Representing Case Variations for Learning General and Specific Adaptation Rules

Fadi Badra, Jean Lieber

Representing Case Variations for Learning General and Specific Adaptation Rules

Fadi BADRA [a] and Jean LIEBER [a]
[a] *LORIA (UMR 7503 CNRS–INPL–INRIA-Nancy 2–UHP)*
BP 239, 54 506 Vandœuvre-lès-Nancy, FRANCE
email: {badra,lieber}@loria.fr

Abstract. Adaptation is a task of case-based reasoning systems that is largely domain-dependant. This motivates the study of adaptation knowledge acquisition (AKA) that can be carried out thanks to learning processes on the variations between cases of the case base. This paper studies the representation of these variations and the impact of this representation on the AKA process, through experiments in an oncology domain.

Introduction

Case-based reasoning (CBR [7,9]) aims at solving a target problem thanks to a case base. A case is the description of a problem-solving episode which can be generally seen as a pair (problem,solution). A CBR system selects a case from the case base and then adapts the associated solution. Adaptation is a difficult task since it requires domain-dependant knowledge that needs to be acquired. The goal of adaptation knowledge acquisition is to detect and extract this knowledge. Some approaches have proved successful in learning such adaptation knowledge from the case base [2,3,6].

Our hypothesis is that this learning task can be improved by choosing an appropriate representation of the variations between cases in the case base. Introducing general knowledge on variations enables the extraction of different types of adaptation rules. In particular, a limited number of general adaptation rules may be presented to the analyst for validation. Afterwards, these rules are helpful to structure a set of specific adaptation rules.

The paper is organized as follows. Section 1 defines general notions on CBR that are used in the rest of the paper. The AKA approach of [6] is summarized in section 2. It involves some knowledge representation and learning issues — representing and learning variations and adaptation rules— that are addressed in section 3 independently of the case representation formalism. Then, this general framework is applied to an attribute-constraint formalism for representing cases (section 4). Section 5 presents some experiments in an oncology domain to validate our hypothesis on the usefulness of choosing an appropriate representation of the variations. Section 6 discusses this work by comparing it to related work. Finally, section 7 concludes and points out some future work.

1. Definitions

Let $\mathcal{L}_{\mathtt{pb}}$ and $\mathcal{L}_{\mathtt{sol}}$ be two languages. A problem (resp. a solution) is by definition an element of $\mathcal{L}_{\mathtt{pb}}$ (resp. of $\mathcal{L}_{\mathtt{sol}}$). The existence of a binary relation $\rightsquigarrow$ on $\mathcal{L}_{\mathtt{pb}} \times \mathcal{L}_{\mathtt{sol}}$ is assumed, but not completely known in general. $\mathtt{Sol(pb)}$ is a solution of $\mathtt{pb}$ if $\mathtt{pb} \rightsquigarrow \mathtt{Sol(pb)}$. A case is a pair $(\mathtt{pb}, \mathtt{Sol(pb)})$ such that $\mathtt{pb} \rightsquigarrow \mathtt{Sol(pb)}$. The case base is the finite set of available cases, called source cases and denoted by $(\mathtt{srce}, \mathtt{Sol(srce)})$. CBR aims at solving a *target problem* $\mathtt{tgt}$ thanks to a case base. It consists in general in retrieving a source case $(\mathtt{srce}, \mathtt{Sol(srce)})$ such that $\mathtt{srce}$ is judged similar to $\mathtt{tgt}$ and in adapting this retrieved case in order to solve $\mathtt{tgt}$: $\mathtt{Adaptation}$: $(\mathtt{srce}, \mathtt{Sol(srce)}, \mathtt{tgt}) \mapsto \mathtt{Sol(tgt)}$ where $(\mathtt{srce}, \mathtt{Sol(srce)}, \mathtt{tgt})$ is an *adaptation problem* and $\mathtt{Sol(tgt)}$ is a candidate solution for $\mathtt{tgt}$ (the relation $\mathtt{tgt} \rightsquigarrow \mathtt{Sol(tgt)}$ is not ensured since CBR is not a deductive reasoning). The adaptation step is based on adaptation knowledge that has to be acquired, which constitutes the adaptation knowledge acquisition issue.

2. AKA Principles

This section reformulates the main principle of the AKA approach proposed in [6]. The adaptation process is assumed to be composed of three steps:

1. $\mathtt{matching}$: $(\mathtt{srce}, \mathtt{tgt}) \mapsto \Delta\mathtt{pb}$. A representation $\Delta\mathtt{pb}$ of the variations from $\mathtt{srce}$ to $\mathtt{tgt}$ is computed.
2. AK : $\Delta\mathtt{pb} \mapsto \Delta\mathtt{sol}$. The adaptation knowledge AK enables to build a representation $\Delta\mathtt{sol}$ of solution variations.
3. $\mathtt{modifying}$: $(\mathtt{Sol(srce)}, \Delta\mathtt{sol}) \mapsto \mathtt{Sol(tgt)}$ such that $\mathtt{matching}(\mathtt{Sol(srce)}, \mathtt{Sol(tgt)}) = \Delta\mathtt{sol}$. $\Delta\mathtt{sol}$ represents the variations from $\mathtt{Sol(srce)}$ to the unknown solution $\mathtt{Sol(tgt)}$ and thus enables to infer $\mathtt{Sol(tgt)}$ from $\mathtt{Sol(srce)}$.

Step 2 requires some adaptation knowledge. Conversely, if a set of pairs $(\Delta\mathtt{pb}, \Delta\mathtt{sol})$ is available, some machine learning techniques may be used to learn AK. The idea is to exploit pairs of source cases to obtain such pairs of variations. If $(\mathtt{srce}_k, \mathtt{Sol}(\mathtt{srce}_k))$ and $(\mathtt{srce}_\ell, \mathtt{Sol}(\mathtt{srce}_\ell))$ are two source cases, let $\Delta\mathtt{pb}_{k\ell} = \mathtt{matching}(\mathtt{srce}_k, \mathtt{srce}_\ell)$ and $\Delta\mathtt{sol}_{k\ell} = \mathtt{matching}(\mathtt{Sol}(\mathtt{srce}_k), \mathtt{Sol}(\mathtt{srce}_\ell))$. Then, AK is learned with the training set $\{(\Delta\mathtt{pb}_{k\ell}, \Delta\mathtt{sol}_{k\ell})\}_{k\ell}$.

3. Representing and Learning Adaptation Rules

In order to make this AKA approach operational, it is necessary to be able to represent the variations $\Delta\mathtt{pb}$ and $\Delta\mathtt{sol}$ and to infer from ordered pairs of problems (resp., of solutions) $\Delta\mathtt{pb}$ (resp., $\Delta\mathtt{sol}$), which constitutes the matching process.

3.1. *Representing Variations*

A variation from a problem to another is the representation of a binary relation $\mathtt{r}$ between problems. The language of such relations $\mathtt{r}$ is denoted by $\mathcal{L}_{\Delta\mathtt{pb}}$. Thus, the semantics of any $\mathtt{r} \in \mathcal{L}_{\Delta\mathtt{pb}}$ is given by its extension: $\text{Ext}(\mathtt{r}) \subseteq \mathcal{L}_{\mathtt{pb}} \times \mathcal{L}_{\mathtt{pb}}$. $\mathcal{L}_{\Delta\mathtt{pb}}$ is assumed to contain the relation $\top_{\Delta\mathtt{pb}}$ and the relation $\langle\mathtt{srce},\mathtt{tgt}\rangle$ for each ordered pair $(\mathtt{srce},\mathtt{tgt}) \in \mathcal{L}_{\mathtt{pb}} \times \mathcal{L}_{\mathtt{pb}}$. The semantics of these elements is: $\text{Ext}(\top_{\Delta\mathtt{pb}}) = \mathcal{L}_{\mathtt{pb}} \times \mathcal{L}_{\mathtt{pb}}$ and $\text{Ext}(\langle\mathtt{srce},\mathtt{tgt}\rangle) = \{(\mathtt{srce},\mathtt{tgt})\}$.

Let $\vDash$ be the entailment relation on $\mathcal{L}_{\Delta\mathtt{pb}}$ (if $\mathtt{r},\mathtt{s} \in \mathcal{L}_{\Delta\mathtt{pb}}$, $\mathtt{r} \vDash \mathtt{s}$ means that $\text{Ext}(\mathtt{r}) \subseteq \text{Ext}(\mathtt{s})$) and $\equiv$ be the equivalence relation on $\mathcal{L}_{\Delta\mathtt{pb}}$ ($\mathtt{r} \equiv \mathtt{s}$ iff $\mathtt{r} \vDash \mathtt{s}$ and $\mathtt{s} \vDash \mathtt{r}$). $\vDash$ (and thus $\equiv$) is assumed to be computable. For $\mathtt{r} \in \mathcal{L}_{\Delta\mathtt{pb}}$, let $\mathtt{DC}(\mathtt{r}) = \{\mathtt{s} \in \mathcal{L}_{\Delta\mathtt{pb}} \mid \mathtt{r} \vDash \mathtt{s}\}$ be the *deductive closure* of $\mathtt{r}$.

Matching two problems $\mathtt{srce}$ and $\mathtt{tgt}$ aims at identifying the relations $\mathtt{r} \in \mathcal{L}_{\Delta\mathtt{pb}}$ relating $\mathtt{srce}$ to $\mathtt{tgt}$, that is, the set $\mathtt{DC}(\langle\mathtt{srce},\mathtt{tgt}\rangle)$. Since any relation $\mathtt{r} \in \mathcal{L}_{\Delta\mathtt{pb}}$ relating $\mathtt{srce}$ and $\mathtt{tgt}$ can be deduced from the relation $\langle\mathtt{srce},\mathtt{tgt}\rangle$, matching is defined as:

$$\mathtt{matching}(\mathtt{srce},\mathtt{tgt}) = \langle\mathtt{srce},\mathtt{tgt}\rangle$$

The representation of solution variations is similar: $\mathcal{L}_{\Delta\mathtt{sol}}$ is the solution variation language containing $\top_{\Delta\mathtt{sol}}$ and $\langle\mathtt{Sol}(\mathtt{srce}),\mathtt{Sol}(\mathtt{tgt})\rangle$; $\vDash$, $\equiv$ and $\mathtt{DC}$ are defined similarly on $\mathcal{L}_{\Delta\mathtt{sol}}$; finally, $\mathtt{matching}(\mathtt{Sol}(\mathtt{srce}),\mathtt{Sol}(\mathtt{tgt})) = \langle\mathtt{Sol}(\mathtt{srce}),\mathtt{Sol}(\mathtt{tgt})\rangle$.

The Symmetry Hypothesis. The symmetry hypothesis is optional in our framework. It states that for each $\mathtt{r} \in \mathcal{L}_{\Delta\mathtt{pb}}$, there exists $\mathtt{r}^{-1} \in \mathcal{L}_{\Delta\mathtt{pb}}$ such that, for each $(\mathtt{srce},\mathtt{tgt}) \in \mathcal{L}_{\mathtt{pb}} \times \mathcal{L}_{\mathtt{pb}}$, $(\mathtt{srce},\mathtt{tgt}) \in \text{Ext}(\mathtt{r})$ iff $(\mathtt{tgt},\mathtt{srce}) \in \text{Ext}(\mathtt{r}^{-1})$. In particular, ${\top_{\Delta\mathtt{pb}}}^{-1} \equiv \top_{\Delta\mathtt{pb}}$ and $\langle\mathtt{srce},\mathtt{tgt}\rangle^{-1} \equiv \langle\mathtt{tgt},\mathtt{srce}\rangle$. A similar hypothesis can be made on $\mathcal{L}_{\Delta\mathtt{sol}}$.

3.2. *Representing Adaptation Rules*

An *adaptation rule* is a piece of knowledge that can be used to solve adaptation problems $(\mathtt{srce},\mathtt{Sol}(\mathtt{srce}),\mathtt{tgt}) \in \mathcal{L}_{\mathtt{pb}} \times \mathcal{L}_{\mathtt{sol}} \times \mathcal{L}_{\mathtt{pb}}$: it aims at giving pieces of information about a solution $\mathtt{Sol}(\mathtt{tgt})$. The adaptation rules studied in this paper are ordered pairs $(\mathtt{r},\mathtt{R}) \in \mathcal{L}_{\Delta\mathtt{pb}} \times \mathcal{L}_{\Delta\mathtt{sol}}$ that can be interpreted as follows:

$$\begin{array}{ll} \textbf{if} & \langle\mathtt{srce},\mathtt{tgt}\rangle \vDash \mathtt{r} \\ \textbf{then} & \mathtt{Sol}(\mathtt{tgt}) \text{ is such that } \langle\mathtt{Sol}(\mathtt{srce}),\mathtt{Sol}(\mathtt{tgt})\rangle \vDash \mathtt{R} \end{array} \tag{1}$$

If there exists at most one $\mathtt{Sol}(\mathtt{tgt}) \in \mathcal{L}_{\mathtt{sol}}$ such that (1) holds for any adaptation problem $(\mathtt{srce},\mathtt{Sol}(\mathtt{srce}),\mathtt{tgt})$ then the adaptation rule $(\mathtt{r},\mathtt{R})$ is said to be *specific*. Otherwise, it is a *general* adaptation rule that is not always sufficient to solve an adaptation problem. Given two adaptation rules $\mathtt{AR}_1 = (\mathtt{r}_1,\mathtt{R}_1)$ and $\mathtt{AR}_2 = (\mathtt{r}_2,\mathtt{R}_2)$, $\mathtt{AR}_1$ is said to be less general than $\mathtt{AR}_2$ —denoted by $\mathtt{AR}_1 \vDash \mathtt{AR}_2$— if $\mathtt{r}_1 \vDash \mathtt{r}_2$ and $\mathtt{R}_1 \vDash \mathtt{R}_2$. This means that $\mathtt{AR}_1$ can be applied on less adaptation problems but is more accurate in the sense that the constraint on $\mathtt{Sol}(\mathtt{tgt})$ is stronger. The most general adaptation rule is $(\top_{\Delta\mathtt{pb}}, \top_{\Delta\mathtt{sol}})$. Given an adaptation rule $\mathtt{AR}$, the deductive closure of $\mathtt{AR}$ is the set $\mathtt{DC(AR)}$ of adaptation rules such that each $\mathtt{AR}' \in \mathtt{DC(AR)}$ is more general than $\mathtt{AR}$ (i.e., $\mathtt{AR} \vDash \mathtt{AR}'$).

Under the symmetry hypothesis (on $\mathcal{L}_{\Delta\mathtt{pb}}$ and on $\mathcal{L}_{\Delta\mathtt{sol}}$), an inverse rule $\mathtt{AR}^{-1} = (\mathtt{r}^{-1}, \mathtt{R}^{-1})$ can be associated to each rule $\mathtt{AR} = (\mathtt{r}, \mathtt{R})$.

3.3. Learning Adaptation Rules

As stated in section 2, AKA consists in using as training set $\mathtt{TS}$ a set of pairs $(\Delta\mathtt{pb}_{k\ell}, \Delta\mathtt{sol}_{k\ell})$ with

$$\Delta\mathtt{pb}_{k\ell} = \langle \mathtt{srce}_k, \mathtt{srce}_\ell \rangle \in \mathcal{L}_{\Delta\mathtt{pb}}$$
$$\Delta\mathtt{sol}_{k\ell} = \langle \mathtt{Sol}(\mathtt{srce}_k), \mathtt{Sol}(\mathtt{srce}_\ell) \rangle \in \mathcal{L}_{\Delta\mathtt{sol}}$$

Thus, an ordered pair $\mathtt{AR}_{k\ell} = (\Delta\mathtt{pb}_{k\ell}, \Delta\mathtt{sol}_{k\ell}) \in \mathtt{TS}$ is a specific adaptation rule that solves only the adaptation problem $(\mathtt{srce}_k, \mathtt{Sol}(\mathtt{srce}_k), \mathtt{srce}_\ell)$ in a solution $\mathtt{Sol}(\mathtt{srce}_\ell)$. AKA consists in highlighting some adaptation rules $(\mathtt{r}, \mathtt{R})$ that are more general than a "large" number of elements of $\mathtt{TS}$. More formally, with $\mathtt{AR}$ an adaptation rule, let

$$\mathtt{support(AR)} = \frac{\mathtt{card}\ \{\mathtt{AR}_{k\ell} \in \mathtt{TS} \mid \mathtt{AR}_{k\ell} \vDash \mathtt{AR}\}}{\mathtt{card\ TS}}$$

Given a uniform distribution of probability on $\mathtt{TS}$, $\mathtt{support(AR)}$ is the probability that a random variable on $\mathtt{TS}$ entails $\mathtt{AR}$. Learning adaptation rules aims at finding the $\mathtt{AR} = (\mathtt{r}, \mathtt{R}) \in \mathcal{L}_{\Delta\mathtt{pb}} \times \mathcal{L}_{\Delta\mathtt{sol}}$ such that $\mathtt{support(AR)} \geq \sigma_s$, where $\sigma_s \in [0; 1]$ is a learning parameter called the support threshold.

It can be noticed that if $\mathtt{AR}_1 \vDash \mathtt{AR}_2$ then $\mathtt{support}(\mathtt{AR}_1) \leq \mathtt{support}(\mathtt{AR}_2)$. Therefore, a presentation of the learned adaptation rules by decreasing support, starting from $(\top_{\Delta\mathtt{pb}}, \top_{\Delta\mathtt{sol}})$ whose support is 1, presents any adaptation rule before all the adaptation rules that are more specific than it.

Under the symmetry hypothesis, $\mathtt{support}(\mathtt{AR}^{-1}) = \mathtt{support(AR)}$ for each adaptation rule $\mathtt{AR}$. It is suggested that the two rules $\mathtt{AR}$ and $\mathtt{AR}^{-1}$ are presented together to an expert for validation.

4. Application to an Attribute-Constraint Formalism

In this section, the general framework described above is applied to an attribute-constraint formalism that extends the attribute-value formalism frequently used in CBR [7].

4.1. Representing Cases

A problem instance is described by the values it takes for some attributes. Problems are defined by specifying some sets of values these attributes may range over. For example, the problem in the domain of breast cancer treatment,

$$\mathtt{pb} = (\mathtt{gender}, \{\mathtt{female}\}) \wedge (\mathtt{age}, [50; +\infty[) \wedge (\mathtt{s}, [4; 7[)$$

represents the class of women older than 50 and for which the tumor size $\mathtt{s}$ is such that $4 \leq \mathtt{s} < 7$ cm.

More formally, a *problem instance* is an element of the Cartesian product $\mathtt{Inst}_{\mathtt{pb}} = \mathtt{V}_1 \times \cdots \times \mathtt{V}_m$, where $\mathtt{V}_i$ is a set of values. In the examples, $\mathtt{V}_i$ is assumed to be either $\mathbb{R}$ (the set of real numbers), $\mathbb{Z}$ (the set of integers), $\mathbb{B} = \{\mathtt{true}, \mathtt{false}\}$ (the set of Boolean) or an enumerated set (e.g. $\{\mathtt{red}, \mathtt{green}, \mathtt{blue}\}$). The *attribute* $\mathtt{a}_i$ ($i \in \{1, 2, \ldots, m\}$) is the i^{th} projection of $\mathtt{Inst}_{\mathtt{pb}}$ on $\mathtt{V}_i$: $\mathtt{a}_i(x_1, \ldots, x_m) = x_i$.

To each $\mathtt{a}_i$ is associated a language of constraints $\mathcal{L}_{\mathtt{a}_i}$: $\mathtt{C} \in \mathcal{L}_{\mathtt{a}_i}$ is interpreted as a subset $\mathrm{Ext}(\mathtt{C})$ of $\mathtt{V}_i$. As an example, the constraint $[50, +\infty[$ stated on $\mathtt{V}_i = \mathbb{Z}$ represents the set of integers that are greater or equal to 50. $\mathcal{L}_{\mathtt{a}_i}$ is assumed (1) to contain the constraints $\top_i$ and $\bot_i$ such that $\mathrm{Ext}(\top_i) = \mathtt{V}_i$ and $\mathrm{Ext}(\bot_i) = \emptyset$, (2) to be closed under conjunction (if $\mathtt{C}, \mathtt{D} \in \mathcal{L}_{\mathtt{a}_i}$ there exists $\mathtt{E} \in \mathcal{L}_{\mathtt{a}_i}$ such that $\mathrm{Ext}(\mathtt{C}) \cap \mathrm{Ext}(\mathtt{D}) = \mathrm{Ext}(\mathtt{E})$). Besides, it is assumed that two distinct constraints cannot denote the same set of objects: if $\mathrm{Ext}(\mathtt{C}) = \mathrm{Ext}(\mathtt{D})$ then $\mathtt{C} = \mathtt{D}$.

A *problem descriptor* is a pair $\mathtt{d} = (\mathtt{a}_i, \mathtt{C})$ where $i \in \{1, \ldots, m\}$ and $\mathtt{C} \in \mathcal{L}_{\mathtt{a}_i}$. It is interpreted as the set of problem instances which value for the i^{th} attribute $\mathtt{a}_i$ is in the extension of $\mathtt{C}$: $\mathrm{Ext}((\mathtt{a}_i, \mathtt{C})) = \mathtt{a}_i^{-1}(\mathrm{Ext}(\mathtt{C})) = \{x \in \mathtt{Inst}_{\mathtt{pb}} \mid \mathtt{a}_i(x) \in \mathrm{Ext}(\mathtt{C})\}$.

Finally, a problem is an expression of the form $\mathtt{pb} = \mathtt{d}_1 \wedge \cdots \wedge \mathtt{d}_p$ where the $\mathtt{d}_i$ are problem descriptors. A problem is interpreted by an intersection: $\mathrm{Ext}(\mathtt{d}_1 \wedge \cdots \wedge \mathtt{d}_p) = \mathrm{Ext}(\mathtt{d}_1) \cap \ldots \cap \mathrm{Ext}(\mathtt{d}_p)$.

A problem $\mathtt{pb}$ is satisfiable if $\mathrm{Ext}(\mathtt{pb}) \neq \emptyset$. A problem $\mathtt{pb}$ is under normal form if $\mathtt{pb} = \mathtt{d}_1 \wedge \cdots \wedge \mathtt{d}_m$ where, for each $i \in \{1, \ldots, m\}$, $\mathtt{d}_i = (\mathtt{a}_i, \mathtt{C})$ for some $\mathtt{C} \in \mathcal{L}_{\mathtt{a}_i}$. It can be shown that a problem in normal form $\mathtt{pb}$ is satisfiable iff it contains no occurrence of $\bot_i$. Moreover, every satisfiable problem $\mathtt{pb}$ is equivalent to a unique problem $\mathtt{pb}'$ in normal form: $\mathrm{Ext}(\mathtt{pb}) = \mathrm{Ext}(\mathtt{pb}')$. If $\mathtt{pb} = (\mathtt{a}_1, \mathtt{C}_1) \wedge \cdots \wedge (\mathtt{a}_m, \mathtt{C}_m)$ is satisfiable and in normal form, then $\mathtt{a}_i(\mathrm{Ext}(\mathtt{pb})) = \mathrm{Ext}(\mathtt{C}_i)$, which justifies the notation $\mathtt{a}_i(\mathtt{pb}) = \mathtt{C}_i$. For the sake of simplicity, a conjunct of a problem of the form $(\mathtt{a}_i, \top_i)$ may be omitted in the notation: $\mathtt{pb} = (\mathtt{a}_1, \mathtt{C}_1) \wedge (\mathtt{a}_2, \top_2) = (\mathtt{a}_1, \mathtt{C}_1)$.

Solutions are defined analogously with a set of attributes $\mathtt{A}_j$ with $j \in \{1, \ldots, n\}$.

4.2. Representing Variations

In this section, problems and solutions are assumed to be satisfiable and in normal form. The problem variation language $\mathcal{L}_{\Delta\mathtt{pb}}$ is defined by four constructors. The constructors $\top_{\Delta\mathtt{pb}}$ and $\langle\mathtt{srce}, \mathtt{tgt}\rangle$ have already been defined in the general framework (cf. section 3.1). The third constructor is the conjunction: if $\mathtt{r}, \mathtt{s} \in \mathcal{L}_{\Delta\mathtt{pb}}$ then $\mathtt{r} \wedge \mathtt{s} \in \mathcal{L}_{\Delta\mathtt{pb}}$ and $(\mathtt{pb}_1, \mathtt{pb}_2) \in \mathrm{Ext}(\mathtt{r} \wedge \mathtt{s})$ iff $(\mathtt{pb}_1, \mathtt{pb}_2) \in \mathrm{Ext}(\mathtt{r})$ and $(\mathtt{pb}_1, \mathtt{pb}_2) \in \mathrm{Ext}(\mathtt{s})$. The last constructor is $\mathtt{a}_i^{\delta}$, where $\mathtt{a}_i$ is an attribute and δ represents a binary relation on $\mathcal{L}_{\mathtt{a}_i}$; δ is chosen in a language $\mathcal{L}_{\Delta\mathtt{a}_i}$. The semantics of $\mathtt{a}_i^{\delta}$ is as follows: two problems $\mathtt{srce}$ and $\mathtt{tgt}$ are related by $\mathtt{a}_i^{\delta}$ if $\mathtt{a}_i(\mathtt{srce})$ and $\mathtt{a}_i(\mathtt{tgt})$ are related by δ. More formally:

$$\mathrm{Ext}(\mathtt{a}_i^{\delta}) = \{(\mathtt{srce}, \mathtt{tgt}) \in \mathcal{L}_{\mathtt{pb}} \times \mathcal{L}_{\mathtt{pb}} \mid (\mathtt{a}_i(\mathtt{srce}), \mathtt{a}_i(\mathtt{tgt})) \in \mathrm{Ext}(\delta)\}$$

For example, let us consider the formalism $\mathcal{L}_{\mathtt{pb}}$ based on the attributes $\mathtt{a}_1 = \mathtt{gender}$, $\mathtt{a}_2 = \mathtt{age}$, and $\mathtt{a}_3 = \mathtt{s}$ ($\mathtt{V}_1 = \{\mathtt{female}, \mathtt{male}\}$, $\mathtt{V}_2 = \mathbb{Z}$, and $\mathtt{V}_3 = \mathbb{R}$). It is assumed that it is an attribute-*value* formalism, meaning that $\mathcal{L}_{\mathtt{a}_i}$ contains only $\top_i$, $\bot_i$, and the singletons $\{x\}$, for $x \in \mathtt{V}_i$. All the $\mathcal{L}_{\Delta\mathtt{a}_i}$ share the relations $=$ and $\neq$. $\mathcal{L}_{\Delta\mathtt{age}}$ and $\mathcal{L}_{\Delta\mathtt{s}}$ share the relations $\delta \in \{<, \leq, \geq, >\}$ on singletons defined by $\{x\}\ \delta\ \{y\}$ if $x\ \delta\ y$ (e.g., $\{3\} < \{4\}$ since $3 < 4$). Thus, if

$$\mathtt{srce} = (\mathtt{gender}, \{\mathtt{female}\}) \wedge (\mathtt{age}, \{65\}) \wedge (\mathtt{s}, \{3\})$$
$$\mathtt{tgt} = (\mathtt{gender}, \{\mathtt{female}\}) \wedge (\mathtt{age}, \{70\}) \wedge (\mathtt{s}, \{2\})$$

then

$$\langle\mathtt{srce}, \mathtt{tgt}\rangle \vDash \mathtt{gender}^{=} \wedge \mathtt{age}^{<} \wedge \mathtt{age}^{\leq} \wedge \mathtt{age}^{\neq} \wedge \mathtt{s}^{>} \wedge \mathtt{s}^{\geq} \wedge \mathtt{s}^{\neq}$$

$\mathcal{L}_{\Delta\mathtt{sol}}$ is defined thanks to four similar constructors.

The adaptation rule language is $\mathcal{L}_{\Delta\mathtt{pb}} \times \mathcal{L}_{\Delta\mathtt{sol}}$. If $\mathtt{AR} = (\mathtt{r}, \mathtt{R})$ and $\mathtt{AR}' = (\mathtt{r}', \mathtt{R}')$ are two adaptation rules then $\mathtt{AR} \wedge \mathtt{AR}'$ denotes the adaptation rule $(\mathtt{r} \wedge \mathtt{r}', \mathtt{R} \wedge \mathtt{R}')$, which is consistent with the semantics of adaptation rules given by (1).

Examples of δ. The definition of $\mathcal{L}_{\Delta\mathtt{pb}}$ has been reduced above to the definition of a $\mathcal{L}_{\Delta\mathtt{a}_i}$, for each attribute $\mathtt{a}_i$. Although the definition of $\mathcal{L}_{\Delta\mathtt{a}_i}$ is a knowledge acquisition issue, some examples of relations $\delta \in \mathcal{L}_{\Delta\mathtt{a}_i}$ that may be useful are presented here.

When a set $\mathtt{V}_i$ is associated to an algebraic structure, the latter may be reused on singletons. For example, since $\leq$ is a relation on $\mathbb{Z}$, it can be used as a relation between singletons of integers, as already mentioned above ($\{3\} < \{4\}$). Another example is related to the law $+$ on $\mathbb{Z}$, that is used to define the binary relation $\delta = \mathtt{add}(\alpha)$ on $\mathbb{Z}$ (for each $\alpha \in \mathbb{Z}$): $x\ \mathtt{add}(\alpha)\ y$ if $x + \alpha = y$.

Since δ relates two (representations of) sets, the classical binary relations between sets ($\subsetneq$,$\subseteq$,$=$,$\supseteq$,$\supsetneq$) can be suggested for elements of $\mathcal{L}_{\Delta\mathtt{a}_i}$. Additionally, let $\mathtt{C} \in \mathcal{L}_{\mathtt{a}_i}$ and let $(\subseteq\mathtt{C}\supseteq)$ be defined, for $\mathtt{C}_1, \mathtt{C}_2 \in \mathcal{L}_{\mathtt{a}_i}$ by $\mathtt{C}_1\ (\subseteq\mathtt{C}\supseteq)\ \mathtt{C}_2$ if $\mathrm{Ext}(\mathtt{C}_1) \subseteq \mathrm{Ext}(\mathtt{C}) \supseteq \mathrm{Ext}(\mathtt{C}_2)$: $\mathtt{C}_1$ and $\mathtt{C}_2$ share the constraint $\mathtt{C}$. Another relation is $\ominus$ (resp., $\oplus$) defined by $\mathtt{C} \ominus \mathtt{D}$ if $\mathtt{C} \neq \top_i$ and $\mathtt{D} = \top_i$ (resp., $\mathtt{C} = \top_i$ and $\mathtt{D} \neq \top_i$). Note that $\mathtt{a}_i^{\ominus} \vDash \mathtt{a}_i^{\subsetneq}$ and $\mathtt{a}_i^{\oplus} \vDash \mathtt{a}_i^{\supsetneq}$. These relations can be applied in particular when $\mathcal{L}_{\mathtt{a}_i} = 2^{\mathtt{V}_i}$, where $\mathtt{V}_i$ is an enumerated set. They can also be applied when $\mathcal{L}_{\Delta\mathtt{a}_i}$ is a finite set of atomic constraints organized in a hierarchy of root $\top_i$. Finally, they can be applied on intervals on, e.g., $\mathbb{Z}$ or $\mathbb{R}$.

Other relations between intervals may be defined thanks to the reuse of Allen relations on temporal intervals [1]. For example, if $\mathtt{C}_1 = [x_1, y_1]$ and $\mathtt{C}_2 = [x_2, y_2]$ are two closed intervals on $\mathbb{R}$, $\mathtt{C}_1\ b\ \mathtt{C}_2$ if $y_1 < x_2$, $a = b^{-1}$, $\mathtt{C}_1\ m\ \mathtt{C}_2$ if $y_1 = x_2$, etc. (b, a, and m stand for *before*, *after*, and *meets*). These *qualitative* relations may be completed with *quantitative* relations such as $\mathtt{addToBound}(\alpha, \beta)$ defined by $\mathtt{C}_1\ \mathtt{addToBound}(\alpha, \beta)\ \mathtt{C}_2$ if $x_1 + \alpha = x_2$ and $y_1 + \beta = y_2$.

5. Experiments

Some experiments have been carried out in the oncology domain in order to evaluate the benefit of choosing an appropriate representation of the variations between cases.

5.1. *Learning algorithm:* Charm

Charm [11] is a data-mining algorithm that efficiently performs the extraction of *frequent closed itemsets* (FCIs). Given a finite set $\mathcal{P}$ (the set of *properties* or *items*), an itemset I is a subset of $\mathcal{P}$: $I \in 2^{\mathcal{P}}$. The input of Charm is a set of itemsets called the *transactions*. The support of an itemset I is the proportion $\texttt{support}(I)$ of the transactions T that contain I ($T \supseteq I$). An itemset I is *frequent* with respect to the threshold $\sigma_s \in [0;1]$ if $\texttt{support}(I) \geq \sigma_s$. I is closed if adding to it any property alters its support: $\texttt{support}(I) > \texttt{support}(I \cup \{x\})$ for any $x \in \mathcal{P}$ such that $x \notin I$.

A benefit of using Charm lies in its efficiency with large sets of transactions. A difficulty is that, since it operates on data (and not on pieces of knowledge), a translation of the training set $\texttt{TS}$ into a set of transactions is required. Since the training set is constituted by specific adaptation rules $\texttt{AR}_{k\ell}$ that have to be apprehended modulo the deduction relation $\vDash$, the idea is to translate $\texttt{AR}_{k\ell}$ into the transaction $\texttt{DC}(\texttt{AR}_{k\ell})$. To be consistent with this definition, the set $\mathcal{P}$ is set to $\bigcup_{\texttt{AR}_{k\ell} \in \texttt{TS}} \texttt{DC}(\texttt{AR}_{k\ell})$. Now, let $\texttt{AR}_1$, $\texttt{AR}_2$, and $\texttt{AR}_3$ be three adaptation rules and $I = \{\texttt{AR}_1, \texttt{AR}_2, \texttt{AR}_3\}$. If there exist exactly n $\texttt{AR}_{k\ell} \in \texttt{TS}$ such that $\texttt{AR}_{k\ell} \vDash \texttt{AR}_1 \wedge \texttt{AR}_2 \wedge \texttt{AR}_3$ then the itemset I is frequent iff $n \geq \sigma_s \times \texttt{card}\ (\texttt{TS})$. If, for the *same* n $\texttt{AR}_{k\ell} \in \texttt{TS}$, $\texttt{AR}_{k\ell} \vDash \texttt{AR}$ with $\texttt{AR} \notin I$, then I is not closed. In other words, if I is not closed, this means that I is an over-generalization that can be specialized in the adaptation rule language without loss of the coverage of the rule in the training set.

A practical problem is raised when some $\texttt{DC}(\texttt{AR}_{k\ell})$ are not finite: this leads to an infinite $\mathcal{P}$ that Charm cannot manage. The idea is then to restrict $\mathcal{P}$ to a finite set and each deductive closure to $\texttt{DC}(\texttt{AR}_{k\ell}) \cap \mathcal{P}$.

5.2. *Experimental Setup*

The application domain of the experiments is breast cancer treatment: a problem describes a class of patients ill with breast cancer and a solution is a treatment. A problem is represented in an attribute-constraint formalism, with 22 attributes with various constraint languages $\mathcal{L}_{\mathsf{a}_i}$: $2^{\mathbb{B}}$, singletons, numerical intervals, and atomic constraint hierarchies. A solution is also represented in an attribute-constraint formalism with 65 attributes. The case base contains 44 cases.

Variation languages are defined in the formalism of section 4.2, with the relations δ given as examples. The training set $\texttt{TS}$ is translated into $44 \times (44 - 1) = 1892$ transactions with $\texttt{card}\ \mathcal{P} = 300$.

5.3. *Results*

From this set of transactions, Charm extracted 342,994 itemsets in about 1 minute on a current PC. About 84% of this result set corresponds to itemsets with a support lower than 3% (i.e., generalizing less than 56 transactions).

Examples of learned rules. The extracted adaptation rules are organized in a hierarchy for $\vDash$. The expert that has to validate these rules navigates in this hierarchy. For example, the expert has found the following adaptation rule:

$$\begin{aligned}\mathtt{AR} = (&\mathtt{age}^{\neq} \wedge \mathtt{ctxt},\\ &\mathtt{nb\text{-}of\text{-}FEC\text{-}cycles}^{\neq} \wedge \mathtt{dose\text{-}of\text{-}FEC}^{\neq} \wedge \mathtt{Ctxt})\end{aligned}$$

where $\mathtt{ctxt}$ denotes some common context the two problems $\mathtt{srce}$ and $\mathtt{tgt}$ have to share. More precisely, $\mathtt{ctxt}$ is a conjunction of $\mathtt{a}_i^{=}$ (and, thus, $\mathtt{ctxt}^{-1} \equiv \mathtt{ctxt}$), which involves that $\mathtt{a}_i(\mathtt{tgt}) = \mathtt{a}_i(\mathtt{srce})$ is a condition of the adaptation rule. Similarly, $\mathtt{Ctxt}$ denotes some common context for the two solutions $\mathtt{Sol}(\mathtt{srce})$ and $\mathtt{Sol}(\mathtt{tgt})$ and is a conjunction of $\mathtt{A}_j^{=}$, which involves that $\mathtt{A}_j(\mathtt{Sol}(\mathtt{tgt})) = \mathtt{A}_j(\mathtt{Sol}(\mathtt{srce}))$. FEC is the name of a drug for chemotherapy that is given in several cycles (attribute nb-of-FEC-cycles with values in $\mathbb{Z}$) with a fixed dose in each cycle (attribute dose-of-FEC with values in $\mathbb{R}$).

This rule expresses that the choice of the FEC treatment depends on the age of the patient, but does not make this dependency explicit. A navigation down the hierarchy gives the following pair of rules:

$$\begin{aligned}\mathtt{AR}_1 = (&\mathtt{age}^{b} \wedge \mathtt{ctxt}',\\ &\mathtt{nb\text{-}of\text{-}FEC\text{-}cycles}^{>} \wedge \mathtt{dose\text{-}of\text{-}FEC}^{>} \wedge \mathtt{Ctxt}')\\ \mathtt{AR}_2 = (&\mathtt{age}^{a} \wedge \mathtt{ctxt}',\\ &\mathtt{nb\text{-}of\text{-}FEC\text{-}cycles}^{<} \wedge \mathtt{dose\text{-}of\text{-}FEC}^{<} \wedge \mathtt{Ctxt}')\end{aligned}$$

where $\mathtt{AR}_2 \equiv \mathtt{AR}_1{}^{-1}$, $\mathtt{ctxt}' \vDash \mathtt{ctxt}$ and $\mathtt{Ctxt}' \vDash \mathtt{Ctxt}$. Each of these rules states that the dependency pointed out thanks to AR is decreasing: when the age increases ($\mathtt{age}(\mathtt{srce})\ b\ \mathtt{age}(\mathtt{tgt})$) the number of cycles and the dose per cycle decrease.

The expert explains this rule by (1) the fact that the growth rate of the tumour is higher for younger patients and thus requires higher doses of chemotherapy and (2) the necessity to make a compromise taking into account the life expectation (that decreases with the age) and the life quality (that decreases with the dose of FEC).

These rules could be learned only because an expressive language was chosen to represent the variations between cases.

Navigating in the results. Using an expressive language to represent the variations between cases also allowed to structure the result set and to provide the expert with efficient means of navigation in it.

Among extracted rules, only a few like AR were extracted with a high support value. These rules constitute a good starting point because their interpretation is quite straightforward, which makes the expert's work much easier during the validation phase. However, they often appear to be too general and their contexts of validity need to be refined. Most of the valid rules that were found had a fairly low support. Some rules are even valid only locally, that is in a very specific context.

To discover these rules, the generality relation $\vDash$ on adaptation rules was used to structure the result set and provide the expert with efficient means of navigation in it. For example, a filter on the result set has been implemented that enables the expert to have access to all rules that are more specific (resp., more general) than a given rule. Using this filter, the expert has for instance been able to visualize the set of all rules that are more specific than $\mathtt{AR}$, among which are the rules $\mathtt{AR}_1$ and $\mathtt{AR}_2$ (since $\mathtt{AR}_1 \vDash \mathtt{AR}$ and $\mathtt{AR}_2 \vDash \mathtt{AR}$).

Towards a methodology for result exploration. A methodology for the exploration of candidate adaptation rules has emerged from these experiments. The exploration starts with a phase of elaboration in which a search context $\mathtt{ctxt}$ is set up together with the expert. This search context is used to restrict the set of rules to search into by considering only the rules that apply to a particular medical situation. $\mathtt{ctxt}$ takes the form of a conjunction of $\mathtt{a}_i^=$ with which the result set is filtered to retain only the rules $\mathtt{AR}$ such that $\mathtt{AR} \vDash \mathtt{ctxt}$. The expert is then provided with means of navigation in the remaining rules and may choose to restrict his/her search to the set of rules that are more specific than a particular rule.

6. Discussion and Related Work

Discussion. The learning process presented above is sensitive to the choice of the variation languages $\mathcal{L}_{\Delta\mathtt{pb}}$ and $\mathcal{L}_{\Delta\mathtt{sol}}$. This choice should be made according to a bias/variance compromise [10]. If the languages are too rich, it may prevent from learning. For example, if the language of adaptation rules $\mathcal{L}_{\Delta\mathtt{pb}} \times \mathcal{L}_{\Delta\mathtt{sol}}$ is closed under disjunction, then the process learns "by heart" the adaptation rule $\bigvee\limits_{\mathtt{AR}_{k\ell} \in \mathtt{TS}} \mathtt{AR}_{k\ell}$. If this language is too poor for expressing relevant adaptation rules, these latter cannot be learned.

In the context of a given application domain, what are the relevant languages $\mathcal{L}_{\Delta\mathtt{pb}}$ and $\mathcal{L}_{\Delta\mathtt{sol}}$? Consider, in the breast cancer treatment domain, two problems $\mathtt{srce}$ and $\mathtt{tgt}$ such that $\mathtt{age}(\mathtt{srce}) = \{50\}$ and $\mathtt{age}(\mathtt{tgt}) = \{70\}$. Both relations $\mathtt{r}_1 = \mathtt{age}^{\delta_1}$ and $\mathtt{r}_2 = \mathtt{age}^{\delta_2}$, with $\delta_1 = \mathtt{add}(20)$ and $\delta_2 = \mathtt{multiplyBy}(1.4)$, relate $\mathtt{srce}$ to $\mathtt{tgt}$, but only $\mathtt{r}_1$ is relevant for the domain experts. Indeed, the comparison of patient ages is sometimes based on differences, never (or rarely) on ratios. Then, choosing $\mathtt{r}_1 \in \mathcal{L}_{\Delta\mathtt{pb}}$ (and, more generally, $\mathtt{add}(\alpha) \in \mathcal{L}_{\Delta\mathtt{age}}$ for $\alpha \in \mathbb{Z}$) and $\mathtt{r}_2 \notin \mathcal{L}_{\Delta\mathtt{pb}}$, can be justified by the assumption that the adaptation rules must be expressed with a language compatible with the way the expert expresses comparisons between cases. Therefore, the choice of a language bias is a knowledge acquisition process that is similar to the process of acquiring the vocabulary for representing cases: the latter consists in pointing out the entities for representing cases whereas the former consists in pointing out the entities for representing case variations.

Related Work. The approach of adaptation learning from the case base presented in this paper is inspired from the seminal work of Kathleen Hanney and Mark T. Keane [5,6], that presents some general principles and tests them successfully in two domains. The cases of these domains are expressed in an attribute-value formalism. More precisely, with the notations defined above, for each i, $\mathtt{V}_i = \mathbb{Z}$ or $\mathtt{V}_i$ is a finite (and rather small) interval of $\mathbb{Z}$, and $\mathcal{L}_{\mathtt{a}_i}$ contains only singletons. The variations are expressed by differences, i.e., with our notations, they are conjunctions of $\mathtt{a}_i^{\mathtt{add}(\alpha)}$ ($\alpha \in \mathbb{Z}$). Therefore, the current paper may be seen as a formalized generalization of [6]. A difference in methodology is the following: in [6], some adaptation rules are generated that are rather specific, and then, they are generalized using some of the R. S. Michalski's generalization rules [8]. In our approach, specific and general adaptation rules are generated at the same time, which enables to organize all these rules in a hierarchy for $\vDash$, that makes navigation among them easier.

Another method for learning adaptation knowledge from the case base is presented in [2], in which the use of the extracted knowledge significantly improved the CBR process. In this work, supervised leaning methods such as decision trees are applied to learn predictive models from a set of adaptation cases. These models are later used to reduce the set of adaptation cases to be used in a case-based adaptation process. There are two main differences between [2] and the work presented in this paper. First, the goal of [2] is to find adaptation knowledge in the form of adaptation cases, whereas we search for adaptation rules. Second, in [2] —and also in [6]— only pairs of *similar* source cases are considered for the learning process, whereas we favour the selection of every distinct ordered pairs of cases. This is motivated by the will of not adding another bias to the learning process.

Finally, in [3], the authors present a knowledge discovery approach for adaptation knowledge acquisition. They use a simple description logic for representing cases, that can be likened to the attribute-constraint formalism presented above. The main difference with the work presented in this paper is the richness of the variation language —and, consequently, of the adaptation rule language— we use. Indeed, with our notations, the comparisons of two cases in [3] are only based on $\mathtt{a}_i^{=}$, $\mathtt{a}_i^{\ominus}$, and $\mathtt{a}_i^{\oplus}$ (with $\mathtt{a}_i$: a problem or solution attribute).

7. Conclusion and Future Work

This paper presents a formalization of the task of adaptation rule learning from variations in the case base and shows, through experiments in the oncology domain, the benefit of an appropriate representation of variations for the purpose of this learning process. Such a representation consists in a set of binary relations between pairs of problems and pairs of solutions. When an attribute-constraint formalism (e.g., an attribute-value formalism) is used to represent cases, these relations can be based on binary relations δ between constraints $\mathtt{C}_1$ and $\mathtt{C}_2$ associated to the same attribute $\mathtt{a}_i$. Examples of such relations δ are presented.

An ongoing work aims at reducing the number of candidate adaptation rules to be examined by the experts, while keeping the same adaptation knowledge. It is based on the notion of adaptation rule composition: the composition of $(\mathtt{r}_1, \mathtt{R}_1)$ and $(\mathtt{r}_2, \mathtt{R}_2)$ is the rule $(\mathtt{r}_2 \circ \mathtt{r}_1, \mathtt{R}_2 \circ \mathtt{R}_1)$, provided that $\mathcal{L}_{\Delta\mathtt{pb}}$ and $\mathcal{L}_{\Delta\mathtt{sol}}$ are closed under binary relation composition $\circ$. If $\mathtt{S}_{\mathtt{CAR}}$ is the set of candidate adaptation rules that have been learned, a generative family of $\mathtt{S}_{\mathtt{CAR}}$ is a set $G \subseteq \mathtt{S}_{\mathtt{CAR}}$ such that its closure under adaptation rule composition contains $\mathtt{S}_{\mathtt{CAR}}$. The design of algorithms giving G with high rates $\frac{\mathtt{card}\ \mathtt{S}_{\mathtt{CAR}}}{\mathtt{card}\ G}$ has begun. The first algorithms implemented give a rate close to 2 (i.e., the expert's work is divided by 2).

Assessing the quality of the learned adaptation rules may be achieved by defining an objective measure of the quality of a rule, as it is usually done for association rules [4]. A future work aims at defining such measures, and studying whether they are useful to identify the most relevant adaptation rules in the result set. For this purpose, some of the measures of association rules, such as confidence or interest, may be adapted to the context of adaptation rules.

References

[1] J. F. Allen, Maintaining knowledge about temporal intervals, *Communications of the ACM* **26(11)** (1983), 832–843.

[2] S. Craw, N. Wiratung, and R. Rowe, Learning adaptation knowledge to improve case-based reasoning, *Artificial Intelligence*, **170(16-17)** (2006), 1175–1192.

[3] M. d'Aquin, F. Badra, S. Lafrogne, J. Lieber, A. Napoli, and L. Szathmary, Case base mining for adaptation knowledge acquisition, in *Proceedings of the International Conference on Artificial Intelligence, IJCAI'07*, 750–756, 2007.

[4] F. Guillet and H. J. Hamilton, *Quality Measures in Data Mining (Studies in Computational Intelligence)*, Springer-Verlag New York, Inc., 2007.

[5] K. Hanney, *Learning Adaptation Rules from Cases*, Master's thesis, Trinity College, Dublin, 1997.

[6] K. Hanney and M. T. Keane, Learning Adaptation Rules from Cases, in *Proceedings of the 3rd European Workshop on Case-Based Reasoning, EWCBR-96*, eds., I. Smith and B. Falting, volume 1168 of LNAI, Springer, 1996.

[7] J. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann Publishers, Inc., 1993.

[8] R. S. Michalski, A Theory and Methodology of Inductive Learning, in *Machine Learning*, 83–134, Springer-Verlag, 1983.

[9] C. K. Riesbeck and R. C. Schank, *Inside Case-Based Reasoning*, Lawrence Erbaum Associates, Inc., Hillsdale, New Jersey, 1989.

[10] D. H. Wolpert, On Bias Plus Variance, *Neural Computation*, **9(6)** (1997), 1211–1243.

[11] M. J. Zaki and C.-J. Hsiao, CHARM: An efficient algorithm for closed itemset mining, in *Proceedings of the Second SIAM International Conference on Data Mining, Arlington, VA, USA, April 11-13, 2002*, eds., Robert L. Grossman, Jiawei Han, Vipin Kumar, Heikki Mannila, and Rajeev Motwani, SIAM, 2002.