

A Distance-Based Approach for Action Recommendation

Ronan Trepos^{1,2}, Ansaf Salleb¹, Marie-Odile Cordier¹,
Véronique Masson¹, and Chantal Gascuel²

¹ IRISA-INRIA, Campus Universitaire de Beaulieu, 35042, Rennes Cedex - France
{rtrepos, asalleb, cordier, masson}@irisa.fr

² INRA UMR SAS, Agrocampus rue de Saint Brieu, 35042, Rennes Cedex - France
Chantal.Gascuel@rennes.inra.fr

Abstract. Rule induction has attracted a great deal of attention in Machine Learning and Data Mining. However, generating rules is not an end in itself because their applicability is not so straightforward. Indeed, the user is often overwhelmed when faced with a large number of rules.

In this paper, we propose an approach to lighten this burden when the user wishes to exploit such rules to decide which actions to do given an unsatisfactory situation. The method consists in comparing a situation to a set of classification rules. This is achieved using a suitable distance thus allowing to suggest action recommendations with minimal changes to improve that situation. We propose the algorithm DAKAR for learning action recommendations and we present an application to an environmental protection issue. Our experiment shows the usefulness of our contribution in decision-making but also raises concerns about the impact of the redundancy of a set of rules in learning action recommendations of quality.

Keywords: Decision support, actionability, rule-based classifier, generalized Minkowski metrics, maximally discriminant descriptions.

1 Introduction

Rule induction has attracted a lot of attention in Machine Learning and Data Mining communities. However, the exploitation of a set of rules induced is usually and merely let to the end-user. Overwhelmed by the number of rules, the user is also often frustrated because the applicability of these rules is not so immediate. This makes the post-analysis of induced rules as a great challenge and a necessary step to assist the user in his work as in decision making. Indeed, rather than simply presenting the rules listing to decision-makers, the ideal is to translate these rules into feasible and concrete actions.

In this paper, we are interested in rule-based classifiers. Although useful in real applications, classifying new instances is not enough; user needs more help when he tries to use the classification model beyond the prediction purpose. It is the case when the user wishes to decide which action to accomplish in order

to improve an unsatisfactory situation. For instance, how to cure an ill patient given a set of rules describing ill and not ill patients. This brings us to the notion of actionability, described in [21] as follows : “a pattern is interesting to the user if the user can do something about it; that is the user can react to it to his or her advantage”. Although some recent works have addressed the problem of learning actionable knowledge [4,5,8,9,13,17,18,23], in our point of view, this topic remains under investigation and deserves more attention.

The purpose of this paper is to develop a new method for recommending actions. We propose the algorithm DAKAR (Discovery of Actionable Knowledge And Recommendations) which works as follows : starting from an unsatisfactory situation and relying on a set of classification rules, DAKAR discovers a set of action recommendations that propose minimal changes to the domain-expert in order to improve that situation.

More precisely, we focus on propositional frameworks, where a situation is expressed by a conjunction of attribute-value pairs. In our case, an action is a modification of values of some attributes of the situation. We compute actions involving “little” changes in the initial situation. This is achieved in DAKAR thanks to the generalized Minkowski metric proposed in [7] which allows us to compute the distance between two descriptions. In our approach, a weight is assigned to each feature in order to take into account how flexible it is. The search space of actions is defined as the set of *maximally discriminant descriptions* that differentiates an unsatisfactory situation from a set of classification rules characterizing what is satisfactory, according to a distance threshold δ . This space is explored considering two properties we have defined, which are the *coherence* and the *validity* of actions. DAKAR uses a beam-search strategy to find the best actions, according to a quality criterion, to suggest to the user.

This paper is organized as follows: in section 2, we present the state of the art of the actionability issue. Section 3 is devoted to our approach and to DAKAR algorithm. Experimental tests are described in section 4 and we conclude in section 5 with a summary of our contribution and future directions.

2 Related Work

This section is a survey of the approaches developed in both Machine Learning and Data Mining literature to address the actionability issue. In [21], Silberschatz and Tuzhilin evoked for the first time the term *actionability* in the context of interestingness measures for patterns evaluation. They classify such measures into objective (data-driven) and subjective (user-driven) measures. According to them, from the subjective point of view, a pattern is interesting if it is :

- actionable : the end-user can act on it to his advantage.
- unexpected : the end-user is surprised by such finding.

As pointed out by the authors, actionability is difficult to capture; they propose rather to capture it through unexpectedness, arguing that unexpected patterns are those that lead the expert to make some actions.

While many works have addressed the unexpectedness issue (see for instance [3] for a survey), the actionability remains to be further investigated even if we noticed recently a great interest from researchers in developing new methods for the discovery of actionable knowledge.

We will focus in the following on the methods developed to address actionability issue by using classifiers. Readers interested in actionability in other tasks, such as clustering, can consult the state of the art proposed by He et al. [5].

In [10], the authors propose a method that prune all non actionable rules from a large set of association rules. The idea is to discard the more general rules using their specialized rules (having more conditions in the left-hand side) with higher quality. Although their approach facilitate the hard task of analyzing large sets of rules, strictly speaking, it does not identify truly action rules.

Lavrač et al. [8] used subgroup discovery to generate actionable knowledge. It is interesting to notice how the authors distinguished actionable rules from operational ones. Operational rules are somewhat actionable “ready to use” rules, which, if applied, will affect a target population immediately.

Shapiro and Matheus [13] developed a health-care system named KEFIR dealing with *deviations*. The system embodies a *recommendation generator* that suggests corrective actions in response to some relevant deviations discovered in the data. Here, corrective actions are defined *a priori* by health-care experts.

A decision-theoretic framework evaluating classification systems from an economic point of view is proposed in [4]. Here, the relationship between the quality of a classification system and the expected payoff to the company is formalized. Classification systems are compared, using their confusion matrices, in terms of their *effectiveness* in decision making. Actions suggested are predefined by the user.

Ling et al. [9] consider mining *optimal actions* in CRM (Customer Relationship Management) relying on decision trees models. They aim at finding those actions that change customers from undesired status to a desired one. An action is merely a change in the value of an attribute. These actions are chosen so as to maximize the expected net profit by taking into consideration their costs.

In [17], the authors show how to discover action rules that allow to move from a given class to another one. This is achieved by comparing pairs of rules having different classes in the right-hand side. Features are divided into: stable (features that cannot be changed) and flexible. An action rule involves only flexible features and shows what are the changes to deploy in order to realize the class change. In business applications, action rules are useful to identify customers for whom some changes in their flexible features will bring them from a profit ranking group to a better one.

The two first approaches [10,8] work directly on rules to address the actionability issue while the others address it via action recommendations. The studies proposed in [13,4] aim at recommending predefined actions to the user and the systems proposed in [9,17] discover actions to be recommended. Comparisons between our approach and the closest works [9,17] are given throughout this paper.

3 Framework Description

3.1 The Learning Task

We propose an approach for mining actionable knowledge. The task we address aims at improving a given situation with regard to a rule-based classifier. It can be defined informally as follows :

“Given a situation for which an unsatisfactory class \ominus corresponds , what are the minimal changes (actions) to do in this situation in order to improve that situation to a satisfactory class \oplus ?”

For instance, given a situation describing a high level pollution in an area, what are the advices to be suggested to reduce that pollution.

In this paper, we propose to discover actions through a set of classification rules. Before giving the algorithm DAKAR, let us first introduce some basic definitions. Let X_1, \dots, X_n be features taking their values in the domains Dom_1, \dots, Dom_n respectively.

Definition 1. (Instance) *An instance is an object described by a conjunction of instantiated features denoted by: $\bigwedge_{i=1, \dots, n} (X_i = v_i)$.*

Definition 2. (Description) *A description is a conjunction defined on a subset of features as follows: $D = \bigwedge_{i=1, \dots, m} (X_{k_i} \in d_{k_i})$*

where $d_{k_i} \subseteq Dom_{k_i}$, $\{k_1, \dots, k_m\} \subseteq \{1, \dots, n\}$ and $k_i \neq k_j \forall i, j$.

Definition 3. (Extended description) *A description can be extended to all the features as follows: $\widehat{D} = \bigwedge_{i=1, \dots, m} (X_{k_i} \in d_{k_i}) \bigwedge_{j \notin \{k_1, \dots, k_m\}} (X_j \in Dom_j)$.*

where $d_{k_i} \subseteq Dom_{k_i}$, $\{k_1, \dots, k_m\} \subseteq \{1, \dots, n\}$.

For the rest of the paper, a description is considered in its extended form and actions and situations are extended descriptions.

Definition 4. (Classification rule) *A classification rule is an implication of the form $Descr \implies Class$ where $Descr$ is a description and $Class$ is the corresponding class label (\ominus or \oplus).*

Definition 5. (Coverage) *We say that a description $D = \bigwedge_{i=1, \dots, n} (X_i \in d_i)$*

covers an instance $I = \bigwedge_{i=1, \dots, n} (X_i = v_i)$ iff $\forall i, v_i \in d_i$. The set of instances (among all possible instances) covered by D will be denoted by $cov(D)$.

Notice that $cov(D) = cov(\widehat{D})$. For a classification rule $R : Descr \implies Class$, we define $cov(R) = cov(Descr)$.

A set of classification rules \mathcal{R} is seen as the union of two sets : \mathcal{R}^\oplus and \mathcal{R}^\ominus corresponding respectively to the set of rules characterizing class \oplus and class \ominus . This can be extended easily to multi-class problems.

Definition 6. (Outcome situation) An action applied to a situation leads to another description called an outcome situation.

Given a situation $S = \bigwedge_{i=1,\dots,n} (X_i \in s_i)$ and an action $A = \bigwedge_{i=1,\dots,n} (X_i \in a_i)$, the outcome situation $outcome(S, A)$ is computed as follows :

$$outcome(S, A) = \bigwedge_{i=1,\dots,n} (X_i \in o_i) \quad \text{where } o_i = \begin{cases} s_i & \text{if } a_i = Dom_i \\ a_i & \text{otherwise} \end{cases}$$

When proposing actions to the user, we have to take into consideration their practical applicability. This can be achieved by considering the *flexibility* of the features involved in the action. This notion is further explained in the following.

3.2 On the Need of a Suitable Distance Metric

In [17], the notion of feature flexibility is used to find action rules. Features are divided into two groups : *stable* (features that cannot be changed) and *flexible*. In their approach, action rules are designed with flexible features.

This division into two groups is interesting but is rather strict and not always sufficient. For example, a physician would prefer, if possible, prescribing medicines to advising a surgical intervention to an ill patient. However, in their work, such two flexible features are identically considered. In our approach, to take into account this differentiation, each feature is assigned with a given *flexibility weight*. Moreover, the notion of feature flexibility is not sufficient. For example, making a diet is in many cases an advice given by physicians but its feasibility depends on the goal we want to achieve. If the weight to loose is about 80 kilograms, this advice is impractical. In the other hand, the advice of loosing 2 kilograms is quite practical. The outcome situation (when action is applied) must be relatively “close” to the initial situation. For these reasons, a metric distance capturing the difficulty of improving a situation thanks to an action seems to be adapted to our problem.

The literature abounds with definitions of metric distances between two descriptions. Relying on an empirical comparative study [11], we have chosen the generalized Minkowski metric proposed by Ichino and Yaguchi [7] which handles both qualitative and quantitative features. It also integrates weights and deals with dissimilarities between two feature values. We rewrite it according to our notations. Let us consider two extended descriptions :

$$\begin{aligned} D &= \bigwedge_{1..n} (X_i \in d_i) & d_i &\subseteq Dom_i \\ D' &= \bigwedge_{1..n} (X_i \in d'_i) & d'_i &\subseteq Dom_i \end{aligned}$$

The normalized and weighted dissimilarity measure called the generalized Minkowski distance of order p defined in [7] is given by :

$$d_p(D, D') = \left[\sum_{i=1}^n \{w_i \psi(d_i, d'_i)\}^p \right]^{1/p}$$

where the flexibility weights $w_i > 0, i \in \{1, \dots, n\}$ are chosen so that $\sum_{i=1}^n w_i = 1$ and $\psi(d_i, d'_i)$ is a normalized distance between the sub-domains d_i and d'_i of

Dom_i (see [7] for more details). The generalized Minkowski distance satisfies $0 \leq d_p(D, D') \leq 1$ and it is proved that this distance satisfies all the axioms for a metric.

3.3 The Search Space of Actions

In this subsection, we define the search space of the actions. We are given an initial situation $S = \bigwedge_{i=1,\dots,n} (A_i \in s_i)$ classified in \ominus and a set of rules $\mathcal{R} = \mathcal{R}^\oplus \cup \mathcal{R}^\ominus$. An obvious approach is to define the search space of actions as the whole space of descriptions. This space is first restricted thanks to the property of *validity* defined as follows.

Definition 7. (The validity of an action relatively to a situation) An action A is said to be valid relatively to a situation S if

$$\text{cov}(S) \cap \text{cov}(A) = \{\}$$

Thus, an instance covered by the initial situation is not covered by the outcome situation when a valid action is applied. To consider only valid actions, we use the *maximally discriminant set* denoted by $discr$ which is defined for S and a description $D = \bigwedge_{i=1,\dots,n} (A_i \in d_i)$ by:

$$discr(S, D) = \bigcup_{i=1,\dots,n} \{A_i \in (d_i - s_i)\}$$

For a rule $R : Descr \implies Class$, we define $discr(S, R) = discr(S, Descr)$. Notice that an element in $discr(S, D)$ is a valid action relatively to S . The notion of maximally discriminant has already been used in [19] in designing a learner inspired by the version spaces framework [12].

In order to consider descriptions that are likely to be “good” actions, we construct the maximally discriminant set between S and the description part of one rule in \mathcal{R}^\oplus . Intuitively, this set is composed of the attribute-value pairs that make the difference between the rules of class \oplus and the situation of class \ominus .

Example 1. Let $\{weight, medicines\}$ be a set of features taking respectively their values in $Dom_1 = [40, 120]$ and $Dom_2 = \{no, tablets, syrup\}$. The quantitative feature *weight* represents the weight of a patient while the qualitative feature *medicines* represents the treatment prescribed to the patient. Let R_1^\oplus , R_2^\oplus and R_3^\ominus be three rules:

$$\begin{aligned} R_1^\oplus &: weight \in [50, 80] \rightarrow \text{not ill} \\ R_2^\oplus &: weight \in [90, 110] \wedge medicines \in \{syrup\} \rightarrow \text{not ill} \\ R_3^\ominus &: weight \in [65, 120] \wedge medicines \in \{no\} \rightarrow \text{ill} \end{aligned}$$

We consider an ill patient in the following situation S :

$$weight \in [70] \wedge medicines \in \{no\}.$$

We compute maximally discriminant sets:

$$\begin{aligned} discr(S, R_1^\oplus) &= \{weight \in [50, 70[, weight \in]70, 80], \\ &\quad medicines \in \{tablets, syrup\}\} \\ discr(S, R_2^\oplus) &= \{weight \in [90, 110], medicines \in \{syrup\}\} \end{aligned}$$

In [18], an action rule relies on only one rule. For each rule in \mathcal{R}^\oplus corresponds an action which guarantees to get an outcome situation with a better class (at least

one rule of \mathcal{R}^\oplus covers the outcome situation). In our point of view, an action should rely on the entire set of rules. Thus, by combining attribute-value pairs of these rules, new actions can be suggested.

The set of elementary actions \mathcal{A} is given by $\bigcup_{R \in \mathcal{R}^\oplus} \text{discr}(S, R)$ and an action is a conjunction of such elements. Our search space of actions is the set :

$$\{\bigwedge_{\text{elem} \in E} \text{elem} \mid E \subseteq \mathcal{A}\}$$

Actions considered by our method embodies those that would be suggested by the system in [18]. Let us also notice that the search space as defined above embodies some actions that are not to be considered because they do not fulfill the *coherency* property defined as follows :

Definition 8. (The coherency of an action) *An action A is said to be coherent if $\text{cov}(A) \neq \{\}$.*

Example 2. In the Example 1, the set of elementary actions \mathcal{A} is $\{\text{weight} \in [50, 70[, \text{weight} \in]70, 80], \text{medicines} \in \{\text{tablets, syrup}\}, \text{weight} \in [90, 110], \text{medicines} \in \{\text{syrup}\}\}$. An action is a conjunction composed by elements of a subset of \mathcal{A} . We have to discard the action $\text{weight} \in [50, 70[\wedge \text{weight} \in [90, 110]$ equivalent to $\text{weight} \in []$ which is not coherent.

In practice, an action is not coherent if it involves an empty value for at least one attribute. Let us emphasize that applying an action to a situation does not guarantee to get an outcome situation with a better class than the initial situation. That is why, we need a criterion for assessing the quality of actions. We classify the outcome situation using the set of rules. We rely on the confidence of the rules covering the outcome situation O to evaluate the quality of an action A . We defined the *quality* measure by :

$$\text{quality}(A) = \sum_{R \in \mathcal{R}^\oplus, \text{cov}(O) \subseteq \text{cov}(R)} \text{conf}(R) - \sum_{R \in \mathcal{R}^\ominus, \text{cov}(O) \subseteq \text{cov}(R)} \text{conf}(R)$$

where $O = \text{outcome}(S, A)$ and where $\text{conf}(R)$ is the usual confidence¹ of the rule R . Notice that $\text{quality}(A) > 0$ means that O is covered by at least one rule in \mathcal{R}^\oplus .

Moreover, as pointed out in subsection 3.2, we favor actions that involve a little change in the initial situation. This is achieved by verifying that the distance between the outcome situation (obtained when an action is applied to the initial situation) and the initial situation itself does not exceed a parameter δ given by the user. We term such privileged actions δ -cost actions.

All the requirements for explaining DAKAR algorithm are now available.

3.4 DAKAR Algorithm

The aim of DAKAR (Discovery of Actionable Knowledge and Recommendations) algorithm is to find the set of the *best* actions. The algorithm explores the search

¹ The confidence of a rule $R : \text{Descr} \implies C$ in a database is the number of examples of class C covered by Descr divided by the total number of examples covered by Descr .

Algorithm 1: Dakar algorithm

Input: - a set of rules $\mathcal{R} = \mathcal{R}^{\oplus} \cup \mathcal{R}^{\ominus}$
 - a situation \mathcal{S}
 - a distance threshold δ
 - a maximal beam size N

Output: - a set of actions

$A = \bigcup_{R \in \mathcal{R}^{\oplus}} \text{discr}(\mathcal{S}, R)$
 $\text{Beam} = \{\}$
 $\text{NewBeam} = \{\text{true}\}$

while $\text{Beam} \neq \text{NewBeam}$ **do**

- $\text{Beam} \leftarrow \text{NewBeam}$
- foreach** $\text{action} \in \text{Beam}$ **do**
 - foreach** $\text{elem} \in A$ **do**
 - $\text{newaction} \leftarrow (\text{action} \wedge \text{elem})$
 - if** newaction coherent and δ -cost **then**
 - $\text{NewBeam} \leftarrow \text{NewBeam} \cup \{\text{newaction}\}$
 - while** $\text{card}(\text{Beam}) > N$ **do**
 - $\text{worstaction} \leftarrow \text{Argmin}_{A \in \text{NewBeam}} \text{quality}(A)$
 - $\text{NewBeam} \leftarrow \text{NewBeam} - \{\text{worstaction}\}$

return sorted Beam

space of actions using a beam search strategy : it maintains a set (called *Beam*) of the best actions the algorithm has constructed up till now. Initially the beam is set to $\{\text{true}\}$, which means that no action is constructed. During the exploration of the search space, DAKAR specializes actions of the beam and keeps only the best ones w.r.t. the criterion we defined earlier. DAKAR is given in algorithm 1.

4 Experiments

We implemented DAKAR in Sicstus Prolog and we conducted an experimental evaluation of our algorithm on an environmental application related to stream-water pollution by pesticides. This application is developed in the context of the project SACADEAU [2]. In our experiments, we used a dataset generated by a model which outputs a class of pollution given information about farming works, climate, soil, etc. The set of attributes and their descriptions is given in Table 1. It is a multi-class application where pollution classes are ordered by experts by taking into account legal thresholds (class 4 is the least satisfactory class). The model is an oracle since it provides the function $\text{simulation} : \text{situation} \rightarrow \text{class}$ and we define the *benefit* of an action A applied to a situation S , by $\text{benefit}(A) = \text{simulation}(S) - \text{simulation}(\text{outcome}(S, A))$. An action A is said *positive* when its benefit is positive.

Our system was tested on 150 unsatisfactory situations (of class 1, 2, 3 or 4). In presented experiments, the size of the beam was 5 ; thus, DAKAR proposed 5 actions for each situation. The distance threshold δ varied from 0 to 0.5.

Table 1. Some attributes of the SACADEAU application and their descriptions

Name	Domain	Flexibility	Description
strat	{pre,post}	0.001	pesticide application strategy of the farmer
molec	{atrazine,new}	0.003	pesticide used by the farmer
hedge	{0%,90%}	0.006	percentage of river border with a hedge
basin	{concave,convex}	0.33	typology of the catchment area
orga_matter	{2%,5%}	0.33	soil composition in organic matter
climate	[1;5]	0.33	wetness of the climate (1:not wet)
class	{0,1,2,3,4}	-	severity of the pollution (0:no pollution)

A set of rules ,with minimal support 20, was generated by ICL [15] (rules are not ordered).

A first evaluation is to show qualitatively the utility of using a distance for recommending actions. Let us give an example of DAKAR execution on a situation of class 4 :

```
strategy=post, molecules=atrazine, hedge=0%,
basin=convex, orga_matter=5%, climate=4
```

The two best actions recommended by DAKAR, with benefit 2 and 1 respectively, are given below :

```
1 - hedge=90%, molec=new, strat=pre      2 - molec=new, strat=pre
   (quality = 1.50 ; distance=0.005)      (quality = 1.35 ; distance=0.002)
```

Both actions suggest to apply pesticides before plants grow up (pre-emergence strategy) and to use new molecules rather than atrazine. The quality of these two actions are almost the same whereas the distance involved by the first action is more than two times the distance involved by the second. Concretely, experts could decide that in a short term, installing a hedge on 90% of the river border is not a necessary action for improving the situation.

Our algorithm uses the distance to take into account a threshold δ , but we can imagine an algorithm using both quality and distance in the criterion as done in [9] where a function of profit is maximized.

A second qualitative evaluation concerns interest of the quality criterion. There is an example of situation of class 3 :

```
strategy=pre, molecules=new, hedge=0%,
basin=convex, orga_matter=2%, climate=3
```

The two best actions recommended by our system (having a benefit of 1 and -1 respectively) are :

```
1 - hedge=90%                               2 - hedge=90%, strat=post
   (quality = 0.64, distance = 0.005)      (quality = -0.56,distance = 0.005)
```

The first action is the only one, among the five recommended actions, to get a positive quality. Installing a hedge on 90% of the river border seems to be the necessary action for improving the situation.

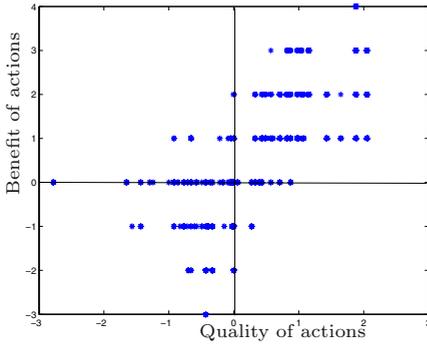


Fig. 1. Actions plotted according to their quality and their effective benefit

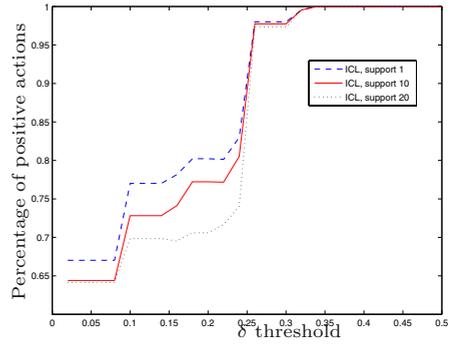


Fig. 2. Percentage of positive recommended actions according to the size of the rule set and the distance parameter δ

From this point, only the action maximizing the quality criterion in the beam are considered. We plotted, in Figure 1, recommended actions according to their quality and their effective benefit. Note that the quality of an action A is a good prediction of the efficiency of A if ($quality(A) < 0$ and $benefit(A) \leq 0$) or ($quality(A) > 0$ and $benefit(A) > 0$).

In Figure 2, the experiment raises another concern about the relationship between the redundancy in a set of classification rules and the efficiency of recommended actions. In [22], the author studied the effect of the redundancy of a classification rule set on the task of predicting a class. In his experiments, a redundant classification rule set has better accuracy on classification.

Using ICL, three classification rule sets were learned with support parameter equal to 1, 10 and 20, leading to set of 19, 35 and 63 rules respectively. We evaluated the three sets of rules, by comparing the efficiency of the actions they suggested when used in DAKAR. We plotted in Figure 2 the efficiency of recommended actions in function of the parameter δ for the three sets of rules.

As expected, the efficiency of actions proposed by DAKAR is globally increasing in the parameter δ . Moreover, we notice that the more the classification rule set is redundant, the more the actions are effective. The impact of the redundancy in rules on Sebag’s distance between two instances expressed by Horn clauses was also pointed out in [20]. Her distance is based on the coverage of instances by a theory and the more the theory is redundant, the more this distance is of interest.

We repeated the same experiment for three rule sets generated by three systems : C4.5 [14], ICL [15] and Apriori² [1]. Supposing that Apriori is more redundant than ICL which is more redundant than C4.5, we compared the efficiency of the actions suggested by the three sets of rules. Once again, the more the set of rules is redundant, the more the actions are effective.

² Note that we constrained the association rules generated by Apriori to contain only the class in their right hand side.

5 Conclusion

In this paper we have investigated the task of learning actionable knowledge and recommendations to the user. We attempted to answer the following question : how to go beyond the simple use of classification rules in prediction by making them actionable?

Our contribution is as follows : given a situation, the algorithm we propose allows the user to further exploit a set of classification rules in order to decide what are the actions to accomplish in order to improve that situation. The algorithm looks for the best actions involving a little change in the initial situation. This is achieved thanks to an actionability approach relying on a *distance*.

Our framework has been applied to an environmental dataset related to pollution. We have learned some actionable knowledge concerning the possible recommendations one can adopt in order to reduce the pollution. Such recommendations take into account the degree of *flexibility* of each feature. Experiments have shown the feasibility of this task. They also raise some concerns about the impact of the redundancy of rules on the quality of actions recommended by our system. The approach we propose can be improved in the following directions. First, extend the framework to first order learners. In this case, we need other kinds of distance metrics handling literals, such as those proposed in [6,16]. Second, study the impact of the chosen distance metric on the quality of the recommended actions.

In our point of view, action recommendation is a promising issue with many practical applications such as healthcare, environmental protection and customer analysis. There is clearly much research to be done in the formalization of the task of learning useful and actionable knowledge from both methods and interestingness measures points of view.

Acknowledgments. We thank INRA for the simulator and Christel Vrain for useful comments on an early draft of this paper.

References

1. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD*, pages 207–216. ACM Press, 1993.
2. M.-O. Cordier. SACADEAU: A decision-aid system to improve stream-water quality. *ERCIM News. Special issue on Environmental Modelling*, (61):35–36, April 2005.
3. B. Duval, A. Salleb, and C. Vrain. Méthodes et mesures d'intérêt pour l'extraction de règles d'exception. *Revue des Nouvelles Technologies de l'Information - Mesures de Qualité pour la Fouille de Données RNTI-E-1*, pages 119–140, 2004.
4. Y. Elovici and D. Braha. A decision-theoretic approach to data mining. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 33(1):42–51, 2003.
5. Z. He, X. Xu, and S. Deng. Data Mining for Actionable Knowledge: A Survey. *ArXiv Computer Science e-prints*, January 2005.

6. A. Hutchinson. Metrics on terms and clauses. In *ECML*, pages 138–145, 1997.
7. M. Ichino and H. Yaguchi. Generalized minkowski metrics for mixed feature-type data analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4):698–708, 1994.
8. N. Lavrač, B. Cestnik, D. Gamberger, and P. Flach. Decision support through subgroup discovery: Three case studies and the lessons learned. *Machine Learning*, 57(1-2):115–143, 2004.
9. C. X. Ling, T. Chen, Q. Yang, and J. Cheng. Mining optimal actions for profitable CRM. In *ICDM*, pages 767–770, 2002.
10. Bing Liu, Wynne Hsu, and Yiming Ma. Identifying non-actionable association rules. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 329–334, New York, NY, USA, 2001. ACM Press.
11. D. Malerba, F. Esposito, V. Gioviiale, and V. Tamma. Comparing dissimilarity measures in symbolic data analysis. In *Joint Conferences on "New Techniques and Technologies for Statistics" and "Exchange of Technology and Know-how" (ETK-NTTS'01)*, pages 473–481, 2001.
12. T. M. Mitchell. Generalization as search. *Artif. Intell.*, 18(2):203–226, 1982.
13. G. Piatetsky-Shapiro and C. Matheus. The interestingness of deviations. In *AAAI Workshop on Knowledge Discovery in Databases*, pages 25–36, Menlo Park, CA, 1994. AAAI Press.
14. J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
15. L. De Raedt and W. Van Laer. Inductive constraint logic. In *ALT '95: Proceedings of the 6th International Conference on Algorithmic Learning Theory*, pages 80–94, London, UK, 1995. Springer-Verlag.
16. J. Ramon, M. Bruynooghe, and W. Van Laer. Distance measures between atoms. In *CompulogNet Area Meeting on Computational Logic and Machine Learning*, pages 35–41. University of Manchester, UK, May 1998.
17. Z. W. Ras and L.-S. Tsay. Discovering extended action-rules (system dear). In *IIS*, pages 293–300, 2003.
18. Z. W. Ras and A. Wiczorkowska. Action-rules: How to increase profit of a company. In *PKDD*, pages 587–592, 2000.
19. M. Sebag. Delaying the choice of bias: A disjunctive version space approach. In *ICML*, pages 444–452, 1996.
20. M. Sebag. Distance induction in first order logic. In *ILP '97: Proceedings of the 7th International Workshop on Inductive Logic Programming*, pages 264–272. Springer-Verlag, 1997.
21. A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. On Knowledge And Data Engineering*, 8:970–974, 1996.
22. L. Torgo. Controlled redundancy in incremental rule learning. In *ECML*, pages 185–195, 1993.
23. Q. Yang, J. Yin, C. X. Ling, and T. Chen. Postprocessing decision trees to extract actionable knowledge. In *ICDM*, pages 685–688, 2003.