Service Selection based on Similarity Measurement for Conditional Qualitative Preference

Hongbing Wang, Jie Zhang*, Hualan Wang, Yangyu Tang, and Guibing Guo*
School of Computer Science and Engineering, Southeast University, China
*School of Computer Engineering, Nanyang Technological University, Singapore
{hbw@seu.edu.cn}

***Abstract*—Similarity measurement is essential in many preference-based personalized applications such as collaborative recommendation and service selection. Up to date, current researches have mainly focused on the measures for quantitative preference rather than for qualitative preference, although the latter has attracted much attention recently. Only a very few methods to measure user similarity are proposed. This paper aims to fill in this gap by proposing an intuitive similarity measure for conditional qualitative preference which is represented by CP-nets. Experimental results based on two expanded real-life datasets demonstrate that our similarity measure is not only able to correctly reflect user's preference changes, but also effective to identify similar users.**

***Keywords*-Service Selection, Similarity Measure, Conditional Qualitative Preference, CP-nets**

I. Introduction

Similarity measure is essential in many preference-based personalized applications. For example, in collaborative recommendation, the rating of an unknown product or service (item) for an active user can be predicted based on the ratings of like-minded (similar) users [1]. Those users are usually selected if their similarity with the active user is greater than a predetermined threshold. In addition, user similarity is also used to weigh their ratings when predicting the ratings for the active user. Therefore, similarity measure has strong influence on the success of personalized recommender systems. Other than the implicitly indicated preferences (ratings), similarity measure is also claimed to be useful in the application of service selection based on explicitly stated preferences [2].

Furthermore, similarity measure is also dependent on the means by which user preference is represented. In general, user preference could be represented in a quantitative or qualitative way. Quantitative preference can be obtained through two ways. The first is to allow users to explicitly report their opinions by indicating the extent to which they prefer the services (e.g. "I prefer Thai Airline at the level of 0.7"). Preference rating is the most commonly used technique. The other is to implicitly track and transform users' behaviors (e.g. click-through, purchase, browse) into preference scales. For instance, 1 indicates a product or service is purchased by some user whereas 0 indicates no purchase happened. Ratings by users themselves are generally more accurate than those transformed from users' behaviors. However, ratings issued may not be available or suitable for some applications in which case similarity computation has to rest on implicit transformations.

In contrast, qualitative preference allows users to express their preferences more directly and intuitively. Specifically, users can describe services they would prefer by comparison (e.g. "I prefer Qantas Airline to Thai Airline") and even go with conditions [3] (e.g. "If time is late, I prefer Qantas Airline to Thai Airline"). It has attracted much attention recently partially due to the fact that users feel more comfortable and natural to express their preferences in such a manner. Conditional preference-networks (CP-nets) [3] is well-known as a powerful tool to represent and reason with conditional qualitative preferences under the *ceteris paribus* ("all else being equal") semantics. Many studies [2], [4], [5] adopt CP-nets to allow users to express their preferences explicitly. However, to date there has been little work devoting to measuring user similarity based on conditional qualitative preference represented by CP-nets. The work of Wang et al. [2] is possibly the first and the only one in addressing this issue with the underlying assumption that users share the same structure of CP-nets. However, this assumption is not realistic since in real life users could have totally different structures of CP-nets.

Therefore, this paper proposes a novel similarity measure in order to fill in the gap between qualitative preference explicitly expressed by users and similarity computation required for the purpose of preference modeling. More specifically, user preference is represented by CP-nets towards a certain service that contains a number of features (attributes). Unlike [2], the structures of CP-nets for two users can vary in the orders of attributes and the number of specified attributes. User similarity is defined as the ratio of the number of comparable combinations of attribute values over the number of all combinations of attribute values. The correctness and effectiveness of our method are demonstrated by experiments based on two expanded real-life datasets. The time complexity of our method is also analyzed.

The rest of this paper is organized as follows. Section II reviews the similarity methods proposed in the literature. Section III introduces the background knowledge about CP-nets and related concepts. After that, Section IV describes in detail our method to compute user similarity and experiments are conducted in Section V to verify the correctness and effectiveness of our method. Finally, Our work is concluded in Section VI.

II. Related Work

There have been many methods proposed in the literature to measure user or item similarity. According to the different kinds of inputs, the similarity measures can be classified into four categories. 1) Rating-based approaches. In recommender systems, users usually express their opinions for an item by giving a quality rating. The vector of all item ratings reported by the users represents their preferences. Vector-based or correlation-based approaches are used to compute user or item similarity. Typical examples are cosine similarity and Pearson correlation coefficient [1]. Other approaches include Spearman rank correlation coefficient [6], concordance similarity [7], etc. 2) Text-based approaches. For information retrieval systems, document similarity is often measured based on the frequencies of the terms that occur in the documents. Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) [1] are the widely used approaches. 3) Link-based approaches. The in-links and out-links of items (e.g. web pages, articles) are utilized to model the global importance of each item iteratively. PageRank [8], HITS [8] and SimRank [9] are typical examples of this kind. 4) Graph-based approaches. This kind of methods stems from graph theory, making use of the structure of items or users along with other items or users. Maximum Flow and Minimum Cut [10] are representative. However, all of these methods are computed based on quantitative preference terms (e.g. ratings, text frequencies, links, weighted structures) and none of them can be adopted to compute user similarity when user preference is explicitly and qualitatively expressed. In this paper, we only focus on the approaches to compute user similarity rather than item similarity.

To the authors' best knowledge, there has been only one work conducted by Wang et al. [2] so far to measure user similarity based on conditional qualitative preference. Specifically, users' preferences are represented by CP-nets but some preferences on certain attributes may not be specified by users and hence missing. The authors attempt to use the collaborative filtering technique to predict users' missing preferences based on the preferences of other similar users. User similarity is computed as the ratio of overlapping preferences over the whole preferences. The underlying assumption behind their approach is that different users share the same structure of CP-nets towards the same kind of services. Although they justify that to some extent this assumption is reasonable, it is possible in real life that different users may have totally different structures of CP-nets even for the same kind of services.

Our work is an extension to that of Wang et al. [2], aiming to solve similarity computation in the case where the assumption of their approach is not applicable. Accordingly, the computation of similarity is no longer limited to the structure of CP-nets but directly rests on the preferences of two users. Basically, we define user similarity as the ratio of the number of comparable combinations of attribute values over all the possible combinations of attribute values.

III. Background

Conditional preference-networks (CP-nets) [3] is a graphical and effective formalism for representing and reasoning with conditional qualitative preference in a compact, intuitive and structured manner, under the *ceteris paribus* ("all else being equal") semantics. It consists of two parts, namely directed dependence graph (DDG) and conditional preference tables (CPTs). DDG contains a set of attributes $V = \{X_1, ...X_n\}$ represented as nodes, where each node X_i is associated with a finite domain $D(X_i) = \{x_{i1}, \ldots, x_{in}\}$. A child node X_i is dependent on a set of direct parent nodes $P(X_i)$. They are connected by arcs from $P(X_i)$ to X_i in the graph. Under the semantics of ceteris paribus, the value of X_i is only dependent on the values of $P(X_i)$. Each attribute X_i could be regarded as a feature of real services.

In addition, each node X_i is annotated with a CPT denoted by $CPT(X_i)$, which accommodates users' explicit preferences over all the attribute values of X_i. A preference between two attribute values x_{i1} and x_{i2} can be specified by the relation $\succ$ given the conditions of the values of $P(X_i)$. For example, the preference $x_{11} : x_{21} \succ x_{22}$ indicates that attribute value x_{21} is preferred to another value x_{22} for attribute X_2 if its parent node X_1 has the value x_{11}.

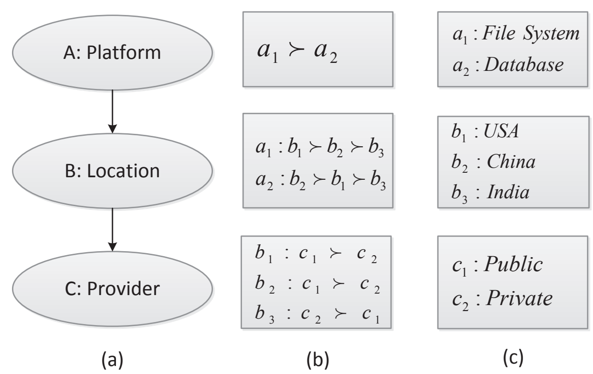


Fig. 1. (a, b) The CP-nets of User u; (c) Attribute Values.

A typical CP-nets is illustrated in Figure 1. It describes the preferences of a company (user) u regarding the data storage and access service which consists of three attributes with respect to the non-functional quality of service (QoS), namely A : Platform, B : Location and C : Provider. In Figure 1, (a) shows the DDG of three attributes, (b) the detailed CPTs over all attribute values and (c) the specific semantics for each attribute value. Specifically, data can be stored in either a file system a_1 or a database a_2 which can be located in USA b_1, China b_2 or India b_3 and can be accessed publicly c_1 or privately c_2. User u has an unconditional preference on Platform that a file system is always preferred to a database. But for the preference of the others, it depends on the choices of previous attributes. For example, if file system a_1 is chosen for data storage, then the location in USA b_1 is preferred to China b_2 which is superior to India b_3. In that case, user u prefers data to be accessed publicly rather than privately. Note that this example will be used throughout the rest of this paper.

A service pattern SP is a combination of values of all attributes of an abstract service. In this example, $a_1b_1c_1$ and $a_1b_2c_1$ are two service patterns, denoted by SP_1 and SP_2 respectively. According to the preferences on attribute B, it is known that attribute values b_1 is preferred to b_2 given that the value of parent node A is a_1. Consequently, it can be concluded that service patterns SP_1 is preferred to SP_2, or SP_1 dominates SP_2. The dominance relationship of two service patterns is defined as a *pre-order* between them, denoted by $< SP_1, SP_2 >$.

IV. Proposed Similarity Measure

In this section, we will first introduce the representation of conditional preference based on partial order[1]. Our method to compute user similarity is then described in detail as well as the algorithm in pseudo codes. Finally, an intuitive example is given to exemplify the procedure of our method to compute user similarity.

A. Conditional Preference

Due to relaxing the assumption made in [2], the structures of two CP-nets could be different either in preference conditions or preference statements. However, part of them could still be common and the generated pre-orders between service patterns could also be the same. Therefore, our work is built on the partial order between two attribute values of a certain attribute. Partial order defines the dominance relationship between two attribute values, given the condition of a certain preference. It is a basic component for our definition of user similarity.

By definition, attribute X_i is associated with a finite domain $D(X_i) = \{x_{i_1}, \ldots, x_{i_n}\}$. Thus for each conditional preference p on attribute X_i, its statement can be decomposed into $\binom{|D(X_i)|}{2}$ partial orders, i.e. the number of combinations of any two attribute values of X_i. According to the ceteris paribus semantics, there are $\prod_{X_j \neq X_i \cap X_j \notin P(X_i)} |D(X_j)|$ pre-orders deduced by each partial order. Note that if $X_j \in \varnothing$, we define $\prod_{X_j \in \varnothing} |D(X_j)| = 1$, i.e. when the partial order covers the values of all attributes (including condition and statement), there is only one pre-order available.

We use matrix notation to denote the preference relationships between attribute values (statements) of partial orders after decomposition. For a conditional preference p on attribute X_i, the statements of decomposed partial orders can be represented by matrix M_p.

$$M_p = \begin{array}{c c} & \begin{array}{cccc} x_{i_1} & x_{i_2} & \cdots & x_{i_n} \end{array} \\ \begin{array}{c} x_{i_1} \\ x_{i_2} \\ \vdots \\ x_{i_n} \end{array} & \begin{pmatrix} e_{11} & e_{12} & \cdots & x_{1n} \\ e_{21} & e_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ e_{n1} & e_{n2} & \cdots & x_{nn} \end{pmatrix} \end{array}$$

[1]http://mathworld.wolfram.com/PartialOrder.html

where each entry e_{kj} refers to the preference relationship between attribute values x_{i_k} and x_{i_j}, and

$$e_{kj} = \begin{cases} 1 & \text{if } x_{i_k} \succ x_{i_j}; \\ -1 & \text{if } x_{i_j} \succ x_{i_k}; \\ 0 & \text{otherwise.} \end{cases}$$

and $|M_p|$ is used to denote the number of positive entries.

Take the preference $p = a_1 : b_1 \succ b_2 \succ b_3$ on attribute B in Figure 1 as an example. It can be decomposed into $\binom{3}{2} = 3$ partial orders, i.e. $p_1 = a_1 : b_1 \succ b_2$, $p_2 = a_1 : b_1 \succ b_3$ and $p_3 = a_1 : b_2 \succ b_3$. For each partial order, say p_1, it can deduce $\prod_{X_j \neq B \cap X_j \notin P(B)} |D(X_j)| = |D(C)| = 2$ pre-orders, i.e. $< a_1b_1c_1, a_1b_2c_1 >$ and $< a_1b_1c_2, a_1b_2c_2 >$. The statements of partial orders decomposed from preference p can be represented by

$$M_p = \begin{array}{c c} & \begin{array}{ccc} b_1 & b_2 & b_3 \end{array} \\ \begin{array}{c} b_1 \\ b_2 \\ b_3 \end{array} & \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix} \end{array}$$

and thus $|M_p| = 3$.

B. Similarity Measurement

User similarity is defined as the ratio of the number of common pre-orders over the number of all available pre-orders existing in the light of conditional preferences described in the two users' CPTs. In other words, the similarity of two users refers to the ratio of common dominance relationships over the whole dominance relationships deduced from CP-nets.

Definition 1 (Similarity): Let U and V be two CP-nets of an abstract service for two users u and v, $G(U)$ and $G(V)$ be the CPTs of U and V, respectively. Let p be a conditional preference of a certain attribute X_i and $N(p)$ denote the number of pre-orders between service patterns deduced by preference p under the semantics of ceteris paribus. The similarity between users u and v can be calculated by

$$Sim(u, v) = \frac{\sum_{p \in G(U) \wedge G(V)} N_c(p)}{\sum_{p \in G(U) \vee G(V)} N_a(p)} \tag{1}$$

where $N_c(p)$ and $N_a(p)$ represent the number of common pre-orders and the number of all possible pre-orders derived for users u and v, respectively.

1) Compute $N(p)$: Generally, as described in Section IV-A, each conditional preference can generate $N_p = \binom{|D(X_i)|}{2}$ partial orders, hence the number of pre-orders deduced by p can be computed as

$$N(p) = N_p \cdot \prod_{X_j \neq X_i \cap X_j \notin P(X_i)} |D(X_j)| \tag{2}$$

Let p_u and p_v be two conditional preferences of users u and v on attribute X_i, and M_{p_u} and M_{p_v} be the corresponding preference matrix of the decomposed partial orders, respectively. Then the number of common partial orders for $N_c(p)$ can be computed as $N_{cp} = |M_{p_u} + M_{p_v}|$. And the number of all partial orders for $N_a(p)$ can be computed as $N_{ap} = |M_{p_u}| + |M_{p_v}| - |M_{p_u} + M_{p_v}|$.

2) Determination of p: The critical point of Equation 1 is to compare the preferences in $G(U)$ and $G(V)$ in order to find out the common preferences. In this paper, we determine whether two conditional preferences are comparable based on the following definition.

Definition 2 (Comparable Preferences): Let $p_u = c_u : s_u$ and $p_v = c_v : s_v$ be two conditional preferences on a certain attribute for two users u and v, respectively, where c_u and c_v are the preference conditions, and s_u and s_v the preference statements. The two preferences are comparable iff their preference conditions are containable, i.e. $\{c_u\} \subseteq \{c_v\}$ or $\{c_v\} \subseteq \{c_u\}$. Otherwise, they are incomparable.

More specifically, if the preference conditions are the same, i.e. $\{c_u\} = \{c_v\}$, it is trivial to judge that they are comparable. This also holds for the case where $\{c_u\} = \{c_v\} = \varnothing$. If the preference conditions are containable (excluding self-included), i.e. $\{c_u\} \subset \{c_v\}$ or $\{c_v\} \subset \{c_u\}$, then the judgement of two preferences is subject to Theorem 1, which further adjusts the conditions to be the same and hence they are comparable.

Theorem 1: Let $C_{v-u} = \{c_v\} - \{c_u\}$ be the set of distinct condition values for two comparable preferences p_u and p_v on attribute X_j of users u and v. For each condition value $c_{v_i} \in C_{v-u}$ belonging to parent attribute X_i, let C_c denote the set of combinations of values of all such attributes. Then CPT(X_j) of user u can be adjusted by transforming each preference $p_u = c_u : s_u$ to multiple preferences $p_u^i = c_u \vee c_{v_i} : s_u$, where $c_{v_i} \in C_c$. The newly generated CPT$'(X_j)$ does not change the dominance relationships described in the original CP-nets.

Proof: When $|C_{v-u}| = 1$, there is only one condition value c_{v_i} of preference p_v different from preference p_u on attribute X_j. Assume that c_{v_i} is a value of parent attribute X_i, C_c will only contain the values of attribute X_i, i.e. $D(X_i)$. For each preference $p_u = c_u : s_u$ in CPT(X_j), it will be transformed into $|D(X_i)|$ new preferences $p_u^{1i} = c_u \vee c_{v_i} : s_u = c^1 : s_u$, where $c_{v_i} \in C_c$. For preference statement s_u, it is dependent on $c_u \vee c_{v_i}$, since $c_{v_i} \in C_c = D(X_i)$ and every possible c_{v_i} is extended in p. We can conclude that preference statement s_u is independent of c_{v_i} but only dependent on c_u. Hence after adjustment, the original dominance relationships are not changed.

When $|C_{v-u}| = k$, there are k distinct condition values. The adjustment procedure will transform preference $p_u = c_u : s_u$ to multiple preferences $p_u^{ki} = c_u \vee c_{v_i} : s_u = c^k : s_u$, where $c_{v_i} \in C_c$. Assume that preference statement s_u is independent of c_{v_i} and hence the dominance relationships are not changed.

When $|C_{v-u}| = k+1$, there are $k+1$ different condition values. Compared with the previous case, the extra condition value is $c_{v_{k+1}}$ that belongs to attribute X_i. The preference $p_u = c_u : s_u$ is transformed into multiple preferences $p_u^{(k+1)i} = c_u \vee c_{v_i} : s_u = c^k \vee c_{v_{k+1}} : s_u = c^{k+1} : s_u$. In comparison with p_u^{ki}, there are $|D(X_i)|$ more preferences generated. Since all values of attribute X_i are occurred in the newly generated CPT$'(X_j)$ when the preference statement is s_u, s_u is independent of the values of attribute X_i, i.e. $c_{v_{k+1}}$. According to the previous case where s_u is also independent of c^k, we can conclude that preference statement s_u is independent of the values of C_{v-u} and the extension is equivalent, i.e. the dominance relationships are kept the same. ■

C. The Algorithm

This section presents the pseudo codes and the time complexity required for calculating user similarity.

1) Pseudo Codes: Algorithm 1 elaborates the procedures to calculate user similarity based on CP-nets in three steps. Firstly, comparable preferences are identified and converted according to Definition 2. And then each of the identified preferences is decomposed into partial orders and the number of deduced pre-orders is computed. Finally, user similarity is calculated using Equation 1.

Input : CPTs $G(U), G(V)$
Output: Similarity $Sim(u,v)$

integer $N_c, N_a, N_{cp}, N_{ap}, N_u, N_v$;
foreach $X \in V_X$ **do**
  vector $vList \leftarrow \varnothing$;
  foreach $p_u \in CPT_u(X)$ **do**
    boolean $found \leftarrow false$;
    foreach $p_v \in CPT_v(X)$ **do**
      if $\{c_u\} \subseteq \{c_v\}$ *or* $\{c_v\} \subseteq \{c_u\}$ **then**
        $p_v \rightarrow vList$;
        $found \leftarrow true$;
        $c_u \leftarrow c_v$ or $c_v \leftarrow c_u$;
        $M_{p_u} \xleftarrow{decomposed} p_u$;
        $M_{p_v} \xleftarrow{decomposed} p_v$;
        $N_u \leftarrow \prod_{Y \neq X \cap Y \notin P_u(X)} |D(Y)|$;
        $N_v \leftarrow \prod_{Y \neq X \cap Y \notin P_v(X)} |D(Y)|$;
        $N_{cp} \leftarrow |M_{p_u} + M_{p_v}|$;
        $N_{ap} \leftarrow |M_{p_u}| + |M_{p_v}| - |M_{p_u} + M_{p_v}|$;
        $N_c \leftarrow N_c + N_{cp} \cdot N_v$;
        $N_a \leftarrow N_a + N_{ap} \cdot N_v$;
    if $!found$ **then**
      $M_{p_u} \xleftarrow{decomposed} p_u$;
      $N_u \leftarrow \prod_{Y \neq X \cap Y \notin P_u(X)} |D(Y)|$;
      $N_a \leftarrow N_a + |M_{p_u}| \cdot N_u$;
  foreach $p_v \in CPT_v(X)$ **do**
    if $!vList.has(p_v)$ **then**
      $M_{p_v} \xleftarrow{decomposed} p_v$;
      $N_v \leftarrow \prod_{Y \neq X \cap Y \notin P_v(X)} |D(Y)|$;
      $N_a \leftarrow N_a + |M_{p_v}| \cdot N_v$;
return $Sim(u,v) \leftarrow N_c / N_a$;

Algorithm 1: Similarity Measurement Procedure

According to Equation 1, two variables N_c and N_a are initialized, representing the number of common pre-orders and the number of all pre-orders derived from two users' CP-nets, respectively. For each attribute from attribute sets $X \in V_X$ (for simplicity, we ignore the subindex i), we check whether there are any conditional preferences p_v from user v that can be compared with another conditional preference p_u from user u (lines 2-6). If they are comparable (line 7), we put p_v into the visited list $vList$ and mark $found$ as true to indicate that comparable preference is found (lines 8-9). Subsequently, the condition is adjusted and preference matrix is generated accordingly (lines 10-12). After adjustment, the number of partial orders that can be decomposed and the number of pre-orders are computed and updated (lines 13-18).

If no comparable preference from user v is found for each user u's preference p_u (line 19), the number of all pre-orders due to u's solo preferences is updated (lines 20-22). Analogously, for the user v's preferences without comparable preferences found from user u, the number of all pre-orders due to v's solo preferences is update (lines 23-27). Finally, the similarity between users u and v is computed and returned as the ratio of the number of common pre-orders over the number of all pre-orders (line 28).

2) Time Complexity: Let n be the number of attributes in CP-nets and m be the maximum number of values that an attribute can have. An attribute will have the most preferences when all the left attributes are represented as its parent nodes. That is, for an attribute, it has at most $n-1$ parent nodes which produce m^{n-1} preferences if every parent node has m attribute values. The time complexity of Algorithm 1 mainly stems from three **foreach** iterations (lines 2-6): $O(n \cdot m^{n-1} \cdot m^{n-1}) = O(n \cdot m^{2n-2})$.

D. Example

This section is introduced here to exemplify how similarity between two users is calculated step by step. Continuing our previous example of user u's CP-nets in Figure 1, we assume that another user v is also using a certain data storage and access service and her preferences are illustrated in Figure 2. Different from user u's preferences, user v has two unconditional preferences on attributes A and C, and C is no longer dependent on attribute B. In contrast, user v concerns attribute B (Location) whose preferred value is determined by the values of both attributes A and C.

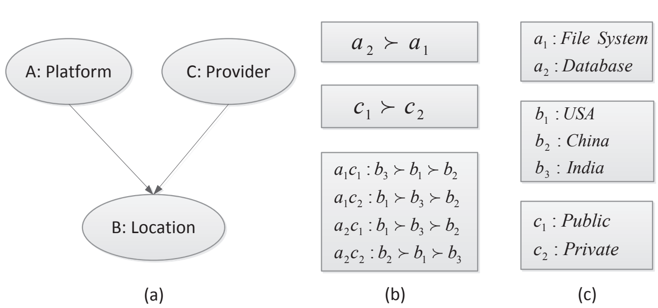


Fig. 2. (a, b) The CP-nets of User v; (c) Attribute Values.

More specifically, for attribute A, the preferences of users u and v are $p_u^A = a_1 \succ a_2$ and $p_v^A = a_2 \succ a_1$, respectively. There are no conditions for two preferences, or the preference conditions are empty ($\varnothing$). Thus they are comparable preferences according to Definition 2. In addition, since the statements only contain two attribute values, they are also partial orders and hence not necessary to be decomposed further. We then compare their preference statements which are totally opposite. Using Algorithm 1, the pre-orders deduced by both p_u^A and p_v^A will be added to the total number of pre-orders (N_a). Accordingly, the numbers of pre-orders deduced by p_u^A and p_v^A are computed by $N_u = \prod_{Y \neq A \cap Y \notin P_u(A)} |D(Y)| = |D(B)| \cdot |D(C)| = 6$ and similarly $N_v = 6$. Therefore, $N_c = 0$ and $N_a = N_u + N_v = 12$.

On attribute B, user u specifies two conditional preferences (for simplicity, we ignore the attribute notation B): $p_u^1 = a_1 : b_1 \succ b_2 \succ b_3$ and $p_u^2 = a_2 : b_2 \succ b_1 \succ b_3$ whereas user v indicates four conditional preferences: $p_v^1 = a_1c_1 : b_3 \succ b_1 \succ b_2$, $p_v^2 = a_1c_2 : b_1 \succ b_3 \succ b_2$, $p_v^3 = a_2c_1 : b_1 \succ b_3 \succ b_2$ and $p_v^4 = a_2c_2 : b_2 \succ b_1 \succ b_3$. For each preference of user u, we go through from p_v^1 to p_v^4 to check if there are any comparable preferences. For example, it is found that p_v^1 and p_v^2 can be compared with p_u^1 since their conditions are containable, i.e. $\{a_1\} \subset \{a_1c_1\}$ and $\{a_1\} \subset \{a_1c_2\}$. According to Theorem 1, we then transform the original p_u^1 into two new preferences: $p_u^{11} = a_1c_1 : b_1 \succ b_2 \succ b_3$ and $p_u^{12} = a_1c_2 : b_1 \succ b_2 \succ b_3$. After transformation, we can obtain $N_u = \prod_{Y \neq B \cap Y \notin P_u(B)} |D(Y)| = \prod_{Y \in \varnothing} |D(Y)| = 1$, $N_v = 1$. The next step is to decompose all the comparable preferences into partial orders, represented as follows.

$$M_{p_u^{11}} = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix} \quad M_{p_u^{12}} = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}$$

$$M_{p_v^1} = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix} \quad M_{p_v^2} = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}$$

Based on these preference matrices, we can yield $N_{cp} = |M_{p_u^{11}} + M_{p_v^1}| + |M_{p_u^{12}} + M_{p_v^2}| = 3$, $N_{ap} = |M_{p_u^{11}}| + |M_{p_v^1}| + |M_{p_u^{12}}| + |M_{p_v^2}| - N_{cp} = 9$. Hence, N_c and N_a can be updated by $N_c = N_c + N_{cp} \cdot N_u = 3$ and similarly $N_a = N_a + N_{ap} \cdot N_v = 21$. Analogously, same process holds for p_u^2 and after update, the numbers of common and all pre-orders are: $N_c = 7$ and $N_a = 29$.

For attribute C, the same procedure will be executed to search for comparable preferences, decompose partial orders and update the summation of common and all pre-orders: $N_c = 9$ and $N_a = 39$. Finally, the similarity between users u and v is computed as $Sim(u, v) = N_c/N_a = 9/39 = 0.2308$.

V. Experiments

Experiments are conducted in this section to verify the fidelity and effectiveness of our method. The fidelity refers to the extent to which computed similarity can correctly reflect the changes of user preference. The effectiveness refers to the extent to which computed similarity is effective to describe the correlation between two users.

A. Experimental Settings

Two real-life datasets are utilized, namely *Adult*[2] and QWS [11]. The former is obtained from the UCI Machine Learning Repository, consisting of 32,561 records. Each record is regarded hereafter as a concrete dating service that contains 14 attributes. An extra QoS attribute *Annual Salary* whose values are determined in a random way is added to each record. The latter contains 2,507 real web services which stem from public sources on the web including Universal Description, Discovery, and Integration (UDDI) registries, search engines, and service portals. Each web service contains nine QoS attributes. Due to the small size of original datasets, we expand the number of records to 30K by generating random values for each QoS attribute based on a normal distribution.

In order to stimulate real situations, an amount of 10K CPTs are generated, each of which represents a user's preference, according to the randomly generated preference dependency graphs. The completeness degree of users' CPTs is a parameter to control the generation of the CPTs. Besides, for evaluating the effectiveness, a number of (varying from 0 to 5 uniformly) records are generated for each user regarding their behaviors on choosing preferred data storage services. These records are generated based on the complete preferences of these users.

B. Fidelity of our Method

Two batches of experiments are conducted to investigate the fidelity of computed user similarity in terms of selected services similarity, and similarity trend when a certain ratio of common preferences are changed continuously.

1) Selected Services Similarity: In general, if the computed similarity is correct, then the more similar two users are in preferences, the more common services they may select. Therefore, we randomly generate two CP-nets and set the completeness of CPTs 1. Based on the CP-nets, services are selected accordingly. After that, user similarity is computed using our method and selected service similarity is computed using the method introduced in [2]. This process is repeated 100 times and the obtained data is plotted in Figure 3. The results show that generally, when user similarity is high, the selected service similarity[12] is also high.

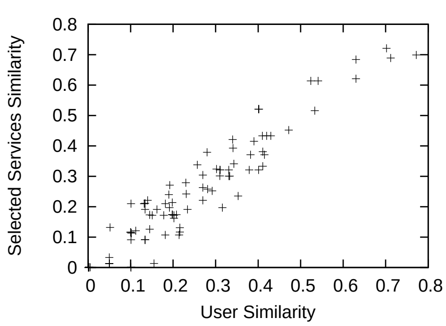


Fig. 3. Correlation between User Similarity and Selected Service Similarity

[2]http://archive.ics.uci.edu/ml/datasets/Adult

2) Similarity Trend with Preference Changes: This section aims to verify the intuition that computed similarity will change corresponding to the changes of user preferences. Specifically, for the preferences of a certain user, we make a certain ratio of changes each time to generate a new set of preferences, i.e. CPTs. To reduce biases, each time we will generate 1000 CPTs and the average of corresponding 1000 similarity values (with original CPTs) is adopted. Along with the different values of change ratios, the relative similarity values are computed and recorded.

We adopt three different strategies to modify the preferences: 1) Modify preference statements and keep the others unchanged (denoted by S1); 2) Modify preference conditions (attribute dependency) and keep the others unchanged (S2); 3) Randomly select one of the above two methods each time to modify user's preferences (S3).

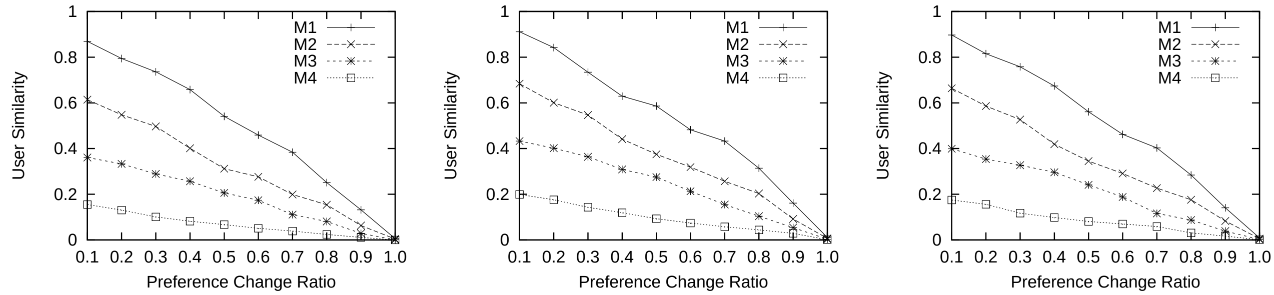
For each type of strategies, we vary the number of attributes from 5 to 9 and the number of attribute values from 2 to 4. The process of service selection is repeated 1000 times and then the average of computed similarity values is adopted. Four experiments (denoted by M1, M2, M3, M4) are carried out for each strategy where the latter experiment (e.g. M4) makes modifications based on the previous modifications made by the former step (e.g. M3), in order to make sure no modifications happen in the same preference components (conditions or statements). The results are illustrated from Figure 4 to Figure 6, corresponding to three different modification strategies, respectively. Consistent trends are obtained from the results: as the ratio of preference changes increases, the computed user similarity is decreased accordingly as expected.

C. Effectiveness of our Method

We evaluate the effectiveness of our method by applying similarity measure into an application scenario where user preferences are incomplete or missing. More specifically, we will complement the missing preferences of some attributes for a certain user by aggregating the preferences of other similar users. We will elaborate the scenario first and then design a set of experiments to verify the effectiveness of our method.

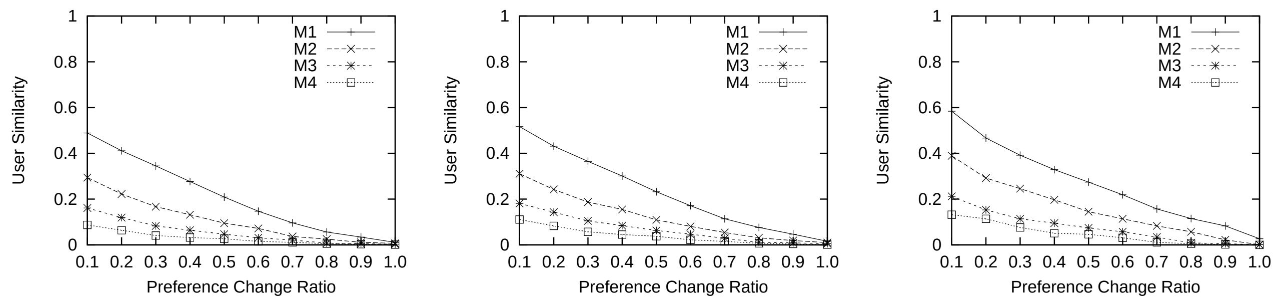
1) Incomplete Preference Scenario: In a service system with multiple participants, users can explicitly describe their preferences for better service selection. However, due to some reasons such as lack of knowledge or proper incentives, users may not provide a detailed, complete preference statements, which in general results inability for the system to provide effective service choices. In this case, users are not able to select the service that most satisfies their real preferences. To solve this problem, we borrow the idea of collaborative filtering [1] from recommender systems.

More specifically, an active user's preference can be predicted based on the preferences of other similar users. In this scenario, user's missing preferences are complemented by aggregating the preferences of other similar users that are detected based on the proposed similarity measure. Specifically, for a preference specified by a similar user on an attribute where no preferences are specified by the active user,



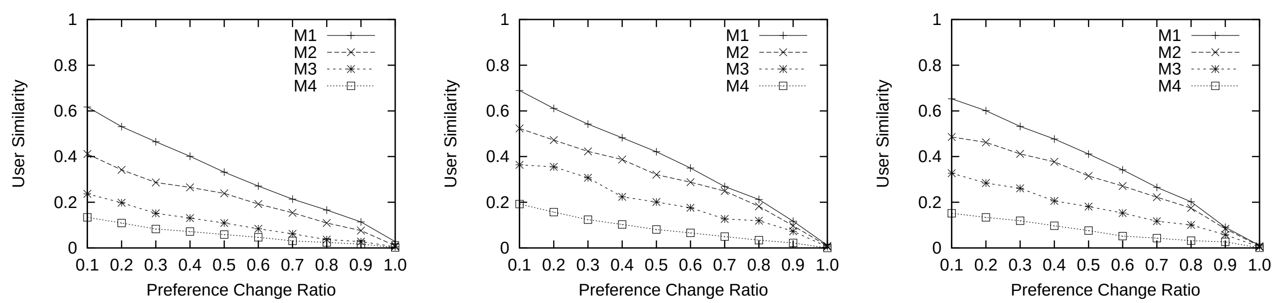
(a) (b) (c)

Fig. 4. Strategy S1: Modify Preference Statements with (#attributes, #attribute values) (a) (9, 2); (b) (5, 4); (c) (9, 4).



(a) (b) (c)

Fig. 5. Strategy S2: Modify Preference Conditions with (#attributes, #attribute values) (a) (9, 2); (b) (5, 4); (c) (9, 4).



(a) (b) (c)

Fig. 6. Strategy S3: Modify Preferences Randomly by S1 or S2 with (#attributes, #attribute values) (a) (9, 2); (b) (5, 4); (c) (9, 4).

we count the number of similar users (votes) who state this preference in their CPTs. The preference with the greatest votes will be adopted as the preference of the active user. After complementing missing preferences, the system will provide users with a new set of services that satisfy their preferences. Hence, the effectiveness of our method can be reflected by the differences between the accuracy of recommended services before and after preferences completion. If the accuracy is significantly improved, we claim that the similarity measure is effective since similar users are identified effectively.

2) Experiments and Results: Two methods are used as the benchmarks, including the method proposed by Wang et al. [2] (denoted by *Dis*) and the method before adjusting preferences (*Before*), in contrast to our method after predicting the missing preferences (*After*). More specifically, we generate a complete CPTs for an active user u based on which the system will recommend a set of services, denoted as the benchmark services S_p. Then, only 40% of all preferences from u's CPTs are kept and the others are removed to form incomplete preferences. Using these incomplete preferences, a set of services are recommended, denoted as S_{ip}. The accuracy of current recommendations is computed by

$$r_i = \frac{|S_{ip} \cap S_p|}{|S_{ip}|} \tag{3}$$

In addition, 1000 other users are created whose completeness of preferences is 60%. Similar users are identified as those whose similarity with user u is greater than a threshold θ. The missing preferences of user u are complemented by the detected similar users. A set of services after complementing preference are generated, denoted as S_{cp}. Then the accuracy r_c of these recommendations is obtained by Equation 3 where S_{ip} is substituted by S_{cp}. We adjust the similarity threshold to control the set of similar users used for predicting missing preferences. The experiments are conducted on two real-life datasets. The results are delineated in Figures 7 and 8.

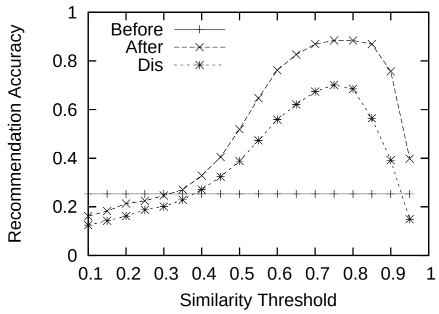


Fig. 7. Effectiveness of Our Method on Adult dataset

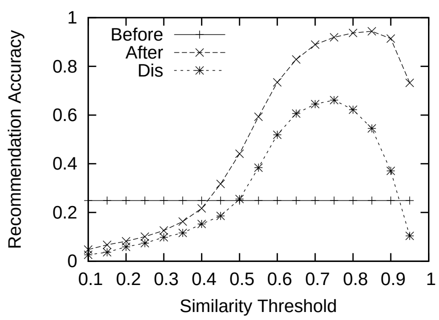


Fig. 8. Effectiveness of Our Method on QWS dataset

The results show that before preference adjustment, recommendation accuracy is kept low (around 0.25 for both datasets) and complementing preferences (both *Dis* and *After*) has a significant impact on recommendation accuracy. Specifically, as the similarity threshold goes up, the recommendation accuracy of our method *After* is increased first up to the highest (0.869 for Adult and 0.944 for QWS) and then decreased. Compared with the situation before adjustment, the accuracy is worse before some threshold values (around 0.3 for Adult and 0.4-0.45 for QWS) but better after these values. The method *Dis* obtains similar results. One possible explanation is that complemented preferences may be not accurate based on the preferences of many less similar users due to the low threshold value. In contrast, when the threshold is high, more reliable similar users are chosen and used to complement users' preferences. Thus the complemented preferences are more accurate and so as the recommendations. However, if the threshold is too high, few similar users will be involved in the prediction procedure and thus decrease the accuracy of completed preferences. Our method *After* is superior to *Dis* because two preferences of the latter are considered as comparable if and only if their conditions are identical which results in less similar users can be identified. Consequently, the recommendation accuracy is less accurate.

VI. Conclusion

This paper proposed a novel method to calculate user similarity based on explicit qualitative preferences expressed by themselves and represented by CP-nets. Instead of relying on the structure of services as in [2], we investigated user similarity at a lower level of CPTs. More specifically, a conditional preference was decomposed into multiple partial orders whose relationships in attribute values were represented by a preference statement matrix. A method to detect and adjust comparable preferences was proposed in Section IV-B. Besides, the detailed algorithm for computing similarity was delineated in Algorithm 1. To verify the correctness and effectiveness of our method, we conducted a series of experiments based on two expanded real-life datasets. The consistent results demonstrated that our similarity measure is not only able to reflect user's preference changes, but also effective to identify similar users.

Acknowledgment

This work is partially supported by NSFC (61232007) and JSNSF of China (No.BK2010417).

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," pp. 734–749, 2005.

[2] H. Wang, S. Shao, X. Zhou, C. Wan, and A. Bouguettaya, "Web service selection with incomplete or inconsistent user preferences," *Service-Oriented Computing*, pp. 83–98, 2009.

[3] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole, "Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements," *J. Artif. Intell. Res. (JAIR)*, vol. 21, pp. 135–191, 2004.

[4] X. Sun and J. Liu, "Representation and realization of binary-valued cp-nets in single-branch tree," in *Proceedings of the 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, vol. 4, 2010, pp. 1908–1911.

[5] H. Wang and W. Liu, "Web service selection with quantitative and qualitative user preferences," in *Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 1, 2011, pp. 404–411.

[6] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," vol. 22, no. 1, pp. 5–53, 2004.

[7] N. Lathia, S. Hailes, and L. Capra, "Private distributed collaborative filtering using estimated concordance measures," in *Proceedings of the 2007 ACM conference on recommender systems*. ACM, 2007, pp. 1–8.

[8] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon, "Pagerank, hits and a unified framework for link analysis," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2002, pp. 353–354.

[9] D. Lizorkin, P. Velikhov, M. Grinev, and D. Turdakov, "Accuracy estimate and optimization techniques for simrank computation," *The VLDB JournalThe International Journal on Very Large Data Bases*, vol. 19, no. 1, pp. 45–66, 2010.

[10] W. Lu, J. Janssen, E. Milios, and N. Japkowicz, "Node similarity in networked information spaces," in *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 2001, p. 11.

[11] E. Al-Masri and Q. Mahmoud, "Discovering the best web service," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 1257–1258.

[12] H. Wang, J. Zhang, Y. Tang, and S. Shao, "Collaborative approaches to complementing qualitative preferences of agents for effective service selection," in *Proceedings of the 2011 IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 2011, pp. 51–58.