Local Reducts and Jumping Emerging Patterns in Relational Databases*

Pawel Terlecki and Krzysztof Walczak

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
P.Terlecki, K.Walczak@ii.pw.edu.pl

Abstract. This paper refers to the notion of minimal pattern in relational databases. We study the analogy between two concepts: a local reduct, from the rough set theory, and a jumping emerging pattern, originally defined for transactional data. Their equivalence within a positive region and similarities between eager and lazy classification methods based on both ideas are demonstrated. Since pattern discovery approaches vary significantly, efficiency tests have been performed in order to decide, which solution provides a better tool for the analysis of real relational datasets.

1 Introduction

Many definitions of knowledge discovery emphasize the importance of patterns in information modeling. There are at least three important reason for this. First of all, patterns are a useful tool in many practical problems, mainly in classification. Secondly, they can be easily understood by the human mind. Unlike neural networks, support vector machines or Bayesian classifiers, the most expressive patterns do not need any additional visualization to be comprehended and evaluated. Last but not least, the simple structure of a pattern and the intensive development of concise representations make them a convenient and powerful tool in knowledge processing and storing.

Many experiments demonstrated the accuracy of rule-based classifiers [1]. However, there are no criteria applicable to all types of data and different sorts of patterns are still being proposed to produce better rules. Notwithstanding this variety, we can figure out some common features, like their highly discriminative power, not overfitting to training data or avoiding exponential result sets.

In this paper, we focus on patterns in relational databases. One of the most elegant descriptions for this kind of data is provided by the rough set theory. Basic concepts triggered intensive research which has brought many methods suitable for practical application. The most accurate classifiers are based on the notion of a local reduct, i.e. a minimal set of attributes capable of distinguishing one particular object from objects belonging to other classes as well as the

* The research has been partially supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

S. Greco et al. (Eds.): RSCTC 2006, LNAI 4259, pp. 358–367, 2006.
© Springer-Verlag Berlin Heidelberg 2006

total set of attributes. This approach allows to induce minimal patterns and corresponding rules that describe a class in a very general manner.

In contrast to these classic solutions, we observe a fast development of methods for transaction databases. The most popular solutions make use of class association rules (CAR) and emerging patterns. Such classifiers as JEP-C, CAEP, DeEP or EJEP-C have already proved their high accuracy in many experiments [1]. Our study focuses on jumping emerging patterns (JEPs), the idea very similar to minimal patterns based on local reducts. A JEP is a pattern that is present in one class and absent in others. In particular, the minimal ones give a possibly general and concise description of each class in contrast to the rest.

As it was mentioned above, a JEP is originally defined by means of the traditional formal apparatus of transaction databases. Nevertheless, it is often used to deal with relational data transformed to a transactional form [2]. Now, the question emerges what the differences are between classification algorithms associated with the concept of a local reduct and a JEP. Another question is which of these approaches allows to discover the set of minimal patterns more efficiently, when relational data is concerned. Our intuition is that algorithms which operate on this kind of data can take advantage of information held in attributes in order to differentiate objects, whereas methods using the transactional approach fail to account for the actual relation between items associated with the same attribute. In addition, the space of attribute sets is often much smaller than the respective space of itemsets, which also depends on attribute value domains. For these reasons, we expect that, at least for large datasets, the efficiency of methods from both streams will be significantly different.

The text is organized as follows. Section 2 provides a formal background for the rough set theory, EPs and a relational-to-transactional transformation. In Sect. 3, we prove the similarity between minimal patterns obtained from local reducts and JEPs. Then, in Sect. 4 basic classification methods from both streams are compared. We discuss popular eager and lazy methods, taking into account differences in the way of selecting minimal patterns and aggregating them in order to obtain a decision. Section 5 explains the main issues of the two methods of minimal pattern discovery: the rough set approach and JEP-Producer. Implementation remarks are discussed in Sect. 6. Our testing procedure and results are presented in Sect. 7. The paper is summarized in Sect. 8.

2 Preliminaries

2.1 Elements of Rough Set Theory

Let a decision table be a triple $(\mathcal{U}, \mathcal{C}, d)$, where $\mathcal{U}$(universum) is a non-empty, finite set of objects, $\mathcal{C}$ is a non-empty finite set of condition attributes and d is a decision attribute. A set of all attributes is denoted by $\mathcal{A} = \mathcal{C} \cup \{d\}$. The domain of an attribute $a \in \mathcal{A}$ is denoted by V_a and its value for an object $u \in \mathcal{U}$ is denoted by a(u). In particular, $V_d = \{k_1, .., k_{|V_d|}\}$ and the decision attribute induces a partition of $\mathcal{U}$ into decision classes $\{U_k\}_{k \in V_d}$. Hereinafter, we use the term *attribute* to denote a condition attribute.

Consider $B \subseteq \mathcal{A}$. An indiscernibility relation $IND(B)$ is defined as follows:

$$IND(B) = \{(u, v) \in \mathcal{U} \times \mathcal{U} : \forall_{a \in B}\ a(u) = a(v)\}$$

Since $IND(B)$ is an equivalence relation, it induces a partition of $\mathcal{U}$ denoted by $\mathcal{U}/IND(B)$. Let $B(u)$ be a block of the partition containing $u \in \mathcal{U}$. A B-lower approximation of a set $X \subseteq \mathcal{U}$ is defined as follows: $B_*(X) = \{u \in \mathcal{U} \mid B(u) \subseteq X\}$, and a B-positive region with respect to a decision attribute d is defined by:

$$POS(B, d) = \bigcup_{X \in \mathcal{U}/IND(\{d\})} B_*(X)$$

We say that a decision table is consistent or deterministic if $POS(\mathcal{C}, d) = \mathcal{U}$. Otherwise, we call it inconsistent or non-deterministic. A local reduct for an object $u \in \mathcal{U}$ (a reduct relative to an object and a decision) is a minimal attribute set $B \subseteq \mathcal{C}$ such that $\forall_{k \in V_d}(\mathcal{C}(u) \cap U_k = \emptyset \implies B(u) \cap U_k = \emptyset)$. It means that the object u can be differentiated by means of B from all the objects from other classes as accurately as by the complete available description $\mathcal{C}$. The set of all local reducts for an object u is denoted by $REDLOC(u, d)$.

Lemma 1 ([3]). *$B \in REDLOC(u, d)$ for $u \in POS(\mathcal{C}, d) \iff B$ is a minimal set such that $B(u) \subseteq U_{d(u)}$.*

2.2 Emerging Patterns

Let a decision transaction database be a tuple $(\mathcal{D}, \mathcal{N}, \mathcal{I}, \mathcal{Z})$, where $\mathcal{D} \subseteq \{(n, t) \in \mathcal{N} \times 2^{\mathcal{I}} : \forall_{(n', t') \in \mathcal{N} \times 2^{\mathcal{I}}} n = n' \implies t = t'\}$ is a set of transactions (database), $\mathcal{N}$ is a non-empty set of transaction identifiers, $\mathcal{I}$ is a non-empty set of items and $\mathcal{Z}$ is a function $\mathcal{Z} : \mathcal{D} \mapsto V_Z$, where V_Z is the set of decision class labels. The function $\mathcal{Z}$ splits the database $\mathcal{D}$ into decision classes $D_k = \mathcal{Z}^{-1}(k)$, for $k \in V_Z$. In addition, for $D \subseteq \mathcal{D}$, we define a complement database $D' = \mathcal{D} - D$. An itemset $X \in 2^{\mathcal{I}}$ is a set of items and its support in a database $D \subseteq \mathcal{D}$ is defined as $supp_D(X) = \frac{|(n,t) \in D : X \subseteq t|}{|D|}$. Given two databases $D_1, D_2 \subseteq \mathcal{D}$, we define a jumping emerging pattern (JEP) from D_1 to D_2 as an itemset X for which $supp_{D_1}(X) = 0$ and $supp_{D_2}(X) \neq 0$. A set of all JEPs from D_1 to D_2 is called a JEP space and denoted by $JEP(D_1, D_2)$.

2.3 Convexity of JEP Space

One of the most useful features of jumping emerging patterns is the possibility to store a JEP space in a concise manner.

Consider a set S. A collection $F \subseteq 2^{\mathcal{S}}$ is a convex space iff $\forall_{X,Z \in F} \forall_{Y \in 2^{\mathcal{S}}} X \subseteq Y \subseteq Z \Rightarrow Y \in F$. A border is an ordered pair $< \mathcal{L}, \mathcal{R} >$ such that $\mathcal{L}, \mathcal{R} \subseteq P(S)$ are antichains and $\forall_{X \in \mathcal{L}} \exists_{Z \in \mathcal{R}} X \subseteq Z$. $\mathcal{L}$ and $\mathcal{R}$ are called a left and a right bound, respectively. A border $< \mathcal{L}, \mathcal{R} >$ represents a set interval $[\mathcal{L}, \mathcal{R}] = \{Y \in P(S) : \exists_{X \in \mathcal{L}} \exists_{Z \in \mathcal{R}} X \subseteq Y \subseteq Z\}$. The left and right bounds consist, respectively, of minimal elements and maximal elements of a set, assuming inclusion relation. It can be demonstrated [2] that every convex space has a unique border.

Consider a decision transaction database $(\mathcal{D}, \mathcal{N}, \mathcal{I}, \mathcal{Z})$ and two databases $D_1, D_2 \subseteq \mathcal{D}$. According to [2] a collection $JEP(D_1, D_2)$ is a convex space. Thus, for $k \in V_Z$, we use a border $< \mathcal{L}_k, \mathcal{R}_k >$ to represent a JEP space $JEP(D'_k, D_k)$.

Lemma 2 ([2]). $\forall_{J \in 2^{\mathcal{I}}} J$ *is minimal in* $JEP(D'_k, D_k) \iff J \in \mathcal{L}_k$.

2.4 Relational to Transactional Transformation

One can analyze relational data by means of methods formulated for transaction databases. In our study, we consider a decision transaction database build for a given decision table. For brevity, we use the following notations introduced in [4]: $patt(u, B) = \{(a, a(u))\}_{a \in B}$, where $u \in \mathcal{U}$ and $B \subseteq \mathcal{C}$, and $attr(X) = \{a \in \mathcal{C} : (a, v) \in X \wedge v \in V_a\}$, for an itemset $X \subset \{(a, v)\}_{a \in \mathcal{C}, v \in V_a}$. Without loss of generality, we assume that a universum can be linearly ordered $\mathcal{U} = \{u_1, .., u_{|\mathcal{U}|}\}$.

Definition 1. *A decision transaction database for a decision table* $(\mathcal{U}, \mathcal{C}, d)$ *is a decision transaction database* $(\mathcal{D}, \mathcal{N}, \mathcal{I}, \mathcal{Z})$*, such that*

- $\mathcal{D} = \{\varphi(u)\}_{u \in \mathcal{U}}$*, where* $\varphi : \mathcal{U} \mapsto \mathcal{D}$*,* $\forall_{i \in \{1..|\mathcal{U}|\}} \varphi(u_i) = (i, patt(u_i, \mathcal{C}))$
- $\mathcal{N} = \mathbb{N}$ *(positive integers)*
- $\mathcal{I} = \{(a, v)\}_{a \in \mathcal{C}, v \in V_a}$
- $V_{\mathcal{Z}} = V_d$ *and* $\forall_{u \in U} \mathcal{Z}(\varphi(u)) = d(u)$

Notice that φ is a bijection, so it is possible to transform the result obtained by some methods for transaction data back to relational form.

3 Relations Between Concepts

Hereinafter, we consider a decision table $\mathcal{DT} = (\mathcal{U}, \mathcal{C}, d)$ and a decision transaction database $\mathcal{RDDT} = (\mathcal{D}, \mathcal{N}, \mathcal{I}, \mathcal{Z})$ for $\mathcal{DT}$. For $u \in \mathcal{U}$, a set of all local reducts for the object u is represented by $REDLOC(u, d)$ and for $k \in V_d$ a JEP space $JEP(D'_k, D_k)$ is represented by the border $< \mathcal{L}_k, \mathcal{R}_k >$.

Rough set reducts and emerging patterns are strongly related concepts. Our previous paper [4] demonstrates the relations between global reducts and JEPs. According to that work, every global reduct P generates with object $u \in \mathcal{U}$ a pattern $patt(u, P)$ that belongs to $JEP(D'_{d(u)}, D_{d(u)})$. The following theorem considers a similar relation for local reducts. It says that every local reduct generates with object $u \in POS(\mathcal{C}, d)$ a jumping emerging pattern that is minimal, i.e. it belongs to $\mathcal{L}_{d(u)}$, the left bound of the border of a space $JEP(D'_{d(u)}, D_{d(u)})$.

Notice that for a consistent decision table this relation holds for each $u \in \mathcal{U}$. Thus, we can use algorithms originating in either rough set theory or emerging patterns approach to compute the set of minimal patterns.

Theorem 1. *Let* $\mathcal{DT} = (\mathcal{U}, \mathcal{C}, d)$ *be a decision table and* $\mathcal{RDDT} = (\mathcal{D}, \mathcal{N}, \mathcal{I}, \mathcal{Z})$ *a decision transaction database for* $\mathcal{DT}$*.*

$\forall_{u \in POS(\mathcal{C}, d)} \forall_{P \subseteq \mathcal{C}} P \in REDLOC(u, d) \iff patt(u, P) \in \mathcal{L}_{d(u)}$.

Proof. Let $P, B \in \mathcal{C}$, $u \in POS(\mathcal{C}, d)$ and $k = d(u)$.

Consider first $B(u) \subseteq U_k \iff patt(u, B) \in JEP(D'_k, D_k)$ (1). We have $B(u) \subseteq U_k \iff u \in B_*(U_k) \iff u \in POS(B, d) \cap U_k$. But, according to Theorem 1 from [4], we have: $u \in POS(B, d) \cap U_k \iff patt(u, B) \in \{J \in JEP(D'_k, D_k) : attr(J) = B\} \iff patt(u, B) \in JEP(D'_k, D_k)$.

Consider $P \in REDLOC(u, d) \Longrightarrow patt(u, P) \in \mathcal{L}_k$. Let $P \in REDLOC(u, d)$. According to Lemma 1, we have: $P \in REDLOC(u, d) \iff P$ is minimal in $\{B \subseteq \mathcal{C} : B(u) \subseteq U_k\}$. Consider $R \subset P$. It means that $R(u) \not\subseteq U_k$, and, according to (1), we obtain $patt(u, R) \notin JEP(D'_k, D_k)$. Summing up, according to (1) we have $patt(u, P) \in JEP(D'_k, D_k)$ and for any $J \subset patt(u, P)$ we have $J \notin JEP(D'_k, D_k)$. Thus, $patt(u, P)$ is minimal in $JEP(D'_k, D_k)$ and, according to Lemma 2, we have $patt(u, P) \in \mathcal{L}_k$.

Consider $P \in REDLOC(u, d) \Longleftarrow patt(u, P) \in \mathcal{L}_k$. Let $patt(u, P) \in \mathcal{L}_k$. According to Lemma 2, we have: $patt(u, P) \in \mathcal{L}_k \iff patt(u, P)$ is minimal in $JEP(D'_k, D_k)$. Consider $R \subset P$. It means that $patt(u, R) \subset patt(u, P) \Longrightarrow patt(u, R) \notin JEP(D'_k, D_k)$, and, according to (1), we obtain $R(u) \not\subseteq U_k$. Summing up, according to (1) we have $P \in \{B \subseteq \mathcal{C} : B(u) \subseteq U_k\}$ and for any $R \subset P$ we have $R(u) \not\subseteq U_k$. Thus, P is minimal in $\{B \subseteq \mathcal{C} : B(u) \subseteq U_k\}$ and, according to Lemma 1, we have $P \in REDLOC(u, d)$.

4 Classification Based on Minimal Patterns

The rough set theory and emerging patterns are often used to build efficient classifiers. Although both approaches use different formal apparatus, they often employ similar algorithmic ideas.

A rough set is a convenient tool for representing approximate concepts. In particular, one of its major applications is to express the classification hypothesis provided by a decision table. Most of rough set classifiers are rule-based and make use of the notion of a reduct and its derivatives. The rule set of a classifier results from the set of reducts used against the objects in a decision table. On the other hand, classifiers based on emerging patterns operate on sets of patterns induced for each decision class. Patterns are discovered according to their characteristics in a transaction database, e.g. minimal support in a positive or negative class, minimal growth-rate, minimal chi-square test value etc.

A classification function is defined as a function $f : \mathcal{U} \mapsto V_d$, such that $f(u) = argmax_{k \in V_d} score(u, k)$, for $u \in \mathcal{U}$, where *score* is a class scoring function $score : \mathcal{U} \times V_d \mapsto \mathbb{R}$. The form of the class scoring function depends on a particular method. For a rule-based classifier, it is determined by the set of rules and the way of aggregating their significance. In fact, a decision rule $\bigwedge_{a \in P}(a = v_a) \Longrightarrow v_d$ can be equivalently expressed by $\{(a, v_a)\}_{a \in P} \Longrightarrow v_d$, for some $P \subseteq \mathcal{C}$, $v_a \in V_a$ for each $a \in P$ and $v_d \in V_d$. Thus, for the sake of this study, we assume that a rule-based classifier operates on a collection of pattern sets $\{P_k\}_{k \in V_d}$ induced for respective classes. Moreover, we use two following notations analogical to [5]. A set of all patterns assigned to a class $k \in V_d$ and matching an object $u \in \mathcal{U}$ is denoted by $MatchPatterns(u, k) = \{R \in P_k : R \subseteq patt(u, \mathcal{C})\}$. On

the other hand, a set of all objects that supports a given pattern is represented by $SupportSet(R) = \{u \in \mathcal{U} : R \subseteq patt(u, \mathcal{C})\}$. Thanks to the two-way nature of the relational-to-transactional transformation, these expressions remain true also when we operate on a respective decision transaction database.

In this study, we limit our interest to classifiers based on minimal patterns. The following sections provide a comparison of methods originating in both families. Our purpose is to demonstrate the analogical solutions and point out the main differences. Comparative accuracy tests can be found in [1,3].

4.1 Pattern Selection Methods

The rough set approach based on the concept of a local reduct is presented in [3,6]. It discovers the set $REDLOC(u, d)$ for each $u \in \mathcal{U}$ in the decision table $DT = (\mathcal{U}, \mathcal{C}, d)$ and then uses it to generate the pattern set collection $\{P_k\}_{k \in V_d}$, where $P_k = \{patt(u, R) : u \in U_k \wedge R \in REDLOC(u, d)\}$ for $k \in V_d$. A similar idea can be found in JEP-C (e.g. [1]) which computes JEP spaces for each class in the respective decision transaction database in order to obtain the collection $\{\mathcal{L}_k\}_{k \in V_d}$. According to Theorem 1, both collections are equal when DT is consistent. Otherwise, every object from outside of the positive region can generate emerging patterns that are not jumping and cannot be generalized in order to obtain JEP. In fact, JEP-C induces patterns only from the positive region of a decision data, i.e. it considers the decision transaction database for the table $(POS(\mathcal{U}), \mathcal{C}, d)$.

This difference remains true also for other propositions based on JEPs, like DeEP or DeEP-NN [7]. The assumption about consistency holds for many real data sets, especially with a large number of attributes; however, in general, the inability to make inference from non-positive data can be a weakness of the classifiers of this type. In particular, they are more prone to noisy data than approaches based on local reducts or other types of patterns, e.g. EP, chi-EP.

One of the improvements of the local reduct-based method, described in [6], is to decrease the size of a rule set by selecting the minimal set of patterns that covers the universum. The main arguments behind this idea refer to the minimum description length principle, classification efficiency and the possible generality of a discovered subset of patterns. Since this step involves solving a set covering problem, in many cases heuristic methods are employed to find a suboptimal solution. As a matter of fact, there is no such proposition for emerging patterns, however, this strategy can be applied in the similar manner. Since, the sets of jumping emerging patterns P_k are exclusive, we can solve k set covering problems, one for each class U_k, instead of dealing with $\mathcal{U}$ at once. It also means that, for inconsistent decision tables, one can obtain a more concise pattern set, when using an approach based on local reducts.

4.2 Class Scoring

Let us consider a pattern set collection $\{P_k\}_{k \in V_d}$, where P_k contains minimal patterns chosen according to some criteria. In order to complete the picture of

rule-based classifiers, popular scoring functions will be discussed. The approaches are divided into two groups depending on what is aggregated to obtain a decision.

In the first group, the algorithm takes into account a set of training objects that support any of the patterns matching a testing object. This approach is commonly used in lazy classifiers. The scoring function for lazy local reduct classification has the form: $strength(u,k) = |\bigcup_{R \in MatchPatterns(u,k)} SupportSet(R)|$. A similar idea is proposed in DeEP classifier [7], however, the number of matching objects is related to the cardinality of a respective class, i.e: $compactScore(u,k) = \frac{strength(u,k)}{D_k}$. The second formula seems to be more adequate in case of an object disproportion within the classes U_k or pattern sets P_k, both for $k \in V_d$.

The second concept focuses on direct aggregation of patterns matching a testing object. A good example is the scoring function used in JEP-C (e.g. [1]) defined as: $collectiveImpact(u,k) = \sum_{R \in MatchPatterns(u,k)} supp_{D_k}(R)$. On the other hand, eager classifiers based on local reducts employ the notion of an approximation space in order to produce a classification result. In the beginning, for a testing object u the algorithm finds a set of all patterns matching u, denoted by $\mathcal{R}$. Then, for each pattern $R \in \mathcal{R}$, the objects of $SupportSet(R)$ are analyzed in order to obtain a partial decision for this pattern. Finally, the partial results are aggregated to indicate a decision class. Although this approach is very general, in the most common case, it becomes similar to *collectiveImpact*. In fact, when we assume consistency, each pattern indicates only one class. Therefore, for a frequency-wise aggregating strategy, both classifiers are equivalent.

In practice, the result of pattern aggregation can be strongly influenced by a disproportion in the number of minimal patterns $|P_k|$ in particular classes. One of the solutions was proposed originally for CAEP (e.g. [1]) and involves dividing a score by the base score for the respective class. The base score for a class $k \in V_d$ is a selected percentile of the distribution of scores for a training data $\{score(u,k) : u \in U_k\}$, e.g. 50-80th within these scores. Last but not least, when we sum the supports of selected patterns, we use the assumption of their independent occurrence in data and ignore possible correlations that can be observed in a training set. As a result, the score can be overestimated.

5 Minimal Pattern Discovery

Due to their generality and sharp discriminating power, minimal patterns are a good basis to build accurate classifiers. However, discovering the collection of all minimal patterns for each class of a decision table can be a time-consuming task. In general, the resulting pattern sets might have an exponential cardinality, which suggests the non-polynomial complexity of any possible algorithm. This opinion is also strengthened by the NP-hardness of finding the decision rule of a minimal cardinality [3]. Moreover, even if the result is not large, there is still a possibility that temporary collections involved in computation can be exponentially large. To efficiently solve the problem for real data sets, much attention should be dedicated to identifying and optimizing the most frequent operations and to using economical data structures. In fact, there are many

propositions concerning the discovery of minimal rules and patterns [8,5]. In our study, we compare the efficiency of a local reduct approach with two different reduct computation methods [6,9] and JEP-Producer based on a border differential operation [2].

The rough set algorithm consists of two stages. First, for each object $u \in \mathcal{U}$, the set of local reducts $REDLOC(u, d)$ is computed. Then, for each local reduct $R \in REDLOC(u, d)$, a minimal pattern $patt(u, R)$ is added to the pattern set $P_{d(u)}$. Local reduct discovery determines a total time. For a wider view, we selected two reduct computation algorithms. Both methods employ a discernibility matrix [5], the structure that contains for each object pair a set of such attributes that can individually discern these objects. Then, for efficiency, one computes the discernibility vector of minimal attribute sets from this matrix. The first tested algorithm is a classic approach (e.g. [6]), based on finding all prime implicants of a monotonous boolean function. The elements of a discernibility vector are examined successively and the set of reducts is updated in every iteration. For comparison, we propose a novel apriori-based method [9]. In this approach, the search space is traversed according to an apriori scheme and the notion of attribute set dependence is used to form a pruning border.

As far as transaction databases are concerned, we study a JEP-based method. In the beginning, a decision transaction database is computed for a given decision table. Then, for each class k, we compute a space $JEP(D'_k, D_k)$ by means of JEP-Producer. Actually, the purpose is to find the left bound of this space, since, according to Theorem 2 from [4], the right bound is trivially equal to $\{t : (n, t) \in D_k\}$. First, the spaces for both classes, referred to as a positive and negative space, are equal to the respective horizontal spaces [2]. To obtain a resulting space, a border differential procedure is performed. This routine is iterative, the sets of the right bound of the positive space are examined successively. If the considered set belongs to the negative space, a specific differential procedure, named BORDER-DIFF, is called in order to correct the bounds of the positive space [2]. Finally, the set of minimal patters is equal to the left bound of the resulting space. The execution time depends mostly on the border computation.

6 Implementation Issues

One of major problems in comparing the efficiency of different algorithms is to choose an adequate methodology. Generally, in sophisticated procedures it is hard to describe the complexity in a theoretical manner. Even more troublesome is to find operations common for different approaches so as to make a reliable comparison. The algorithms studied in our paper operate on sets of attributes or items and on collections and vectors of these sets. Thus, the crucial thing is to base their implementation on the same data structures. Actually, many set implementations have been studied [10], in particular: a byte array, a bit array or a balanced tree. We tested algorithms using all these three approaches. For the comparison, we chose the first one due to its simplicity, high efficiency and absence of optimizations that can disturb the result. More specifically, the

current size of a set is stored instead of being computed on-demand. It appears important e.g. when a collection of sets is stored as cardinality-wise buckets [2].

Our implementation is coded in Java 1.5. In particular, the collection of sets is represented by a balanced tree based on the class java.util.TreeSet and a vector of sets by java.util.ArrayList. The local reduct computation is based on [6,5,9] and JEP-Producer is implemented with optimization remarks described in [2].

7 Efficiency Tests

The algorithms have been tested against consistent decision tables from [11]. We repeated executions for each data set and each method to obtain a reliable average execution time. Tests have been performed on Intel Pentium M 2.13GHz with 2GB of RAM, switched to a stable maximum performance, using the Windows XP operating system and Sun JRE 1.5.0.06.

For small datasets (lymn, zoo) all methods have similar efficiency. The classic approach scales worse than the apriori method with the number of attributes (dna, lung, mushroom). On the other hand, JEP-Producer is slower for a high number of items (dna, geo, lung, mushroom), which depends on attributes and their domain values. The results for large universums (krkopt, mushroom, nursery) suggest that JEP-Producer tends to be significantly slower than rough set approaches. Based on the tested datasets the apriori-like method [9] seems to be more appropriate, when a large number of objects or attributes is concerned.

Table 1. Experimental results summary (time in ms)

Dataset	Obj.	Attr.	Items	Red. classic time	Red. apriori time	JEP-Producer time
car	1728	6	25	880	917	5276
dna	500	20	80	3682802	72109	468937
geo	402	10	78	369	338	1796
house	435	16	48	6120	3823	3224
krkopt	28056	6	43	355052	274593	2946906
lung	32	55	220	6653937	25453	2426344
lymn	148	18	59	2406	1301	1401
tic-tac-toe	958	9	27	2276	2729	2396
zoo	101	16	39	114	114	109
nursery	12960	8	27	102823	103750	516807
mushroom	8124	22	117	117568	81854	1180822

8 Conclusions

In the paper, we have discussed the concept of minimal patterns in relational data. We have focused on two similar ideas: minimal patterns obtained from local reducts and jumping emerging patterns. As far as relational data is concerned, we have demonstrated the equivalence between both types of patterns in a positive region. Moreover, similarities are present in classification methods originating in

both streams. The pattern sets used in JEP-C and DeEP are equivalent to sets induced for positive objects in respective eager and lazy local reduct methods.

On the contrary, results for methods of minimal pattern discovery vary significantly due to differences in the form of data. Efficiency tests performed for large consistent decision tables confirmed the intuition that methods using the information held in attributes outperform the solutions operating on a more general, transactional form of data, like JEP-Producer. Nevertheless, all the methods behave similarly for small datasets and time differences are not unequivocal. The results suggest that rough set methods seem more appropriate in the analysis of large relational data. In particular, an apriori-like algorithm appears more efficient than a classic method that minimizes an indiscernibility function.

In our opinion both modes of reasoning of thought bring a number of interesting ideas that can be interchanged in order to develop more efficient methods for the analysis of relational and transactional data.

References

1. H. Fan, *Efficient Mining of Interesting Emerging Patterns and Their Effective Use in Classification.* University of Melbourne: PhD thesis, 2004.
2. G. Dong and J. Li, "Mining border descriptions of emerging patterns from dataset pairs," *Knowl. Inf. Syst.*, vol. 8, no. 2, pp. 178–202, 2005.
3. J. Wroblewski, *The Adaptive Methods for Object Classification.* Warsaw University, Institute of Informatics: PhD thesis, 2002.
4. P. Terlecki and K. Walczak, "On the relation between rough set reducts and jumping emerging patterns," *Information Sciences*, 2006 (to be published).
5. A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems," in *Intelligent Decision Support* (R. Slowinski, ed.), (Dordrecht, The Netherlands), pp. 331–362, Kluwer Academic Publishers, 1992.
6. J. Bazan, H. S. Nguyen, S. H. Nguyen, P. Synak, and J. Wroblewski, "Rough set algorithms in classification problem," *Rough set methods and applications: new developments in knowl. disc. in inf. syst.*, pp. 49–88, 2000.
7. J. Li, G. Dong, K. Ramamohanarao, and L. Wong, "Deeps: A new instance-based lazy discovery and classification system," *Mach. Learn.*, vol. 54, no. 2, pp. 99–124, 2004.
8. N. Shan and W. Ziarko, "An incremental learning algorithm for constructing decision rules," in *Rough Sets, Fuzzy Sets and Knowledge Discovery* (W. Ziarko, ed.), pp. 326–334, Springer Verlag, Berlin, 1994.
9. P. Terlecki and K. Walczak, "Attribute set dependence in apriori-like reduct computation," in *Rough Sets and Knowl. Techn.*, 2006 (to be published).
10. M. Jürgens and H.-J. Lenz, "Tree based indexes vs. bitmap indexes - a performance study.," *Int. Journal of Cooperative Inf. Syst.*, vol. 10, no. 3, pp. 355–376, 2001.
11. C. B. D.J. Newman, S. Hettich and C. Merz, "UCI repository of machine learning databases," 1998.