteq F$ . A complete and consistent hypothesis  $H$  can be found using only features from set  $F'$  iff for each possible  $p/n$  pair from the training set  $E$  there exists at least one feature  $f \in F'$  that covers the  $p/n$  pair.

The theorem is proved in [10]. Its importance for the theory of relevancy is manifold. First, it points out that when deciding about the relevancy of features it will be significant to detect which  $p/n$  pairs are covered by the feature. Second, the theorem implies that useless features are those that do not cover any  $p/n$  pair. An important property of pairs of features can now be defined—coverage of features—which was defined in [9, 10] as follows.

**Definition 3: Coverage of features.**

Let  $f \in F$ . Let  $E(f)$  denote the set of all  $p/n$  pairs covered by feature  $f$ . Then  $f_{rel}$  covers feature  $f$  (i.e.,  $f_{rel}$  is more relevant than  $f$ ) iff  $E(f) \subseteq E(f_{rel})$ .

*Example 1.* Consider a domain with two positive examples,  $P = \{p_1, p_2\}$ , two negative examples  $N = \{n_1, n_2\}$ , and six features: three positive features  $F_p = \{f_1, f_2, f_3\}$  and their complements  $F_n = \{\bar{f}_1, \bar{f}_2, \bar{f}_3\}$ , illustrated in Table 1.<sup>4</sup>

Examples		Features					
Ex.	Cl.	$f_1$	$\bar{f}_1$	$f_2$	$\bar{f}_2$	$f_3$	$\bar{f}_3$
$p_1$	$\oplus$	false	true	true	false	false	true
$p_2$	$\oplus$	false	true	false	true	true	false
$n_1$	$\ominus$	true	false	true	false	true	false
$n_2$	$\ominus$	false	true	false	true	false	true

**Table 1.** Training examples represented as vectors of truthvalues of positive and negative features.

In this example feature  $f_1$  does not cover any  $p/n$  pair,  $E(f_1) = \emptyset$ , therefore it can be eliminated as irrelevant for learning a concept description  $H$ . Its logical

<sup>4</sup> If  $f_1$  is feature  $A_1 = v_1$  then its logical complement  $\bar{f}_1$  is feature  $A_1 \neq v_1$ . For  $A_2 > v_2$  its logical complement  $\bar{f}_2$  is feature  $A_2 \leq v_2$ .

complement  $\overline{f_1}$  covers two  $p/n$  pairs,  $E(\overline{f_1}) = \{p_1/n_1, p_2/n_1\}$ . Feature  $f_2$  covers one  $p/n$  pair,  $E(f_2) = \{p_1/n_2\}$  and its logical complement  $\overline{f_2}$  covers only the pair built of  $p_2$  and  $n_1$ . Although  $\overline{f_2}$  is a logical complement of  $f_2$ , the sets of  $p/n$  pairs covered by  $f_2$  and  $\overline{f_2}$  are different, therefore both the feature and its complement need to be considered as relevant for learning. Consequently, in hypothesis construction and in feature elimination we should consider a set of features  $F$  consisting of positive and negative features:  $F = F_p \cup F_n$ . □

## 2.4 Related work

The problem of relevance has been addressed already in early inductive concept learning research [13]. This problem is actually encountered by every inductive learner. Basically, all learners are concerned with the selection of relevant literals. Usually, at each step of learning, the choice of the ‘best’ or ‘most informative’ literal needs to be made. This choice is frequently based on the distribution of positive and negative examples covered by the hypothesis before and after literal selection (e.g., [16, 17]). Whereas in most learning systems the selection of significant or informative literals is part of the learning process, the presented theory of relevance is aimed at pointing out which literals constitute a set of relevant literals and which literals are irrelevant and can be discarded, without even entering the ‘best literal’ competition. Such filtering of irrelevant literals can thus be done in preprocessing of the set of training examples. Whereas most other algorithms only consider the ‘local training set’ (e.g., a subset of examples covered by the currently developed rule, or a subset of examples in the currently developed node of a decision tree) when deciding about the importance/relevance of literals, we are concerned with finding ‘globally relevant’ literals w.r.t. the entire set of training examples. This is important since the elimination of globally irrelevant literals guarantees that literal elimination will not harm the hypothesis formation process.

The problem of relevance has recently attracted much attention in the context of feature subset selection in propositional learning [7]. An extensive discussion of different approaches to feature subset selection can be found in [6], which distinguishes between filter and wrapper approaches, and introduces the notions of totally irrelevant, weakly relevant and strongly relevant features. In this categorisation, our work belongs to filter approaches which eliminate totally irrelevant features in preprocessing. Other filtering approaches include different versions of the RELIEF algorithm [5, 8], the FOCUS algorithm [1] and an approach to feature selection proposed in [14]. Wolf and Shashua [19] perform feature selection using relevancy constraints, providing an algebraic definition of relevancy.

## 3 Interpretation of the relevancy concept in ROC space

The concept of feature coverage is important as a relevancy constraint for the inductive learning process. The concept is not valid only for features but also for any logical combination of features and for complete rules. Filtering based

on absolute and relative relevancy are applicable to any domain. While the aim of absolute relevancy is to ensure minimal quality that must be satisfied by every feature (rule), relative relevancy aims to ensure that only the best among available features (rules) can enter (stay in) the rule construction process. The definition of relative irrelevancy is very important because it does not depend on user defined constraints and its usage is suggested for all feature based machine learning applications [9]. The methodology can be useful also as a preprocessing filter for attribute based learners, like decision tree induction systems because complete attributes can be detected as irrelevant if all features generated for these attribute are detected as relatively or absolutely irrelevant.

### 3.1 Relative relevancy

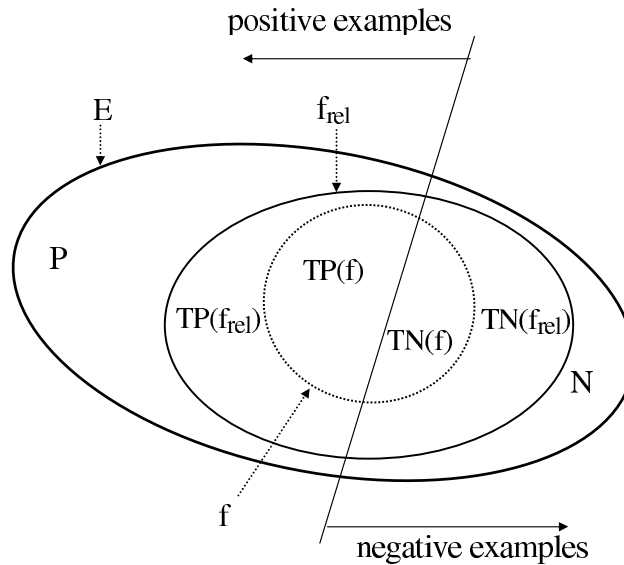
Let us now reinterpret feature relevancy discussed in Section 2.2 in the context of ROC and  $TP/FP$  space.

**Definition 4: Coverage of features revisited.**

Feature  $f_{rel}$  covers feature  $f$  (i.e., feature  $f_{rel}$  is more relevant than  $f$ ) iff true positives of  $f$  are a subset of true positives of  $f_{rel}$  and true negatives of  $f$  are a subset of true negatives of  $f_{rel}$ , i.e., iff  $TP(f) \subseteq TP(f_{rel})$  and  $TN(f) \subseteq TN(f_{rel})$  (see Figure 3).

**Definition 5: Relative relevancy.**

Feature  $f$  is relatively irrelevant iff there exists another feature  $f_{rel}$  such that  $f_{rel}$  covers  $f$ .



**Fig. 3.** The concept of relative relevancy illustrated by features  $f$  and  $f_{rel}$ . Feature  $f$  is relatively irrelevant because  $TP(f) \subseteq TP(f_{rel})$  and  $TN(f) \subseteq TN(f_{rel})$ .

**Theorem 2.**

If feature  $f_{rel}$  covers feature  $f$  and feature  $g_{rel}$  covers  $g$  then  $f_{rel} \wedge g_{rel}$  covers  $f \wedge g$ .

It is trivial to prove the theorem by first fixing one of the two conjuncts, e.g.,  $g_{rel} = g$  and showing that in this case  $TP(f \wedge g) \subseteq TP(f_{rel} \wedge g)$  and  $TN(f \wedge g) \subseteq TN(f_{rel} \wedge g)$ . Next, the same relationship can be shown also for the case when  $g_{rel}$  covers  $g$ .  $\square$

Theorem 2 could be proved also for logical *OR* operation ( $f_{rel} \vee g_{rel}$ ). Consequently, if for feature  $f$  there exists another feature  $f_{rel}$  with the property that if in any rule  $f$  is substituted by  $f_{rel}$  the rule quality measured by the number of correct classifications  $|TP|$  and  $|TN|$  does not decrease, then it means that  $f_{rel}$  can be always used instead of  $f$ , and that we actually do not need  $f$ . This means that  $f$  can be eliminated as irrelevant.

The importance of the concept lies in the fact that some feature  $f$  is not irrelevant because of small  $|TP|$  or  $|TN|$  values but because there exists at least one other feature with better covering properties. In this way it can not happen to eliminate features that become relevant only in the context of some other features. Relative relevance ensures the quality of induced rules but, what is perhaps even more important from the view of avoiding overfitting, it ensures that the rule inducers may use only the best available features.

Consider the simplest form of rules, whose condition consists of a single feature. Suppose such a rule is plotted in the  $TP/FP$  space, meaning that each feature represents a point in the space. ‘Good’ features are those as close as possible to point  $(0, P)$  in the  $TP/FP$  space ( $(0, 1)$  in the ROC space).  $TP/FP$  space at the left-hand side of Figure 4 presents the concept of relative relevancy. Let us plot two features  $f$  and  $f_{rel}$  in the space. As  $|TP(f)| \leq |TP(f_{rel})|$ , feature  $f_{rel}$  is plotted higher along the  $TP$ -axis. As  $|TN(f)| \leq |TN(f_{rel})|$ , therefore  $|FP(f_{rel})| \leq |FP(f)|$  and feature  $f_{rel}$  is plotted more to the left (closer to the  $TP$ -axis) along the  $FP$ -axis than feature  $f$ . The figure shows feature  $f$ , a shaded area left-up from  $f$  showing the part of  $TP/FP$  space of features  $f_{rel}$  (more relevant features) that may cover feature  $f$ , and a shaded area right-down from  $f_{rel}$  showing the part of space of features that may be irrelevant due to  $f_{rel}$ . It must be noted that not all features left-up from  $f$  are more relevant and not all features right-down from  $f_{rel}$  are irrelevant, but only those that satisfy Definition 4.

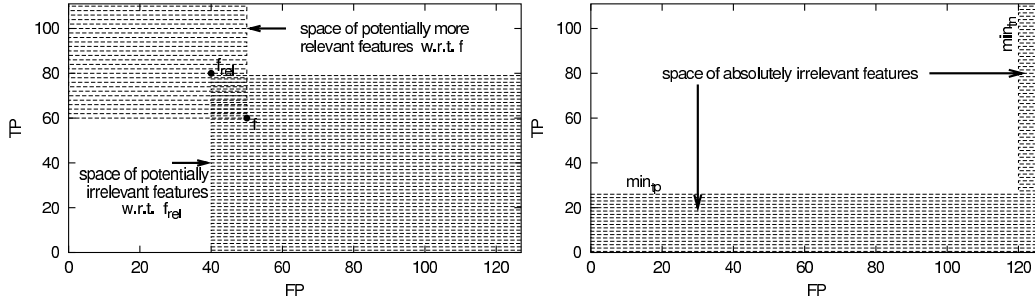
**3.2 Total and absolute relevancy**

In addition to irrelevant features defined through relative relevancy, also totally irrelevant features—those which are totally useless for distinguishing between the classes—can be eliminated in preprocessing.

**Definition 6: Total irrelevancy.**

Feature  $f$  with  $|TP(f)| = 0$  or  $|TN(f)| = 0$  is totally irrelevant.

In order for a feature to be acceptable as a building block of rules representing some genuine dependencies between classes and attribute values, the feature itself must demonstrate some interesting covering properties on the training set. These can be defined in terms of user-defined support constraints.



**Fig. 4.** The left-hand side figure presents the concept of relative relevancy while the right-hand side figure presents the concept of absolute relevancy.

**Definition 7: Absolute irrelevancy.**

*Feature  $f$  that has either  $|TP(f)| < min\_tp$  or  $|TN(f)| < min\_tn$  is absolutely irrelevant, for  $min\_tp$  and  $min\_tn$  being user defined constants.*

Feature  $f$  with  $|TP(f)| < min\_tp$  is true for a small number of target class examples and feature  $g$  with  $|TN(g)| < min\_tn$  is false for a small number of non-target class examples. Such small numbers may be due to statistical chance so that it seems reasonable not to use features with either of these properties in the rule construction process. The concept of absolute relevancy is presented in the right-hand side of Figure 4.

Although the significance of rules is proportional to their distance from the diagonal of the ROC space (Figure 2), this property is not appropriate as a criteria for the features. Logical combinations of features lying on the diagonal or very near to it, can result in very significant rules. In this situation only relative and absolute relevancy constraints as those defined in this work are applicable for features.

By conjunctive connection of features, the generated rule will have  $|TP|$  equal or smaller than the smallest  $|TP|$  value of the features forming a conjunctive subgroup description. In contrast, the  $|TN|$  value of a rule will be at least as large as the largest  $|TN|$  of the used features. This is the reason why  $min\_tp$  is typically selected higher than  $min\_tn$  (see the right-hand side of Figure 4) and it can be as large as the minimal estimated number of examples that must be covered by any acceptably good subgroup for the domain. The problem with absolute irrelevancy is that both  $min\_tp$  and  $min\_tn$  are user defined constraints and that any value, regardless how high it is, can not guarantee that a feature is actually relevant. A good starting values for gene expression domains are  $min\_tp = |P|/2$  and  $min\_tn = \sqrt{|N|}$  which have been used in all experiments reported in Section 4. Good news is that selection of these constants is not very critical for the final result because the majority of absolute irrelevant features is detected as relative irrelevant as well. With mentioned  $min\_tp$  and  $min\_tn$  values in gene expression domains more than 90% of absolute irrelevant features were also detected as relative irrelevant.

## 4 Experiments in functional genomics

Gene expression monitoring by DNA microarrays (gene chips) provides an important source of information that can help in understanding many biological processes. The database we analyze consists of a set of gene expression measurements (examples), each corresponding to a large number of measured expression values of a predefined family of genes (attributes). Each measurement in the database was extracted from a tissue of a patient with a specific disease; this disease is the class for the given example. The standard goal of machine learning is to start from such available labeled examples and construct classifiers that can successfully classify new, previously unseen examples. Such classifiers are important because they can be used for diagnostic purposes in medicine and because they can help to understand the dependencies between classes (diseases) and attributes (gene expressions values).

### 4.1 Subgroup discovery experiments

The gene expression domain, described in [18, 3] and used in our experiments, is a typical scientific discovery domain characterised by very many attributes compared to the number of available examples. It is a domain with 14 different cancer classes and 144 training examples in total. Eleven classes have 8 examples each, two classes have 16 examples and only one has 24 examples. The examples are described by 16063 attributes presenting gene expression values. As will be shown in Section 4.2, in all the experiments we have used the gene presence call values ( $A$ ,  $P$ , and  $M$ ) to describe the training examples. The domain can be downloaded from <http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi>. There is also an independent test set with 54 examples.

The experiments were performed separately for each cancer class so that a two-class learning problem was formulated where the selected cancer class was the target class and the examples of all other classes formed non-target class examples. In this way the domain was transformed into 14 inductive learning problems, each with the total of 144 training examples and with between 8 and 24 target class examples. For each of these tasks a complete procedure consisting of feature construction, elimination of irrelevant features, and induction of subgroup descriptions in the form of rules was repeated. Finally, using the SD subgroup discovery algorithm [2], for each class a single rule with maximal  $q_g$  value has been selected, for  $q_g = \frac{|TP|}{|FP|+g}$  being the heuristic of the SD algorithm, and  $g$  being a user defined *generalization parameter*. The rules for all 14 tasks consisted of 2–4 features. The induced rules have been tested on the independent example set.

There are very large differences among the results on the test sets for various classes (diseases) and the precision higher than 50% has been obtained for only 5 out of 14 classes. There are only three classes (lymphoma, leukemia, and CNS) with more than 8 training cases and all of them are among those with high precision on the test set, while for only two out of eleven classes with 8

training cases (colorectal and mesothelioma) high precision was achieved. The classification properties of rules induced for classes with 16 and 24 target class examples (lymphoma, leukemia and CNS) are comparable to those reported in [18], while the results on eight small example sets with 8 target examples were poor. An obvious conclusion is that the use of the subgroup discovery algorithm is not appropriate for problems with a very small number of examples because overfitting can not be avoided in spite of the heuristics used in the SD algorithm and the additional domain-specific techniques used to restrict the hypothesis search space. But for larger training sets the subgroup discovery methodology enabled effective construction of relevant knowledge.

The induced rules for lymphoma, leukemia and CNS have been evaluated by a domain expert and most of features used in them were recognized as known disease markers for the target class cancers [3]. Expert evaluation, which is out of scope of this work, proved the relevancy of induced rules. Both good prediction results on an independent test set as well as expert interpretation of induced rules prove the effectiveness of described methods for avoiding overfitting in scientific discovery tasks. Results achieved in these three tasks are presented in Table 2.

Cancer	Training set			Test set		
	Sens.	Spec.	Prec.	Sens.	Spec.	Prec.
lymphoma	16/16	128/128	100%	5/6	48/48	100%
leukemia	23/24	120/120	100%	4/6	47/48	80%
CNS	16/16	128/128	100%	3/4	50/50	100%

**Table 2.** Covering properties on the training and on the independent test set for rules induced for 3 classes with 16 and 24 examples. Sensitivity is  $|TP|/|P|$ , specificity is  $|TN|/|N|$ , while precision is defined as  $|TP|/(|TP| + |FP|)$ .

## 4.2 Specific constraints for functional genomics domain

This section presents an effective approach that can strongly reduce the number of features and its application is suggested for subgroup discovery in gene expression domains, used as a case study in the exploration of feature relevancy.

**Choice of the language of features.** Gene expression scanners measure signal intensity as continuous values which form an appropriate input for data analysis. The problem is that for continuous valued attributes there can be potentially many boundary values separating the classes, resulting in many different features for a single attribute. There is also a possibility to use presence call (signal specificity) values computed from measured signal intensity values by the Affymetrix GENECHIP software. The presence call has discrete values  $A$  (absent),  $P$  (present), and  $M$  (marginal). The  $M$  value can be interpreted as a 'do not know state' and for the remaining values  $A$  and  $P$  it holds that feature  $Attribute = A$  is identical to  $Attribute \neq P$ ; consequently, for every attribute there are only two distinct features  $Attribute = A$  and  $Attribute = P$ .

The subgroup discovery induction as well as the filtering based on feature and rule relevancy are applicable regardless of using the signal intensity or the presence call attribute values. Typically signal intensity values are used [12] because they impose less restrictions to the classifier construction process and because the results do not depend on the GENECHIP software presence call computation. For subgroup discovery we prefer the later approach based on presence call values. The reason is that features presented by conditions like  $A_i$  is true ( $A_i$  is present) or  $A_j$  is false ( $A_j$  is absent) are very natural for human interpretation and that the approach can help in avoiding overfitting, as the feature space is very strongly restricted.

<i>Examples</i>		<i>Genes</i>			<i>Features</i>					
<i>Ex.</i>	<i>Cl.</i>	<i>X</i>	<i>Y</i>	<i>Z</i>	$X = A$	$X = P$	$Y = A$	$Y = P$	$Z = A$	$Z = P$
$p_1$	$\oplus$	<i>A</i>	<i>A</i>	<i>A</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>
$p_2$	$\oplus$	<i>P</i>	<i>P</i>	<i>A</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>
$p_3$	$\oplus$	<i>A</i>	<i>A</i>	<i>P</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
$p_4$	$\oplus$	<i>P</i>	<i>P</i>	<i>A</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>
$p_5$	$\oplus$	<i>M</i>	<i>A</i>	<i>A</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>
$n_1$	$\ominus$	<i>A</i>	<i>P</i>	<i>P</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
$n_2$	$\ominus$	<i>P</i>	<i>P</i>	<i>P</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
$n_3$	$\ominus$	<i>M</i>	<i>M</i>	<i>A</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>false</i>
$n_4$	$\ominus$	<i>P</i>	<i>P</i>	<i>A</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>

**Table 3.** Training examples represented as vectors of truthvalues of positive and negative features. Notice that value *M* (marginal) is treated as a ‘do not know’ value as follows: for a positive example its value is *false* for both its positive and negative feature, and for a negative example both feature values are *true*.

**Illustrating feature relevancy and handling of ‘do not know’ values.**

Marginal presence call attribute value *M* is treated as a ‘do not know state’. The value can neither be used to support the relevancy of a feature or a rule, nor can it be used for prediction purposes. To obtain this effect, an attribute ‘do not know state’ in a positive example is in all features constructed from this attribute coded by *false* truthvalues while a ‘do not know state’ occurring in a negative example is coded by *true* values in all features constructed from the attribute. The coding is illustrated in Table 3.

The table presents five positive and four negative examples of a selected leukemia class. The example shows that two features are generated for each of the three genes *X*, *Y* and *Z*, leading to the generation of the following six features:  $X = A$ ,  $X = P$ ,  $Y = A$ ,  $Y = P$ ,  $Z = A$  and  $Z = P$ . In this example, following Definition 5 of relative relevancy, feature  $X = A$  is relatively irrelevant because of feature  $Y = A$ , and feature  $X = P$  is relatively irrelevant because of feature  $Z = A$ . Consequently, both features generated for gene *X* can be eliminated as irrelevant.

Tasks	All	Total	Absolute	Relative
Task 1 Real domain with 16063 att.	32126	23500	9628	4445
Task 2 Randomly generated domain with 16063 att.	32126	27500	16722	16722
Task 3 Combination of 16063 real and 16063 randomly generated attributes	64252	51000	26350	15712

**Table 4.** This table presents mean numbers of the constructed features for lymphoma, leukemia, and CNS domains. Presented are the total number of features (All), the number of features after the elimination of totally irrelevant features (Total), the number of features after the elimination of absolutely irrelevant features (Absolute), and the number of features after the elimination of absolutely and relatively irrelevant features (Relative). These three values are shown for following training sets: the real training set with 16063 genes (with 32126 gene expression activity values, constructed as  $Gene = A$  and  $Gene = P$ ), a randomly generated set with 16063 genes, and a set with 32126 genes which is a combination of 16063 real and 16063 random attributes.

### 4.3 Experiments in feature filtering

In the rest of this paper experiments are performed on three classes with a sufficient number of training instances—lymphoma, leukemia, and CNS—for which successful induction of rules was possible. For the selected three classes both concepts of absolute and relative relevancy have been tested both for real and randomly generated attributes. Experiments have been done on three different tasks with results summarized in Table 4.

**Task 1.** In this task the real domain with 16063 attributes has been used. For the selected three classes both concepts of absolute and relative relevancy were very effective in reducing the number of features. About 60% of all features were detected as absolutely irrelevant while relative irrelevancy was even more effective as it managed to eliminate up to 75% of all features. Their combination resulted in the elimination of 75% to 85% of all features. These results are presented in the first row of Table 4. The set of all features in these experiments was generated so that for each gene (attribute) two features were constructed ( $Gene = A$  and  $Gene = P$ ), followed by eliminating totally irrelevant features (with  $|TP| = 0$  or  $|TN| = 0$ ), which substantially reduced the total number of features.

**Task 2.** Another domain with 16063 completely randomly generated attribute values has also been constructed, and the same experiments were repeated on this artificial domain as for the real gene expression domain. The results (repeated with 5 different randomly generated attribute sets) were significantly different: there have been only about 40% of absolutely irrelevant features and practically no relatively irrelevant features. The results are presented in the second row

of Table 4. Comparing the results for the real and for the randomly generated domain, especially large differences can be noticed in the performance of relative relevancy. It is the consequence of the fact that in the real domain there are some features that are really relevant; they cover many target class examples and a few non target class examples and in this way they make many other features relatively irrelevant. The results prove the importance of relative relevancy for domains in which strong and relevant dependencies between classes and attribute values exist.

**Task 3.** The experiments with feature relevancy continued with another domain with 32126 attributes, generated as the combination of two previous domains with 16063 attributes each: the real and the randomly generated domain. The results are presented in the last row of Table 4. After the elimination of absolutely irrelevant features the number of features is equal to the sum of features that remained in the two independent domains with 16063 attributes. In contrast, relative relevancy was much more effective. Besides eliminating many features from the real attribute part it was now possible to eliminate also a significant part of features of randomly generated attributes.

**Summary of the experiments.** From this analysis it is obvious that the elimination of features is very effective in real domains. The same result has been confirmed in experiments with domains with only 8 target class examples. It is important that in domains which are combinations of real and random attributes the methodology is effective: in Task 3 less features remained after feature elimination (15712 features) than in Task 2 (16722 features). This proves that the presented methodology, especially relative relevancy, can be very useful in avoiding overfitting by reducing the hypothesis search space through the elimination of non-significant dependencies between attribute values and classes. This property is important because it can be assumed that among 16063 real attributes there are many of them which are irrelevant with respect to the target class.

## 5 Conclusions

This work reinterprets the theory of feature relevancy, described in [9, 10], in the context of ROC analysis. Moreover, it provides an experimental evaluation of the usefulness of feature elimination in a functional genomics domain. We have implemented domain dependent restrictions by using discrete instead of continuous attribute values, and domain independent restrictions by the elimination of irrelevant features. Interpretation of marginal gene values as a '*do not know state*' helps in reducing the feature space and ensures the robustness of induced rules. Subgroup discovery approach proved to be a useful framework for the implementation of different relevancy conditions and an appropriate tool for descriptive induction tasks.

## References

1. H. Almuallim and T.G. Dietterich. Learning with many irrelevant features, In *Proceedings of the 9th National Conference on Artificial Intelligence*, 547–552. The MIT Press, 1991.
2. D. Gamberger and N. Lavrač. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research* 17: 501–527, 2002.
3. D. Gamberger, N. Lavrač, F. Železný, and J. Tolar. Induction of comprehensible models for gene expression datasets by the subgroup discovery methodology. *Journal of bioinformatics* (to appear).
4. T.R. Golub et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286: 531–537, 1999.
5. K. Kira and L.A. Rendell. A practical approach to feature selection, In *Proceedings of the 9th International Conference on Machine Learning*, 249–256. Morgan Kaufmann, 1992.
6. R. Kohavi and G.H. John, Wrappers for feature subset selection. *Artificial Intelligence, Special Issue on Relevance*, 97(1–2):273–324, 1997.
7. D. Koller and M. Sahami. Toward optimal feature selection. *Proceedings of the 13th International Conference on Machine Learning*, 284–292. Morgan Kaufmann, 1996.
8. I. Kononenko. Estimating attributes: Analysis and extensions of Relief, In *Proceedings of the 7th European Conference on Machine Learning*, LNAI 784, 171–182. Springer, 1994.
9. N. Lavrač, D. Gamberger and P. Turney. A relevancy filter for constructive induction. *IEEE Intelligent Systems and their Applications* 13: 50–56, 1998.
10. N. Lavrač, D. Gamberger and V. Jovanoski. A study of relevance for learning in deductive databases. *Journal of Logic Programming* 40: 215–249, 1999.
11. N. Lavrač, B. Kavšek, P. Flach and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5: 153–188, 2004.
12. J. Li and L. Wong. Geography of differences between two classes of data. In *Proceedings of 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2002)*, 325–337, Springer, 2002.
13. R.S. Michalski. A theory and methodology of inductive learning, In: R. Michalski, J. Carbonell and T. Mitchell (eds.) *Machine Learning: An Artificial Intelligence Approach*, Tioga, 83–134, 1983.
14. A.L. Oliveira and A. Sangiovanni-Vincentelli. Constructive induction using a non-greedy strategy for feature selection. In *Proceedings of the 9th International Conference on Machine Learning*, 354–360. Morgan Kaufmann, 1992.
15. F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3): 203–231, 2001.
16. J.R. Quinlan, Learning logical definitions from relations, *Machine Learning* 5(3): 239–266, 1990.
17. J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
18. S. Ramaswamy et al. Multiclass cancer diagnosis using tumor gene expression signatures. In *Proceedings Natl. Acad. Sci. USA*, 98(26): 15149–15154, 2001.
19. L. Wolf and A. Shashua. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weighted-based approach. In *Proceedings of the International Conference on Computer Vision*, Volume 1, 378–384, 2003.
20. S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery*, 78–87. Springer, 1997.