CPrefMiner: A Bayesian Miner of Conditional Preferences

Nádia Félix F. da Silva, Sandra de Amo

Universidade Federal de Uberlândia, Brazil
Faculdade de Computação
nadia.felix@gmail.com, deamo@ufu.br

Abstract. Customizing database queries by considering user preferences is a research topic that has been raising a lot of interest within the database community in recent years. Such preferences are used for sorting and selecting the best tuples, those which most fulfill the user wishes. A topic of interest within this context is the elicitation of preferences, consisting of methods to enable the user to inform his choice on pairs of objects belonging to a database. Depending on the size of the database, this task may require a great effort from the user, and consequently may discourage him/her to use the system. In this paper, we propose a first step towards the design and implementation of an automatic tool for inferring preferences from a given sample of user preferences. The method *CPrefMiner* we propose is based on the framework of Bayesian Networks and aims at mining a special kind of preferences, the *conditional preferences.* The two main learning tasks accomplished by *CPrefMiner* are: (1) learning the graph underlying the conditional preference network; (2) learning the preference probability tables associated with each node of the graph. This paper focuses on the first task.

Categories and Subject Descriptors: H. Information Systems [**H.m. Miscellaneous**]: Databases

Keywords: preference mining, elicitation of preferences, conditional preferences, preference learning

1. INTRODUCTION AND MOTIVATION

The development of recommendation systems has been attracting a lot of interest in several application areas such as electronic commerce and marketing. In order to satisfy this demand, the database community has been studying ways of employing user preferences as a tool for customizing queries. A topic of interest in this context is the *elicitation of preferences* which basically consists in providing the user a way to inform his/her choice on pairs of objects belonging to a database table, with a minimal effort for the user.

Preference elicitation can be formalized under either a *quantitative* or a *qualitative* framework. In order to illustrate the elicitation of preferences under a quantitative form, consider we are given a collection of movies and we wish to know which films are most preferred by a certain user. For this, we can ask the user to rate each movie and after that we simply select those films with the higher score. This method may be impractical when dealing with a large collection of movies. In order to accomplish the same task using a *qualitative* formulation of preferences, we can ask the user to inform some generic rules that reflect his/her preferences. For example, if the user says that he/she prefers romance movies to drama movies, then we can infer a class of favorite movies without asking the user to evaluate each film individually.

A qualitative framework for preference elicitation consists in a mathematical model able to express user preferences. In this paper, we consider the *conditional preference rules* (cp-rules) introduced in [Wilson 2004]. A cp-rule allows to inform the preference on the values of an attribute depending on

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

the values of some other attributes. For example in our movie database scenario, a user can specify his/her preference concerning the attribute *director* depending on the value of the attribute *gender*: For movies whose director is *Woody Allen* he/she prefers *comedy* to *suspense* and for movies from director *Steven Spielberg* he/she prefers *action* films to *drama*.

On both frameworks for expressing preferences (quantitative or qualitative), it is important to develop strategies to avoid the inconvenience for the user to report his/her preferences explicitly, a process that can be tedious and take a long time, causing the user not willing to provide such information. In this context, the development of preference mining techniques allowing the automatic inference of user preferences becomes very relevant.

Prior to the development of preference mining techniques we must be aware of the following questions: (1) *Whose preference we are interested in mining: a single user or a group of users ?*; (2) *How is formatted the data from which the user preferences will be mined* ?; (3) *What model will be used for expressing the preferences we are interested in mining ?*; (4) *What learning technique will be employed ?* In this work we are interested in mining the preferences of a *single user*. The data consist of a set of pair of tuples (t_1, t_2) provided by the user, meaning that he/she prefers t_1 to t_2. The method we propose intends to mining conditional preference expressed by a set of cp-rules [Wilson 2004]. And finally, the technique for preference mining we propose is based on the bayesian network technique used in classification tasks.

Like Bayesian Network classifiers, our miner technique *CPrefMiner* consists in the discovering of a graph and a set of tables of conditional probabilities expressing the user preferences on individual attributes. We call this pair of components a *preference network*. The method *CPrefMiner* receives as input a set of user preferences (pairs of tuples) and infers from that a set of conditional preference rules that will be used to infer new preferences about new attributes values. This paper focuses on the first discovering task, that is, we are interested firstly in proposing a tool for discovering the topology of the preference network. The mining technique we propose adapts the technique for learning bayesian network structures introduced in [Cooper and Dietterich 1992].

Main contributions. The main contributions of this paper can be summarized as follows: (1) We introduce the *preference networks* which are a graph formalism to express conditional preference rules with uncertainty; (2) We propose the technique *CPrefMiner* that, given a set of pair of tuples as input is able to infer the structure (topology) of the preference network; (3) We implement the technique and test it on synthetic data. The preliminary experimental results show that *CPrefMiner* is able to infer a preference network compatible with the input data distribution.

Organization. This paper is organized as follows. In Section 2 we briefly discuss some related work concerning Conditional Preference Modeling and Reasoning as well as Preference Mining. In Section 3 we formalize the problem of *conditional preference mining*, introducing all the necessary theorectical background. In Section 4 we present *CPrefMiner*, a greedy method for constructing a preference network from a set of pairs of tuples as input. Some preliminary empirical evaluation using synthetic datasets is reported in Section 5. In Section 6 we conclude the paper and discuss future work.

2. RELATED WORK

The research literature on preference reasoning and eliciting over objects is extensive. To the best of our knowledge there are no research studies involving Conditional Preference Mining techniques.

Conditional Preference Modeling and Reasoning. The approach of CP-Nets [Boutilier et al. 1999; Boutilier et al. 2004] uses a very simple graphical model which captures users qualitative conditional preference over tuples, under a *ceteris paribus* semantics. The approach of TCP-Nets [Brafman et al. 2006] generalizes the CP-Nets by introducing the ability of expressing absolute and relative importance of attributes. The approach introduced in [Wilson 2004] uses a logical framework for expressing conditional preference statements. It consists of a formalism in the same lines of CP-Nets but with a richer language allowing to express not only the usual CP-Nets statements but also TCP-Nets statements and more general conditional statements (called *stronger conditional statements*). In the present paper, we use the formalism of [Wilson 2004] to express cp-rules, but we restrict ourselves to the fragment of this formalism corresponding to the CP-Nets.

Preference Mining. In [Holland et al. 2003] the authors propose a technique for mining user preferences whose underlying model is the *pareto preference model*. In this model, preferences are not conditional, that is, preferences on values of attributes do not depend on the values of other attributes. Such preference rules are obtained from log data generated by the server when the user is accessing a site. Another approach to preference mining is presented in [Jiang et al. 2008]. 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 in the database. The underlying preference model is the *pareto preference model* as in [Holland et al. 2003]. In [Crammer and Singer 2001] and [Cohen et al. 1999] algorithms for mining quantitative preferences are proposed. In this work the main goal is to find automatically a prediction rule which assigns a score to each tuple of the database. The order is obtained by ranking the scores.

Use of Bayesian Networks as a Tool for Customization. In [Radde et al. 2008; Radde and Freitag 2010] an inference engine based on Bayesian Networks is proposed. It aims at inferring user preferences about values of technical attributes (such as memory size, broadband internet connectivity, etc, in a mobile communication domain). The preference elicitation is accomplished by asking the users some simple (not technical) questions about their needs and expectations. Their answers are entered as evidence into a Bayesian Network that models the relationships of user needs and technical properties of products. The Bayesian Network is built by an expert in the domain application. It is used to infer user preferences about technical attributes values. In our work, Bayesian Network represents the conditional preference rules and is built *automatically* from the preference sample provided by the user. The dependence between the attributes is discovered from the input and not provided by an expert. Moreover, the preference model underlying the work of [Radde et al. 2008; Radde and Freitag 2010] is the *pareto preference model* (like in [Holland et al. 2003; Jiang et al. 2008]), differently from our approach which uses the conditional preference model.

3. PROBLEM FORMALIZATION

In this section we introduce the main concepts related to the problem of mining conditional preferences and also the concepts necessary to understand the *CPrefMiner* method.

The main goal of a preference mining method is the ability to provide a *preference relation* over a given dataset. A *preference relation* on a finite set of objects $A = \{a_1, a_2, ..., a_n\}$ is a strict partial order over A, that is a binary relation $R \subseteq A \times A$ satisfying the irreflexivity and transitivity properties. Typically, a strict partial order is represented by the symbol $<$. So if $<$ is a preference relation, we denote by $a_1 < a_2$ the fact that a_2 is preferred to a_1.

Definition 3.1 *Preference Sample.* Let $R(A_1, A_2, ..., A_n)$ be a relational schema. Let $\mathrm{Tup}(R)$ be the set of all tuples over R. A *preference sample* over $\mathcal{H}$ is a finite set $\mathcal{H} \subset \mathrm{Tup}(R) \times \mathrm{Tup}(R)$ which is *consistent*, that is, if $(u, v) \in \mathcal{H}$ then $(v, u) \notin \mathcal{H}$. Intuitively, the pair (u, v) represents the fact that the user prefers *the tuple u to the tuple v*.

Example 3.1 Preference Sample. Let $R(A, B, C, D)$ be a relational schema with attribute domains given by $\mathbf{dom}(A) = \{a_1, a_2, a_3\}$, $\mathbf{dom}(B) = \{b_1, b_2\}$, $\mathbf{dom}(C) = \{c_1, c_2\}$ and $\mathbf{dom}(D) = \{d_1, d_2\}$. Let I be an instance over R as shown in Figure 1. Figure 2 illustrates a preference sample over R, comparing some tuples of I.

Id	A	B	C	D
1	a_1	b_1	c_1	d_1
2	a_1	b_1	c_1	d_2
3	a_2	b_1	c_1	d_2
4	a_1	b_2	c_1	d_2
5	a_2	b_1	c_2	d_1
6	a_3	b_1	c_1	d_1

Fig. 1. An instance over R

Id	A	B	C	D	A	B	C	D
1	a_1	b_1	c_1	d_1	a_1	b_1	c_1	d_2
2	a_1	b_1	c_1	d_1	a_2	b_1	c_1	d_2
3	a_1	b_2	c_1	d_2	a2	b_1	c_2	d_1
4	a_2	b_1	c_2	d_1	a_1	b_1	c_1	d_1
5	a_2	b_1	c_2	d_1	a_3	b_1	c_1	d_1
6	a_2	b_1	c_1	d_2	a_2	b_1	c_2	d_1
7	a_1	b_2	c_1	d_2	a_1	b_1	c_1	d_1

Fig. 2. A set of preference samples

Definition 3.2 *Preference Network.* A *preference network* over a relational schema $R(A_1, ..., A_n)$ is a structure $(B_S, \Im)$ where:

(1) B_S is a directed acyclic graph whose nodes are attributes in $\{A_1, ..., A_n\}$ and the edges stands for attribute dependency.

(2) $\Im$ is a mapping that associates to each node of B_S a conditional probability table of preferences. A conditional probability table of preferences is a finite set of conditional probabilities of the form $P[E_2|E_1]$ where (1) E_1 is an event of type $(A_{i_1} = a_{i_1}) \wedge \ldots \wedge (A_{i_k} = a_{i_k})$ such that $\forall j \in \{1, ..., k\}$, $a_{i_j} \in \mathbf{dom}(A_{i_j})$, and (2) E_2 is an event of type "$(B = b_1)$ is preferred to $(B = b_2)$", where B is an attribute of R, $B \neq A_{i_j}$ $\forall j \in \{1, ..., k\}$ and $b_1, b_2 \in \mathbf{dom}(B)$.

Example 3.2 Preference Network. Let $R(A, B, C, D)$ be the relational schema of Example 3.1. Figure 3 illustrates a preference network PrefNet$_1$ over R.

Each conditional probability in the probability table associated to a node X in the graph B_S represents a degree of belief of preferring some values for X to other ones, depending on the values assumed by its parents in the graph. For instance $P[D = d_1 > D = d_2|C = c_1] = 0,6$ means that the probability of $D = d_1$ be preferred to $D = d_2$ is 60% given that $C = c_1$ and the probability of $D = d_2$ be preferred to $D = d_1$ is 40% under the same condition $C = c_1$.

In classification tasks, Bayesian Networks are used as a tool to classify tuples. In our preference mining scenario, Preference Networks will be used to compare pairs of tuples. For instance the preference network PrefNet$_1$ depicted in Figure 3 allows to infer a preference ordering on tuples over $R(A, B, C, D)$. According to this ordering, tuple $t_1 = (a_2, b_2, c_1, d_1)$ is preferred to tuple $t_2 = (a_2, b_2, c_1, d_2)$. Indeed, these tuples differ only on attribute D and the conditional probability $P[d_1 > d_2|C = c_1] = 0,6$ in the probability table of attribute D allows to conclude that $d_1 > d_2$. Thus t_1 is

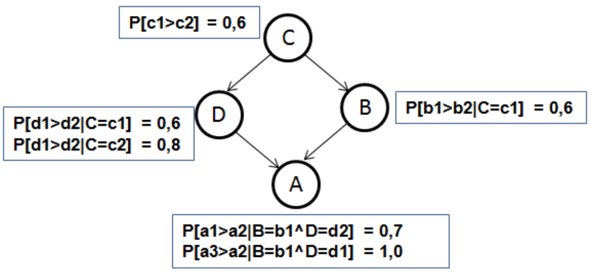

Fig. 3. Preference Network PrefNet$_1$

preferred to t_2. Note that this information may not be originally provided the user, but just inferred by the Preference Network.

The quality of a preference network as an ordering tool is measured by means of its *accuracy*. In order to define the *accuracy* of a preference network, we would need a rigorous definition of the *strict partial order* inferred by the preference network. For lack of space, we do not provide this rigorous definition here. Nonetheless, as it will be clear in the end of this section, this definition will not be needed in the remaining sections.

Definition 3.3 *Accuracy of a preference network*. Let **PNet** be a preference network over a relational schema R. Let $\mathcal{H}$ be a preference sample over R. The *accuracy* of **PNet** with respect to $\mathcal{H}$ is defined by Acc(**PNet**,$\mathcal{H}$)= $\frac{N}{M}$, where M is the cardinality of $\mathcal{H}$ and N is total amount of pairs of tuples $(t_1, t_2) \in \mathcal{H}$ compatible to the preference ordering inferred by **PNet** on the tuples t_1 and t_2. That is, the accuracy of **PNet** with respect to $\mathcal{H}$ is the percentage of *bituples* of $\mathcal{H}$ which are correctly ordered by **PNet**.

Our Preference Mining Problem is formalized as follows:

Input: A relational schema $R(A_1, ..., A_n)$, a training preference sample T_1 over R and a testing preference sample T_2 of R.

Output: A *preference network* $(B_S, \Im)$ over the relational schema $R(A_1, ..., A_n)$ with a good accuracy with respect to T_2.

The solution of this problem consists in: (1) producing the structure B_S which reflects the interdependency among the attributes as far as preferences are concerned and (2) producing the mapping $\Im$ which reflects how some attributes influence the preference over the values of other attributes. In this paper, we tackle just the subtask (1). So, we will not be able to measure the accuracy of the output, since the probability tables are needed in order to calculate the accuracy of a network. We will need another measure to evaluate the quality of the structures returned by our method. In the next section we present the algorithm *PrefK2* to solve the subtask (1) as well as a suitable measure to evaluate its output.

4. MINING THE PREFERENCE NETWORK STRUCTURE

The part of the method *CPrefMiner* dedicated to discover the preference network structure is based on the algorithm *K2* of [Cooper and Dietterich 1992] designed to learn classic bayesian networks topology. The algorithm *K2* takes as input a set of classified tuples and returns a topology (graph) that has the highest probability of reflecting the distribution of the input data. Analogously, the algorithm *PrefK2* we propose takes as input a set of preference samples $\mathcal{H}$ and returns a structure B_S that best reflects the distribution of $\mathcal{H}$. *PrefK2* is a greedy algorithm that basically consists in generating a collection of structures and associating a *score*$(B_S, \mathcal{H})$ to each structure B_S generated accordingly to a heuristic. This score[1] measures how the structure B_S is compatible with the probability distribution of the input preference sample $\mathcal{H}$. Before defining the measure *Score*$(B_S, \mathcal{H})$ (Definition 4.2) we need to introduce some previous notation.

Definition 4.1 *The natural partition*. Let A be an attribute of the relational schema $R(A_1, ..., A_n)$ and $\mathcal{H}$ a preference sample over R. Let dom$(A, \mathcal{H})$ be the projection of $\mathcal{H}$ on the attribute A. We denote by *Comp*(A) the set of pairs $(a_1, a_2) \in$ dom$(A, \mathcal{H})$ satisfying the following: (1) $a_1 \neq a_2$ and (2) there exists $(t_1, t_2) \in \mathcal{H}$ such that $t_1[A] = a_1$ and $t_2[A] = a_2$ or $t_1[A] = a_2$ and $t_2[A] = a_1$.

Let *Comp*$(A) = \{p_1, ..., p_m\}$. For each $l \in \{1, ..., m\}$ let $P_l = \{(t_1, t_2) \in \mathcal{H} \mid (t_1[A], t_2[A]) = p_l\}$. The set $\{P_1, ..., P_m\}$ defined in this way is called the *natural partition* of the preference sample $\mathcal{H}$ with respect to the attribute A.

[1] adapted from the function $P(B_S|D)$ used in [Cooper and Dietterich 1992] which corresponds to the probability of the network B_S reflecting the distribution of data in the database D of tuples.

Example 4.1. Let us consider the preference sample illustrated in Figure 2. In this example we have $Comp(A) = \{(a1, a2), (a2, a3)\}$. Figures 4 and 5 illustrate the natural partitions with respect to attributes A and B respectively.

	Id	A	B	C	D		A	B	C	D
	2	a_1	b_1	c_1	d_1		a_2	b_1	c_1	d_2
(a_1,a_2)	3	a_1	b_2	c_1	d_2		a2	b_1	c_2	d_1
	4	a_2	b_1	c_2	d_1		a_1	b_1	c_1	d_1

	Id	A	B	C	D		A	B	C	D
(a_2,a_3)	5	a_2	b_1	c_2	d_1		a_3	b_1	c_1	d_1

Fig. 4. Natural Partition of Comp(A)

	Id	A	B	C	D		A	B	C	D
(b_1,b_2)	3	a_1	b_2	c_1	d_2		a2	b_1	c_2	d_1
	7	a_1	b_2	c_1	d_2		a_1	b_1	c_1	d_1

Fig. 5. Natural Partition of Comp(B)

Definition 4.2 $Score(B_S, \mathcal{H})$. Let $R = \{A_1, ..., A_n\}$ be a relational schema and $\mathcal{H}$ a preference sample over R with cardinality k. Let B_S be a preference network structure over R. For each attribute A_i of B_S we denote by π_i the set of its parents in B_S. Let $Comp(A_i) = \{p_1, ..., p_m\}$. Let us consider the natural partition $P_1 \cup ... \cup P_m$ of $Comp(A_i)$ (as described in Definition 4.1). For each subset P_l, let us consider all instantiations w^j_{il} of the parent attributes $\pi_i = \{B_1, ..., B_p\}$ of A_i corresponding to P_l and satisfying the following condition:
(*) for each $(t_1, t_2) \in w^j_{il}$ we have $t_1[B_1] = t_2[B_1]$,..., $t_1[B_p] = t_2[B_p]$. We denote by q^l_i the total amount of such instantiations (that is, $j \in \{1, ..., q^l_i\}$).

Let N_{ijl1} be the number of *bituples* (t_1, t_2) in P_l such that $(t_1[A], t_2[A]) = p_l$ and such that the attributes in π_i are instantiated as w^l_{ij}. Similarly, let N_{ijl0} be the number of *bituples* (t_1, t_2) in P_l such that $(t_2[A], t_1[A]) = p_l$ and such that the attributes in π_i are instantiated as w^l_{ij}. The function $Score(B_S, \mathcal{H})$ is defined by the equation 1.

$$\text{Score}(B_S, \mathcal{H}) = \prod_{i=1}^{n} \sum_{l=1}^{m} \frac{1}{m} \prod_{j=1}^{q^l_i} \frac{N_{ijl1}! N_{ijl0}!}{(N_{ijl1} + N_{ijl0})!}. \tag{1}$$

The formula (1) has been adapted from the formula given in [Cooper and Dietterich 1992] for evaluating $P(B_S|D)$, the probability of the bayesisan network B_S reflecting the distribution of the database D of **tuples**. In our case, Score(B_S,$\mathcal{H}$) is related to the probability of the preference network B_S reflecting the preference sample $\mathcal{H}$.

Example 4.2. Let us consider the preference network structure B_S corresponding to the preference network PrefNet$_1$ of Figure 3 and the preference sample $\mathcal{H}$ illustrated in Figure 2. The attributes of B_S are $\{A, B, C, D\}$. For $i = 1$ (corresponding to the first attribute A) we have $\pi_1 = \{B, D\}$ and $Comp(A) = \{p_1 = (a_1, a_2), p_2 = (a_3, a_2)\}$. For $l = 1$, P_1 is the set of *bituples* corresponding to identifiers 2, 3 and 4 in Figure4. We have only one instantiation for B,D satisfying condition **(*)** namely $B = b1, D = d1$, corresponding to the *bituple* 4. For this unique *bituple* (t_1, t_2) we have $t_1[A] = a_2$ and $t_2[A] = a_1$. So, $N_{1110} = 1$ and $N_{1111} = 0$

Applying equation 1 to evaluate the score of B_S1 with respect to the preference sample $\mathcal{H}$:

$$mboxScore(B_S, \mathcal{H}) = \frac{\frac{2!2!}{(2+2)!}}{1} \frac{\frac{2!1!}{(2+1)!}}{1} \frac{\frac{1!0!}{(1+0)!}}{1} \frac{\frac{1!0!}{(1+0)!} \frac{1!0!}{(1+0)!}}{2} = 0.0277 \tag{2}$$

The task of learning the preference network structure is a nontrivial task since the size of the search space of the candidate structures is exponential on the number of the attributes. *PrefK2* is a greedy algorithm that restricts the search space by assuming an ordering among attributes. Our method consists in applying the algorithm several times with different orders on the attributes and then choosing the best structure, that with the higher score. The algorithm *PrefK2* searches for each attribute A_i a set of parents that maximizes the function $g(i, \pi_i)$ below.

$$g(i,\pi_i) = \sum_{l=1}^{m} \frac{1}{m} \prod_{j=1}^{q_i^l} \frac{N_{ij1}!N_{ij0}!}{(N_{ij1}+N_{ij0})!}. \tag{3}$$

Figure 6 describes the algorithm *PrefK2* for building the preference network structure.

Input:A set n of attributes $\mathbf{A} = \{A_1, ..., A_n\}$, an order $\mathbf{A}$, a preference sample $\mathcal{H}$ over the relational schema $R(A_1, ..., A_n)$.
Output: A graph B_S with nodes in $\{A_1, ..., A_n\}$.
1. **For** $i = 1$ **to** n **do**
2. **Build** the natural partition $\mathrm{P} = \{P_1, ..., P_m\}$ for each $Comp(A_i)$
3. $\Pi_i := \emptyset$; //stores the best parents for A_i
5. **For each** $P_j \in \mathrm{P}$
6. $\pi_i := \emptyset$;//stores the best parents for A_i with respect to each $P_j \in \mathrm{P}$
7. $Score_{old} := g(i, \pi_i)$;
8. $Proceder :=$ **TRUE**;
9. **While** $Proceder$ **do**
10. Given Z the set of nodes in Predecessors(A_i) $-\pi_i$ that maximizes $g(i, \pi_i \cup \{Z\})$;
11. $Score_{new} := g(i, \pi_i \cup \{Z\})$;
12. **If** $Score_{new} > Score_{old}$ **then**
13. $Score_{old} := Score_{new}$;
14. $\pi_i := \pi_i \cup \{Z\}$;
15. **Else** $Proceder :=$ **FALSE**
16. $\Pi_i := \Pi_i \cup \pi_i$
17. **White** ('Node:',A_i,'Parents of this Node', π_i)

Fig. 6. Algorithm *PrefK2*

5. EXPERIMENTAL EVALUATION

The experiments have been limited to the validation of the topology of the preference network. We developed a synthetic dataset generator that creates a set of structures with random probability tables and, for each structure S, creates a preference sample $\mathcal{H}_S$ statistically consistent with S. Such a method is based on the *Probabilistic Logic Sample* methodology [Henrion 1988]. The quality of the preference network **PNet** generated by the algorithm *PrefK2* on the synthetic preference sample $\mathcal{H}_S$ is evaluated by comparing the values of $Score(S, \mathcal{H}_S)$ and $Score(\mathbf{PNet}, \mathcal{H}_S)$. Figure 7 illustrates the validation process.

The experiments have been conducted by varying the following parameters: (1) the number of nodes (8, 10, 15 and 30 attributes), (2) the size of the preference samples (100, 200, 300, 1.000, 2.000, 3.000 and 10.000),

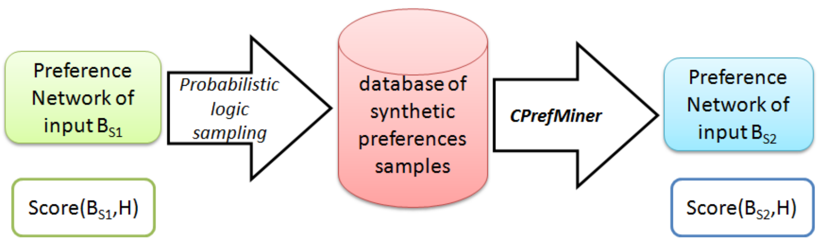

Fig. 7. Validation

(3) the number of rules in the probability tables associated with each node (8, 10, 15, 30 and 50) and the orders of the attributes. The tests showed that the preference networks produced by our method *CPrefMiner* is fully compatible with the distribution of the training preference sample.

6. CONCLUSION AND FURTHER WORK

Nowadays conditional preferences are aubiquitous. They can be very relevant in situations where user preferences over the values of a particular attribute depend on his/her context ([Holland and Kießling 2004]). In this paper we introduced the formalism of the *preference network* for expressing the user preferences. We proposed the algorithm *PrefK2* for discovering the structure of a preference network. At the best of our knowledge, there is no technique for solving the problem of mining conditional preferences in the litterature. A lot of work remains to be done in order to make *CPrefMiner* a tool for preference mining. Presently, we are working on the second task of *CPrefMiner*, that is, the discovery of the preference network probability tables. This first proposal opens a wide spectrum for future research. We intend to improve the *Score* function in order to obtain more refined network topologies. We also intend to propose another method for learning the network topology that takes into account the complexity of the network and where there is no need for a prior ordering of the attributes. Finally, we will test the performance of *CPrefMiner* on real datasets by evaluating the accuracy of the results. For that, we will use the benchmark for personalized consultations available at `http://apmd.prism.uvsq.fr/SubProject4/TestPlatform/IntegratedDB.html`. This benchmark is a relational database that integrates data for movie recommendation from the sites *MovieLens* (`http://www.movielens.org`) and IMDB (`http://www.imdb.com/`). This benchmark contain data about 6040 users, 3881 films and the evaluations they gave to these films. We also are investigating other methods for mining conditional preferences inspired on techniques for pattern discovery.

REFERENCES

Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., and Poole, D. CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. *Journal of Artificial Intelligence Reasearch* 21 (1): 135–191, January, 2004.

Boutilier, C., Brafman, R. I., Hoos, H. H., and Poole, D. Reasoning with conditional ceteris paribus preference statements. In *Proceedings of Annual Conference on Uncertainty in Artificial Intelligence*. Stockholm, Sweden, pp. 71–80, 1999.

Brafman, R. I., Domshlak, C., and Shimony, S. E. On graphical modeling of preference and importance. *Journal of Artificial Intelligence Research* 25 (1): 389–424, January, 2006.

Cohen, W. W., Schapire, R. E., and Singer, Y. Learning to order things. *Journal of Artificial Intelligence Research* vol. 10, pp. 243–270, 1999.

Cooper, G. F. and Dietterich, T. A bayesian method for the induction of probabilistic networks from data. In *Machine Learning*. pp. 309–347, 1992.

Crammer, K. and Singer, Y. Pranking with ranking. In *Advances in Neural Information Processing Systems*. MIT Press, pp. 641–647, 2001.

Henrion, M. Propagating uncertainty in bayesian networks by probabilistic logic sampling. *Uncertainty in Artificial Intelligence* vol. 2, pp. 149–163, 1988.

Holland, S., Ester, M., and Kiessling, W. Preference mining: A novel approach on mining user preferences for personalized applications. In *Proceedings on Principles and Practice of Knowledge Discovery in Databases*. Cavtat-Dubrovnik, Croatia, pp. 204–216, 2003.

Holland, S. and Kiessling, W. Situated preferences and preference repositories for personalized database applications. In *Proceedings of International Conference on Conceptual Modeling*. Shanghai, China, pp. 511–523, 2004.

Jiang, B., Pei, J., Lin, X., Cheung, D. W., and Han, J. Mining preferences from superior and inferior examples. In *Proceeding of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, NV, USA, pp. 390–398, 2008.

Radde, S. and Freitag, B. Using bayesian networks to infer product rankings from user needs. In *Workshop on Intelligent Techniques for Web Personalisation and Recommender Systems*, 2010.

Radde, S., Kaiser, A., and Freitag, B. A model-based customer inference engine. In *Proceedings of European Conference on Artifical Intelligence - Workshop on Recommender Systems*. Patras, Greece, pp. 2–7, 2008.

Wilson, N. Extending cp-nets with stronger conditional preference statements. In *Proceedings of AAAI Conference on Artificial Intelligence*. San Jose, CA, USA, pp. 735–741, 2004.