Mining Contextual Preference Rules for Building User Profiles

Sandra de Amo[1], Mouhamadou Saliou Diallo[2,3], Cheikh Talibouya Diop[3],
Arnaud Giacometti[2], Haoyuan D. Li[2], and Arnaud Soulet[2]

[1] Universidade Federal de Uberlândia, Brazil, deamo@ubu.br
[2] Université de Tours, France, forename.surname@univ-tours.fr
[3] Université Gaston Berger de Saint-Louis, Sénégal, forename.surname@ugb.edu.sn

Abstract. The emerging of ubiquitous computing technologies in recent years has given rise to a new field of research consisting in incorporating context-aware preference querying facilities in database systems. One important step in this setting is the *Preference Elicitation* task which consists in providing the user ways to inform his/her choice on pairs of objects with a minimal effort. In this paper we propose an automatic preference elicitation method based on mining techniques. The method consists in extracting a *user profile* from a set of user preference samples. In our setting, a profile is specified by a set of contextual preference rules verifying properties of soundness and conciseness. We evaluate the efficacy of the proposed method in a series of experiments executed on a real-world database of user preferences about movies.

1 Introduction

Elicitation of preferences consists basically in providing the user a way to inform his/her preferences on objects belonging to a dataset, with a minimal effort for him/her. It can be achieved by following different strategies: (a) by using a query interface where users are asked to express their preferences [**?**], or (b) by capturing implicit user's choices and applying preference mining algorithms [**?**]. The first alternative is not efficient since the users in general are not able to express their preferences in an exact and consistent way. This paper is focused on the second alternative for preference elicitation. We assume our data is constituted by pairwise comparisons. We do not discuss in this paper the way the user informed his/her choices, knowing that different strategies can be applied [**?**]. Our method simply assume that *pairs of objects* expressing the user preferences have been collected somehow. The running example below illustrates the preference mining problem we tackle in this paper. In this example we assume that the user preferences are informed by means of the number of clicks on certain tags.

Motivating Example. A web service regularly provides recommendation about movies to its subscribers. In order to capture my preferences on films without being too annoying and intrusive, the service offers me a trial period during which I can freely access information about films. I indicate the films I am interested in by clicking on different tags. For instance, I can click on tags *Action*, *Spielberg* and *War* to indicate that I am interested in obtaining information on films directed by Steve Spielberg, with a script based on a war story, and having a lot of action. My clicks are automatically collected during the trial period. The relation $\mathcal{D}$ depicted on Table 1 presents some of my access during the trial period. Tags A, B, C, D and E stands for *Spielberg*, *Tom Hanks*, *Action*, *Leonardo di Caprio* and *War* respectively. Each t_i $(i = 1, ...5)$ represents the set of tags I selected each time I accessed the service. They are called *transactions*. Let us suppose that during the trial period I accessed the service ten times by clicking on the set of tags t_1 and only five times by clicking on the set of tags t_3. Thus, I implicitly indicated that I am more interested on films associated to tags t_1 than to tags t_3 as indicated by the first pair (t_1, t_3) in relation $\mathcal{P}$ depicted on Table 1. Notice that both t_1 and t_3 contain the tags A and C. Between them I prefer the one containing the tag D than the one containing the tag B. So, the following *contextual preference rule* can be inferred: Between two *action* movies directed by *Spielberg* I prefer the one played by Leonardo di Caprio than the one played by Tom Hanks. Tags *Action* and *Spielberg* constitute the *context* of the rule. Notice that some pairs of transactions (for instance, (t_1, t_2)) do not appear in relation $\mathcal{P}$, indicating that the number of clicks on each of these sets of tags is the same or differs by a *negligible* amount of clicks (below a given threshold).

$\mathcal{D}$

Tid	Transactions
t_1	$A\ C\ D$
t_2	$A\ B\ D$
t_3	$A\ B\ C\ E$
t_4	$C\ D$
t_5	$A\ B$

$\mathcal{P}$

Pid	Preference
p_1	$\langle t_1, t_3 \rangle$
p_2	$\langle t_2, t_3 \rangle$
p_3	$\langle t_2, t_4 \rangle$
p_4	$\langle t_3, t_4 \rangle$
p_5	$\langle t_4, t_5 \rangle$

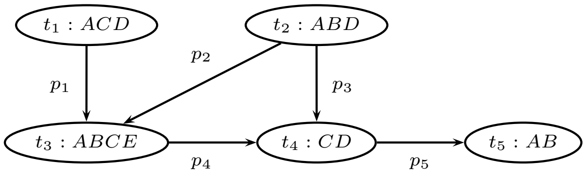

Fig. 1. Preferences on Transactions

In this paper we propose a method for building the profile of a user from a sample of his/her preferences previously captured by the system. A user's profile is specified by a set of *contextual preference rules* [**?**] satisfying some interestingness criteria, namely *soundness* and *conciseness*. The *soundness* property guarantees that the preference rules specifying the profiles are in agreement with a large set of the user preferences, and contradicts a small number of them. On the other hand, *conciseness* implies that profiles are small sets of preference rules. We argue that this approach has many advantages if compared to other preference models found in the literature. The model is easy to understand and manage due to its conciseness and its *qualitative* aspect (it is constituted by a set of preference rules and it does not employ score function explicitly assigning grades to each transaction [**?**,**?**,**?**,**?**]). Moreover, the soundness property guarantees that our method builds user profiles with good predictive properties.

This paper is organized as follows. In Section 2 we discuss some related work. In Section 3 we rigorously define the mining problems we treat in this paper. Section 4 is dedicated to present the preference rule mining algorithm, whereas Section 5 presents the user profile construction algorithm. In Section 6 we describe and analyze experimental results on real datasets.

2 Related Work

Methods for Preference Learning can be categorized following different criteria such as *Preference Specification* (*qualitative* or *quantitative*) and *Preference Semantics* (the pareto model, conditional preference model). The techniques presented in this section are inherently distinct. Nevertheless they have a common main goal: given a pair of objects, to predict which one is the most preferred.

In a *qualitative approach*, preferences are specified by a compact set of preference rules from which a preference relation can be inferred. The method we propose in this paper follows a qualitative approach. Some other qualitative approaches are [**?**,**?**]. In [**?**] the authors propose a technique for mining user preferences whose underlying model is the *pareto preference model*. Such preference rules are obtained from log data generated by the server when the user is accessing a web site. Another approach to preference mining is presented in [**?**]. In this work the authors propose using preference samples provided by the user to infer an order on any pair of tuples in the database. Such samples are classified into two categories, the *superior* and *inferior* samples and contain information about some preferred tuples and some non-preferred ones. From these rules, an order is inferred on the tuples. The underlying preference model is the *pareto preference model* as in [**?**]. In this model, preferences are not conditional or contextual, that is, preferences on values of attributes do not depend on the values of other attributes. Our contextual preference model is more expressive.

In contrast with the above papers, where preferences are specified following a *qualitative* approach, in [**?**] and [**?**] algorithms for mining *quantitative* preferences are proposed. In these works preferences are specified by a score function and the main goal is to find automatically a prediction rule which assigns a score to each tuple of the database. The mining task in this approach is sometimes called *learning to rank*. Several efficient methods for learning to rank have been proposed so far in the information retrieval domain, including Rank SVM [**?**], RankBoost [**?**], RankNet [**?**] and AdaRank [**?**]. In all these methods, the learning task is formalized as classification of object pairs in two classes: correctly or incorrectly ranked. Different classification techniques are employed such as SVM (Rank SVM), Boosting (AdaRank, RankBoost) and Neural Network trained by a Gradient Descent algorithm (RankNet). In comparison, our method has the advantage of making explicit the preferences of the user through the profile.

3 Problem Formalization

3.1 Preference Database and Contextual Preference Rules

Let $\mathcal{I}$ be a set of distinct literals called *items* (or *tags*), an itemset is a subset of $\mathcal{I}$. The language of itemsets corresponds to $\mathcal{L} = 2^{\mathcal{I}}$. A transactional dataset

$\mathcal{D}$ is a multi-set of itemsets in $\mathcal{L}$. Each itemset, usually called *transaction*, is a database entry. Figure 1 presents a transactional dataset $\mathcal{D}$ where 5 transactions denoted by $t_1, \ldots, t_5$ are described by 5 items denoted by $A, \ldots, E$.

A *preference database* $\mathcal{P} \subseteq \mathcal{D} \times \mathcal{D}$ is a set of pairs of transactions representing a sample of user preferences over the dataset $\mathcal{D}$. Intuitively, a user preference $\langle t, u \rangle \in \mathcal{P}$ means that the user prefers the transaction t to the transaction u. Given a user preference $\langle t, u \rangle \in \mathcal{P}$, t is said to be the *preferred* transaction (according to the user). Figure 1 shows a set of 5 user preferences labeled $p_1, \ldots, p_5$. The preference database plus the transactions are also synthesized by a graph as illustrated in Table 1[4]. We emphasize that in general $\mathcal{P}$ is not necessarily *transitive* as in our running example, since in this particular case the user preferences have been obtained by comparing the number of clicks on each set of tags.

The main objective of this paper is to extract a user profile from a preference database provided by the user. A user profile is specified by a set of preference rules verifying some interesting properties.

Definition 1 (Contextual preference rule [?]). *A contextual preference rule is of the form* $i^+ \succ i^- \,|\, X$ *where* X *is an itemset of* $\mathcal{L}$*,* i^+ *and* i^- *are items of* $\mathcal{I} \setminus X$*.*

The left-hand side of a preference rule specifies the choice while the right-hand side is the context. For instance, $D \succ E \,|\, AB$ means that the context AB leads to choose the item D to the item E. $\mathcal{CP}(\mathcal{L})$ denotes the set of the contextual preference rules based on $\mathcal{L}$ (we often omit the language when it is implicit in the context). Of course, the interest behind $i^+ \succ i^- \,|\, X$ is its ability to compare transactions. A transaction t is preferred to u according to $\pi : i^+ \succ i^- \mid X$, denoted by $t \succ_\pi u$ if $(Xi^+ \subseteq t) \wedge (Xi^- \subseteq u) \wedge (i^- \notin t) \wedge (i^+ \notin u)$. For instance, ACD is preferred to $ABCE$ according to the contextual preference rule $D \succ E \,|\, A$ i.e., $ACD \succ_{D \succ E|A} ABCE$.

Naturally, a given contextual preference rule π can *agree with* a user preference $\langle t, u \rangle \in \mathcal{P}$ (i.e. $t \succ_\pi u$) or *contradict* $\langle t, u \rangle \in \mathcal{P}$ (i.e. $u \succ_\pi t$). In both cases, we say that the contextual preference rule *covers* the user preference $\langle t, u \rangle$. For instance, the user preference $p_1 = \langle t_1, t_3 \rangle$ is covered by both $D \succ E \,|\, A$ (agreement) and $B \succ D \,|\, C$ (contradiction).

3.2 The Contextual Preference Rule Mining Problem

Basically, we adapt the support-confidence framework of association rules by considering that the context X and the preference $i^+ \succ i^-$ corresponds respectively to the antecedent and the consequent of an association rule. Thereby, we analogically define the concept of support, confidence and minimality as interestingness criteria for filtering out non relevant contextual preference rules.

Definition 2 (Support). *The support of a contextual preference rule* π *in* $\mathcal{P}$ *is defined as:* $supp(\pi, \mathcal{P}) = \frac{|\{\langle t,u \rangle \in \mathcal{P} \;|\; t \succ_\pi u\}|}{|\mathcal{P}|}$

[4] For the sake of simplifying the presentation, some arrows obtained by transitivity are not depicted in the graph (for instance the arrow between t_1 and t_4).

The support of a contextual preference rule π (ranged from 0 to 1) estimates the probability that π agrees with a pair of $\mathcal{P}$. The interest of a contextual preference rule increases with its support. For instance, as $ACD \succ_{D \succ E|A} ABCE$ and $ABD \succ_{D \succ E|A} ABCE$, we obtain $supp(D \succ E \,|\, A, \mathcal{P}) = |\{p_1, p_2\}|/|\mathcal{P}| = 0.4$. Similarly, $supp(D \succ E \,|\, B, \mathcal{P}) = |\{p_2\}|/|\mathcal{P}| = 0.2$. So, $D \succ E \,|\, A$ is more interesting than $D \succ E \,|\, B$.

Now we also need to evaluate the disagreement between a contextual preference rule and the preference database. To this end, the *confidence* of a contextual preference rule π measures the proportion of user preferences in agreement with π among pairs covered by π:

Definition 3 (Confidence). *The confidence of a contextual preference rule π in $\mathcal{P}$ is defined as:* $conf(\pi, \mathcal{P}) = \frac{|\{\langle t,u \rangle \in \mathcal{P} \ | \ t \succ_\pi u\}|}{|\{\langle t,u \rangle \in \mathcal{P} \ | \ t \succ_\pi u \vee u \succ_\pi t\}|}$

In other words, the confidence evaluates whether a contextual preference rule contradicts many user preferences. This criterion shows that $D \succ E \,|\, A$ is more valuable than $D \succ E \,|\, \emptyset$ because $conf(D \succ E \,|\, A, \mathcal{P}) = 2/2 = 1$ whereas $conf(D \succ E \,|\, \emptyset, \mathcal{P}) = 2/3$. The set of all contextual preference rules exceeding a minimal support threshold σ and a minimal confidence threshold κ is denoted by $\mathcal{CP}_{\sigma,\kappa}(\mathcal{L}, \mathcal{P})$ (or $\mathcal{CP}_{\sigma,\kappa}$ in brief).

At this point, the support and the confidence discard respectively the infrequent and unreliable contextual preference rules. But, the redundancies between several contextual preference rules of $\mathcal{CP}_{\sigma,\kappa}$ are not detected. Given the example of Figure 1, we observe that $D \succ E \,|\, B$ and $D \succ E \,|\, AB$ have the same support and the same confidence. Intuitively, the contextual preference rule $D \succ E \,|\, B$ is more relevant than $D \succ E \,|\, AB$ because its context is smaller. For this purpose, we introduce the notion of *minimal* contextual preference rule:

Definition 4 (Minimal preference rule). *A contextual preference rule $i^+ \succ i^- \,|\, X$ is minimal in $\mathcal{P}$ iff there is no contextual preference rule $i^+ \succ i^- \,|\, Y$ such that $Y \subset X$ and $supp(i^+ \succ i^- \,|\, Y, \mathcal{P}) = supp(i^+ \succ i^- \,|\, X, \mathcal{P})$ and $conf(i^+ \succ i^- \,|\, Y, \mathcal{P}) = conf(i^+ \succ i^- \,|\, X, \mathcal{P})$.*

Following on, $\mathcal{MCP}_{\sigma,\kappa}(\mathcal{L}, \mathcal{P})$ (or $\mathcal{MCP}_{\sigma,\kappa}$) denotes the whole set of minimal contextual preference rules having its support and confidence respectively greater than σ and κ. In practice, this minimality criterion drastically reduces the number of contextual preference rules.

Given a sample preference database, the first problem that we consider deals with the extraction of all *interesting* preference rules, i.e. those rules which are minimal and have acceptable support and confidence. More precisely:

Problem 1 (Preference Rule Mining). Given a preference database $\mathcal{P}$, a minimal support threshold σ and a minimal confidence threshold κ, find the set $\mathcal{MCP}_{\sigma,\kappa}$ of minimal contextual preference rules.

Obviously, a naive enumeration of all preference rules for computing $\mathcal{MCP}_{\sigma,\kappa}$ is unfeasible and some pruning criteria are necessary for reducing the search space. In Section 4 we present ContPrefMiner, a levelwise algorithm inspired on Apriori [?] which takes advantage of the downward closure of $\mathcal{MCP}_{\sigma,0}$ to reduce the search space.

3.3 The User Profile Construction Problem

In our approach, a *user profile* is specified by a set of contextual preference rules which is both *concise* and *sound* with respect to the preference samples he/she has previously provided. Roughly speaking, the *conciseness* of a set of preference rules is evaluated by means of its cardinality. On the other hand, the *soundness* of a set of preference rules is evaluated by means of two standard measures, *precision* and *recall* (see Definition 5).

We have to precise how two transactions can be compared according to a set S of contextual preference rules to evaluate the ability of a user profile to make good predictions. First, we say that two transactions are *comparable* with respect to a set S of preference rules if they can be compared by at least one rule in S. Then, one important issue is when two transactions can be compared in different ways using different rules in S. In this paper, we define a total order on the set of contextual preference rules (see Definition 6), and propose to select the *best* preference rule to decide which transaction is the preferred one. More precisely, we say that a transaction $t \in \mathcal{L}$ is preferred to $u \in \mathcal{L}$ according to a user profile S, denoted by $t \succ_S u$, it there exists a preference rule $\pi \in S$ such that $t \succ_\pi u$ and π is the best rule in S that can be used to compare t and u.

In order to evaluate the predictive quality of a user profile, we now introduce the *precision* and *recall* measures as follows:

Definition 5 (Precision and recall). *Given a preference database $\mathcal{P}$ and a set of contextual preference rules S, the precision of $\succ_S$ with respect to $\mathcal{P}$, denoted $Prec(\succ_S, \mathcal{P})$, is defined by:*

$$Prec(\succ_S, \mathcal{P}) = \frac{|\{\langle t,u\rangle \in \mathcal{P} | t \succ_S u\}|}{|\{\langle t,u\rangle \in \mathcal{P} | t \succ_S u \vee u \succ_S t\}|}$$

Moreover, the recall of $\succ_S$ with respect to $\mathcal{P}$, denoted $Rec(\succ_S, \mathcal{P})$, is defined by:

$$Rec(\succ_S, \mathcal{P}) = \frac{|\{\langle t,u\rangle \in \mathcal{P} | t \succ_S u\}|}{|\mathcal{P}|}$$

Notice that if S is a singleton then the precision and recall of S coincide with the confidence and support of the single rule in S.

Using Definition 5, we can now define precisely the second main problem we consider in this paper, i.e. the construction of a user profile that is concise and sound with respect to a set of user preferences.

Problem 2 (User profile construction). Given a preference database $\mathcal{P}$ and a set of contextual preference rules S, select $\Pi \subseteq S$ that maximizes precision and recall with respect to $\mathcal{P}$ and that is as concise as desired. Π is called the *user profile* associated to $\mathcal{P}$.

Notice that in this problem statement, S can be any set of preference rules. In practice, S will be the set of all interesting minimal contextual preferences rules, as defined in problem 1. It is also important to note that with large datasets, the construction of the smallest set of preference rules that maximizes recall and precision is a hard problem. Indeed, it can be shown that this problem is closely related to the red-blue set cover problem that is NP-complete [**?**][5]. To cope with this difficulty, we propose in Section 5 a heuristic approach based on the same ideas used by associative classification methods such as CBA [**?**]. More precisely, given a preference database, a set of interesting preference rules and a parameter k (called *minimal agreement threshold*) that allows to control the size of the user profile returned, we present an iterative algorithm called ProfMiner that maximizes precision and recall.

4 Discovery of Contextual Preference Rules

As indicated in the previous section, we cope with Problem 1 by using pruning criteria stemming from anti-monotone constraints that reduce the search space $\mathcal{CP}$. Before detailing the proposed algorithm, let us recall that a constraint q is anti-monotone iff whenever $i^+ \succ i^- \,|\, X$ satisfies q, any generalization of $i^+ \succ i^- \,|\, X$ (i.e., $i^+ \succ i^- \mid Y$ such that $Y \subseteq X$) also satisfies q. Such constraints like the minimal support provide powerful pruning conditions of the search space [**?**]. Interestingly, the minimality leads to another anti-monotone constraint: whenever a contextual preference rule $i^+ \succ i^- \,|\, X$ is minimal, all the contextual preference rules $i^+ \succ i^- \mid Y$ satisfying $Y \subseteq X$ are also minimal. As an example, let us consider $r : D \succ E \mid AB$ with $\mathit{supp}(r, \mathcal{P}) = 0.2$ and $\mathit{conf}(r, \mathcal{P}) = 1$. Since r is not a minimal contextual rule (because $\mathit{supp}(r, \mathcal{P}) = \mathit{supp}(D \succ E \mid B, \mathcal{P})$ and $\mathit{conf}(r, \mathcal{P}) = \mathit{conf}(D \succ E \mid B, \mathcal{P})$), we are sure that there is no more minimal rule concluding on $D \succ E$ containing AB in its context. Such pruning technique drastically reduces the search space in a levelwise mining method as Algoritm 1.

Now we detail ContextualPreference rule Miner where the set $\mathcal{C}and_i$ (resp. $\mathcal{MCP}_i$) contains all the candidates (resp. minimal contextual rules) whose context has a cardinality i. Basically, Line 1 initializes the candidates with rules having an empty context. For this purpose, all the pairs of items (i_1, i_2) are considered. While there are candidates of context length i, Line 4 computes all the minimal contextual preference rules of length i satisfying the constraint $\mathit{supp}(r, \mathcal{P}) \geq \sigma$ (test step). Line 5 generates the new candidates of length $i + 1$ (generate step). Finally, Line 8 returns the complete collection of the minimal contextual preference exceeding a minimal confidence threshold (with the corresponding support and confidence).

[5] Given a finite set of "'red" elements R (here, $\langle u, t\rangle$ such that $\langle t, u\rangle \in \mathcal{P}$), a finite set of "blue" elements B (here, $\mathcal{P}$) and a family of $\mathcal{S} \subseteq 2^{R \cup B}$, the red-blue set cover problem is to find a subfamily $S \subseteq \mathcal{S}$ which covers all blue elements, but which covers the minimum possible number of red elements.

Algorithm 1 ContPrefMiner

```
Input: A preference database $\mathcal{P}$, a minimal support threshold $\sigma$, a minimal confidence
    threshold $\kappa$
Output: All the minimal contextual preference rules with support and confidence
    exceeding $\sigma$ and $\kappa$ respectively.
$\mathcal{C}and_0 := \{i_1 \succ i_2 \,|\, \emptyset \in \mathcal{CP}$ such that $(i_1, i_2) \in \mathcal{I} \times \mathcal{I}\}$
$i := 0$
while $\mathcal{C}and_i \neq \emptyset$ do
    $\mathcal{MCP}_i := \{r \in \mathcal{C}and_i$ such that $r$ is minimal and satisfies $supp(r, \mathcal{P}) \geq \sigma\}$
    $\mathcal{C}and_{i+1} := \{i_1 \succ i_2 \,|\, X \in \mathcal{CP}$ such that $|X| = i + 1$ and $\forall i \in X, i_1 \succ i_2 \,|\, X \setminus \{i\} \in$
    $\mathcal{MCP}_i\}$
    $i := i + 1$
od
return $\{(r, supp(r, \mathcal{P}), conf(r, \mathcal{P})) \mid r \in \bigcup_{j<i} \mathcal{MCP}_j \wedge conf(r, \mathcal{P}) \geq \kappa\}$
```

5 User Profile Construction

Basically, the construction of the user profile iterates two main principles over the contextual preference rules returned by ContPrefMiner until all user preferences in the database are in agreement with at least one preference rule in the profile: (1) select the best contextual preference rule and (2) remove the unnecessary contextual preference rules. Indeed, even if the minimality criterion removes many redundant contextual preference rules, some superfluous contextual preference rules remain among those returned by ContPrefMiner. For instance, in our running example the preference rule $D \succ B \,|\, A$ (only in agreement with p_1) can be removed from $\mathcal{MCP}_{0.2,0.6}$ (see Table 1) since $D \succ E \,|\, A$ already agrees with p_1 and has a better support (with the same confidence). More generally, a contextual preference rule π provides a substantial value if it agrees with user preferences of $\mathcal{P}$ that are not in agreement with other better preference rules. Note that such a kind of iterative process for building a model is quite classical in the literature [**?**].

5.1 Ordering Contextual Preference Rules

The main strategy of the algorithm ProfMiner responsible for building user profiles is the ability of selecting the *best* contextual rule to decide which transaction is the preferred one. The following definition introduces a total order on the set of contextual preference rules $\mathcal{MCP}$.

Definition 6 (Best rule order). *The best rule order on $\mathcal{MCP}$, denoted by $>_{best}$, is a total order defined for any contextual preferences π and π' as:*

$$\pi >_{best} \pi' \Leftrightarrow \begin{cases} conf(\pi) > conf(\pi') \text{ or} \\ conf(\pi) = conf(\pi') \text{ and } supp(\pi) > supp(\pi') \text{ or} \\ conf(\pi) = conf(\pi') \text{ and } supp(\pi) = supp(\pi') \\ \qquad \text{and } |context(\pi)| < |context(\pi')| \text{ or} \\ conf(\pi) = conf(\pi') \text{ and } supp(\pi) = supp(\pi') \\ \qquad \text{and } |context(\pi)| = |context(\pi')| \text{ and } \pi <_{\mathcal{CP}} \pi' \end{cases}$$

As the profile should contradict at most a very small number of user preferences (in order to have a high precision), the confidence is the most important criterion. The support criterion naturally comes in second place, followed by the size of the context. The fourth criterion (where $<_{\mathcal{CP}}$ is an arbitrary total order) is only used to definitely decide between two indistinguishable rules.

Table 1 (the left part) illustrates the best rule order $>_{best}$ over the minimal contextual preference rules with $\sigma = 0.2$ and $\kappa = 0.6$ on our running example. Note that the arbitrary order $<_{\mathcal{CP}}$ justifies to arrange $A \succ C \,|\, D$ before $A \succ D \,|\, C$ and $B \succ C \,|\, D$ as well as $D \succ B \,|\, A$ before $D \succ E \,|\, AC$.

$\mathcal{MCP}_{0.2,0.6}$				Profile construction			
Cont. pref.	*supp*	*conf*	Agreement	step 1	step 2	step 3	step 4
$D \succ E \,|\, A$	0.4	1	p_1,p_2	□			
$D \succ C \,|\, \emptyset$	0.2	1	p_2	□			
$A \succ C \,|\, D$	0.2	1	p_3		□		
$A \succ D \,|\, C$	0.2	1	p_4			□	
$B \succ C \,|\, D$	0.2	1	p_3		□		
$D \succ B \,|\, A$	0.2	1	p_1	□			
$D \succ E \,|\, B$	0.2	1	p_2	□			
$D \succ E \,|\, AC$	0.2	1	p_1	□			
$D \succ B \,|\, \emptyset$	0.4	2/3	p_1,p_5				□
$D \succ E \,|\, \emptyset$	0.4	2/3	p_1,p_2	□			

Table 1. Rules of $\mathcal{MCP}_{0.2,0.6}$ ordered according $>_{best}$ and profile construction ($k = 1$)

5.2 The Algorithm ProfMiner

Given a preference database $\mathcal{P}$, a set of contextual preference rules S, a minimal agreement threshold k, ProfMiner returns a user profile Π by selecting suitable contextual preference rules from S (see Algorithm 2). Note that the agreement threshold k enables us to adjust the size of the user profile as indicated in Problem 2. The greater the minimal agreement k, the smaller the profile.

After initializing the profile (Line 1), the main loop (Line 2-7) selects the best contextual preference rule according to $>_{best}$ (Line 3) and adds it to the profile (Line 4) until that S becomes empty (Line 2). This condition is ensured by the reduction of $\mathcal{P}$ (Line 5) and the reduction of S (Line 6). Indeed, a contextual preference rule π is unnecessary with respect to the profile in progress whenever π does not agree with at least k remaining user preferences (i.e., not still in agreement with other preference rules of the profile).

Table 1 (the right part) illustrates ProfMiner on our running example (see Figure 1) with $S = \mathcal{MCP}_{0.2,0.6}$ and $k = 1$. At the first iteration, Line 3 selects $D \succ E \,|\, A$ (symbol □) as the best rule according to $>_{best}$ (see Table 1). Line 5 removes the user preferences p_1 and p_2 and then, Line 6 removes 5 contextual

Algorithm 2 ProfMiner

```
Input: A preference database $\mathcal{P}$, a set of preference rules $S$, a minimal agreement $k$
Output: A user profile $\Pi$
$\Pi := \emptyset$
while $S \neq \emptyset$ do
    $\pi_{best} = \max_{>_{best}} S$
    $\Pi := \Pi \cup \{\pi_{best}\}$
    $\mathcal{P} := \{\langle t, u \rangle \in \mathcal{P} | t \not\succ_{\pi_{best}} u\}$
    $S := \{\pi \in S | supp(\pi, \mathcal{P}) \geq k/|\mathcal{P}|\}$
od
return $\Pi$
```

preference rules from S (symbol □). Note that $D \succ B \,|\, \emptyset$ is preserved because it also covers p_5. The second iteration adds $A \succ C \,|\, D$ to the profile because it is the best remaining contextual preference rule. This process stops at the end of the 4th iteration because S is empty (see Line 2 of ProfMiner). So, the final profile is $\{D \succ E \,|\, A, A \succ C \,|\, D, A \succ D \,|\, C, D \succ B \,|\, \emptyset\}$.

6 Experimental Results

This experimental study aims at evaluating the conciseness and the soundness of user profiles mined by our approach. Indeed, a comprehensive study of the effectiveness of our approach has been conducted on real world datasets based on the APMD-Workbench [?] built from MovieLens (`www.movielens.org`) and IMDB (`www.imdb.com`). The used datasets and detailed data preparation process are available on the CPrefMiner project repository (`www.lsi.ufu.br/cprefminer/`). All the tests were performed on a 3 GHz Intel processor with Windows XP operating system and 1 GB of RAM memory. The overall process of preference rule mining and user profile construction is performed in at most 113 seconds, for the largest preference database P30000 described below.

Database	Items ($\mathcal{I}$)	Trans. ($\mathcal{D}$)
P301	125	32
P3000	342	99
P30000	857	309

Fig. 2. Real world preference databases over movies

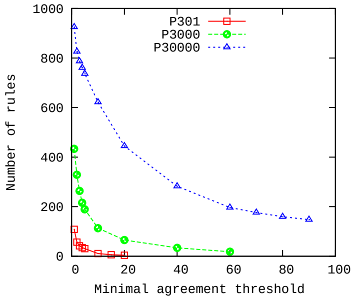

Fig. 3. Number of preference rules per profile w.r.t various k values.

Basically, the datasets consist in 6 user preference databases about movies, one database per user. In each database, a user preference about movies is represented by a pair of movie records $\langle m_1, m_2 \rangle$ meaning that "the user prefers the movie m_1 to the movie m_2". A movie record is based on a set of attributes such as Genre, Director, Years, Actor, etc. Genre, Director and Actor are multi-valued attributes. Hence, to apply our approach relying on contextual preference rules, we shall itemize each distinct attribute value so that each movie record becomes a transaction corresponding to our data model. Due to the space limitation, we only present the experimental results of 3 preference databases named P301, P3000, and P30000 as shown by Figure 2. The results on the 3 other preference databases are very similar. Each database is named by its number of user preferences, e.g., the database P301 contains 301 user preferences corresponding to a set $\mathcal{D}$ of 32 distinct movie records described by a set $\mathcal{I}$ of 125 distinct items.

For each preference database, the user profile mining and preference prediction have been performed using a 10-fold cross-validation method, and the metric values (e.g., precision and recall) on the different iterations are averaged to yield an overall one. The minimal contextual preference rules are mined using ContPrefMiner with $\sigma = 0.001$ and $\kappa = 0.5$. Note that other minimal thresholds have been tested (not reported here due to space limitation) showing that the increase of σ systematically damages the quality of user profiles while the increase of κ has a lower impact. The user profile construction was done with ProfMiner by varying the minimal agreement threshold k.

Contextual preference rule	Support	Confidence
1. LAN:German $\succ$ LAN:English $\vert$ $\emptyset$	0.017	1.00
2. GEN:Fantasy $\succ$ GEN:War $\vert$ GEN:Drama	0.015	1.00
3. GEN:Crime $\succ$ GEN:Adventure $\vert$ GEN:Action	0.012	1.00
4. GEN:Crime $\succ$ GEN:Horror $\vert$ GEN:Sci-Fi	0.012	1.00
5. GEN:Romance $\succ$ GEN:War $\vert$ GEN:Drama	0.011	1.00
6. GEN:Crime $\succ$ GEN:Adventure $\vert$ GEN:Sci-Fi	0.010	1.00
7. GEN:Crime $\succ$ GEN:Drama $\vert$ $\emptyset$	0.010	1.00
8. GEN:Fantasy $\succ$ GEN:Action $\vert$ GEN:Drama	0.009	1.00
9. LAN:German $\succ$ LAN:Vietnamese $\vert$ GEN:War	0.009	1.00
10. GEN:Sci-Fi $\succ$ GEN:Western $\vert$ GEN:Action	0.009	1.00

Table 2. Top-10 preference rules discovered from the database P3000 ($k = 1$).

We start by analyzing the conciseness of the user profile according to the minimal agreement threshold. Figure 3 plots the number of preference rules when the minimal agreement threshold k varies from 1 to 90. Even with $k = 1$, the number of preference rules contained in the user profile is drastically reduced compared to the inital number of contextual preference rules: from 5319.4 to 108.7 for P301; from 4833.9 to 432.9 for P3000; and from 4913.3 to 925 for P30000. Moreover, Figure 3 shows that the size of the user profile rapidly decreases with k and then, the user profile can be as concise as desired by the user.

The preference prediction was performed using the orders induced by the profile. Figure 4 estimates the effectiveness of the user profiles according to their size. For facilitating comparisons between the different preference databases, the size of a user profile $|\Pi|$ is normalized by means of the *profile reduction rate* defined by $(|\Pi_{k=1}| - |\Pi|)/|\Pi_{k=1}|$ where $|\Pi_{k=1}|$ is the cardinality of the user profile obtained from $k = 1$.

Figure 4 reports the precision, the recall and F-measure (i.e., $2 \times precision \times recall/(precision + recall)$) for the profile when the profile reduction rate varies. The first important observation is that the predictive quality of the mined profiles can be very high . More precisely, the precision always remains very high, while the recall deeply depends on the size of the user profile.

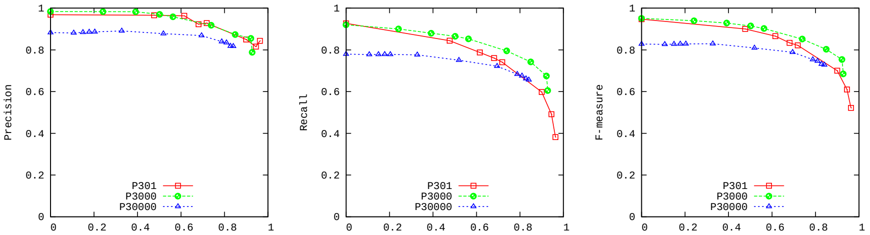

Fig. 4. Predictive quality of constructed profiles.

In brief, this set of experiments demonstrates that the conciseness of user profiles is controlled by the minimal agreement threshold and that even with strong reduction, the soundness of the profile remains at an acceptable level. But what does the mined profiles look like? Table 2 lists the top-10 preference rules of a user profile discovered from the database P3000 with $k = 1$. It demonstrates that a mined profile is easy readable. For example, rules 1 and 9 means that the user prefers german movies than english or vietnamese movies. We can also see that the user especially enjoys crime movies (see rules 3, 4, 6 and 7). Between two drama movies, this profile finally shows that the user prefers fantasy movies than war movies (see rule 2) or action movies (see rule 8), that he/she prefers romance movies than war movies (see rule 5).

7 Conclusion and Future Work

In this paper we proposed the method ProfMiner for mining user profiles from preference databases. A set of experiments on a real-world database of user preferences about movies showed the efficiency of the method. More interestingly, our approach is the first one to build readable user profile based on the notion of contextual preference rules.

The overall aim of a profile is to order a set of transactions. Thus, it would be expected that the preference relation associated to the user profile be a strict partial order over transactions. However, this is not the case since the induced order is not transitive in general. Presently, we are developing two other methodologies for extracting a strict partial order from a given set of pairs of transactions, one based on Bayesian Network classifiers and other based on a voting system.

As future work ,we finally plan to compare the predictive quality of our method with well-known ranking methods as RankNet, Rank SVM, Ada Rank and RankBoost [**?**,**?**,**?**,**?**], knowing that existing prototypes that implement these methods have to be adapted (in order to take directly as input pairwise preferences, and not only quantitative preferences).

Acknowledgments. We thank the French Embassy of Dakar, the Brazilian Research Agencies CNPq, CAPES (SticAmSud Project 016/09) and FAPEMIG for supporting this work.