2011 23rd IEEE International Conference on Tools with Artificial Intelligence

Collaborative Approaches to Complementing Qualitative Preferences of Agents for Effective Service Selection

Hongbing Wang Jie Zhang* Yangyu Tang Shizhi Shao
Key Lab of Computer Network and Information Integration
School of Computer Science and Engineering, Southeast University, China
*School of Computer Engineering, Nanyang Technological University, Singapore
{hbw@seu.edu.cn}

***Abstract*—Incomplete preferences of agents may render service selection ineffective. We address this problem by proposing a set of collaborative approaches to complementing agents' incomplete preferences in a qualitative way. For an agent, the approaches first find its similar agents, and then base the similar agents' qualitative preferences to complement this agent's missing preferences. We analyze and compare these approaches and provide experimental results to justify our arguments. We also compare our approach with the classic collaborative filtering and show the competitive advantages of our approach in service selection. Our work thus serves as an important step towards effective service selection.**

***Keywords*-Service Selection; Collaborative Filtering; CP-nets; Multi-Agents;**

I. Introduction

With more and more services emerging on the Internet, service selection becomes a significant approach [1], [2] for agents to find their desirable services. During the process of service selection, agents' preferences on nonfunctional attributes of services can be helpful for selecting more satisfactory services to them. However, it is often that agents' preferences are incomplete. Incompleteness of preferences represents an absence of knowledge about the relationship between certain pairs of outcomes. It arises naturally when we have not fully elicited agents' preferences or when agents have difficulty with explicitly specifying their complete preference orderings.

Some approaches have been proposed for service selection based on incomplete preferences of agents [3]. However, these approaches simply make use of conditional ceteris paribus, a concept of all else being equal. To explain, if an agent does not specify its preferences on a service attribute, it is assumed that the agent has no preference difference over different values of the attribute. This results into many services that are unable to be differentiated, and the pool of services recommended to the agent is often large, rendering service selection ineffective.

We address the incomplete preference problem by proposing a set of collaborative approaches. Agent preferences are represented in a qualitative way in our approaches using CP-nets [4]. This is different from quantitative representations of many existing approaches [5] that use an utility function to make a quantitative assessment of the agents' preferences. In many complicated and realistic cases, a quantitative approach may be less successful [6].More specifically, we propose a clustering-based approach, a threshold-based approach and a behavior-based approach. For an agent with incomplete preferences, these approaches find a set of similar agents as that agent. The preferences of those similar agents will then be used to complement the missing preferences of the agent. The clustering-based and threshold-based approaches find similar agents based on the available preferences of all agents (i.e. available information in their CP-nets). The behavior-based approach, in contrast, makes use of agents' behavior of choosing services to find the other agents with similar behavior as the agent.

We analyze and compare the effectiveness and complexity of these approaches and provide experimental support for our analysis, using a real dataset. We further compare the behavior-based approach with the classic collaborative filtering approach in terms of the effectiveness in recommending services and the computation complexity. The experiments confirm that our approaches provide promising results and heavily improve the effectiveness of service selection.

II. Related Work

Many quantitative approaches have been proposed for conducting service selection efficiently. For example, Lamparter et al. [5] have used utility function policies drawing from the multi-attribute decisions theory to develop algorithms for optimal service selection. Unfortunately, because of lacking agents' preferences on attributes of services (incomplete preferences), the results selected in the quantitative manner may not well suit agents' real requirements. To address the incomplete preference problem in quantitative preferences, different preference elicitation approaches have been proposed. For example, Hines and Larson [7] recently introduce a querying method that allows a combination of minmax regret preference elicitation and cumulative prospect theory, a descriptive model of human reasoning for risky choices. Of course, these preference elicitation approaches are also quantitative.

Although quantitative methods for service selection complemented by preference elicitation may be efficient, they may be less successful in more complicated and realistic scenarios [6]. In addition, preference elicitation may require numerous interactions between agents and their human users, which is costly. On another hand, qualitative approaches may

1082-3409/11 $26.00 © 2011 IEEE
DOI 10.1109/ICTAI.2011.17

be more feasible, natural and general [6], [4]. Garcia et al. [8] present a service selection framework that transforms qualitative preferences into an optimization problem. However, the problem of incomplete qualitative preferences also exists. Wang et al. [3] propose an approach to order service patterns in multi-agent scenarios, where each agent may express different preferences on service attributes which may be incomplete. Service patterns are ordered first for each agent. Finally, the strict order of service patterns for multiple agents can be calculated with the consideration of the weight of each agent. However, this approach simply makes use of conditional ceteris paribus, a concept of all else being equal, to deal with incomplete preferences, which may result in a large number of service patterns' orders not being differentiated. In contrast, we propose three effective approaches to complementing agent preferences in a qualitative way.

In our approaches, the idea of collaborative filtering is adopted. The missing preferences of one agent can be complemented by other similar agents' preferences. However, our approaches are evidently different from the traditional recommendation systems using collaborative filtering for the recommendation of services [9], in several aspects. First, traditional recommendation systems usually provide recommendations of services with implicit representation of preferences. Our approaches, on another hand, represent preferences in an explicit way using CP-nets. In this transparent way, human owners of agents can directly modify their preferences if needed. Second, traditional recommendation systems have to compute similarity between agents every time when a service needs to be recommended, which is known to be computationally expensive. Our approaches, on another hand, once preferences are completed, can directly provide recommendations of services based on agent preferences, thus saving much computation time.

III. Background

CP-net [4] is a graphical model for representing conditional preference relations in a qualitative manner. And now it becomes popular as an effective tool to model preferences expressed by agents representing their users.

Definition 1. Let V = X_1,...,X_n be a set of attributes of services. A CP-net over V is a directed graph G (called *dependency graph*) over X_1,...,X_n, in which each node is annotated with a Conditional Preference Table, denoted by CPT(X_i) that associates a total order of X_i's values with each instantiation of X_i's parents.

As shown in Figure 1, a simple example of a CP-net for Jane (represented by an agent) regarding dating services can be described by several attributes, including Native Country (**A**: China or America), Dating Site (**B**: Chinese restaurant or coffee shop) and Dessert (**C**: tea or coffee).

For the attribute of Native Country, as a Chinese girl, Jane always prefers her dating mate born in China because of traditional consciousness, but her preference on attribute Dating Site depends on the native country of her dating mate.

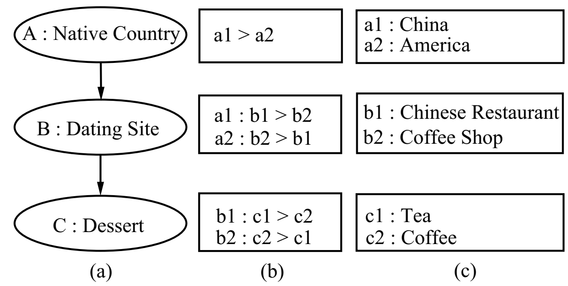

Fig. 1. (a, b) CP-net for Jane's Agent; (b) CPTs; (c) Attribute Constraints

If her dating mate was born in China, then she would rather make an appointment in a Chinese restaurant than in a coffee shop. If her dating mate was born in America, she would like to choose a coffee shop to meet. Moreover, Jane's preference on attribute Dessert depends on the kind of Dating Site she chooses. If she enjoys an appointment with her partner in a Chinese Restaurant, she prefers tea over coffee after dinner. On the other hand, if she chooses a coffee shop to meet with her partner, she would rather drink coffee than tea.

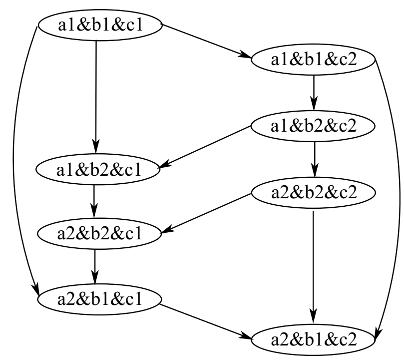

Fig. 2. Induced Preference Graph

A. Agent-based Service Selection

In a typical scenario of agent-based service selection, a human user represented by an agent describes her preferences through the agent and following the rules of CP-net. Then the system identifies relevant services or service compositions that meet the agent's requirements. The process of service selection can be further broken down as follows. Firstly, the system generates service patterns, each of which is a combination of attribute constraints (values) for service attributes. These patterns will then be ranked according to an induced preference graph from the agent's CP-net [4], [3]. From top to bottom based on their positions in the graph, patterns ranked the best to the worst. Finally, the concrete services that match the best service patterns will be recommended to the agent.

For example, according to Jane's CP-net representation of preferences in Figure 1, we can easily induce a detailed preference graph for her, as shown in Figure 2. The preference graph represents her explicit preferences among all possible types of services, each of which is a service pattern. The service pattern (a_1&b_1&c_1) that consists of China, Chinese

Restaurant and tea can be seen as the best choice for her. If an available service satisfies all the preference constraints in this pattern, it will be recommended to Jane.

B. A Scenario of Incomplete Preferences

However, it is often that agents' preferences are incomplete. Incompleteness of preferences represents an absence of knowledge about the relationship between certain pairs of outcomes. It arises naturally when we have not fully elicited agents' preferences or when agents have difficulty with explicitly specifying their complete preference orderings. Let us come back to the earlier example of selecting dating services for Jane. Suppose Jane does not specify her preferences on Dating Site. Because of this, Jane would receive more candidates that match her preferences according to the semantics of CP-net. For instance, service patterns $a_1\&b_1\&c_1$ and $a_1\&b_2\&c_1$ become incomparable. So, services matching both these patterns will be recommended to Jane. The effectiveness of service selection is thus decreased. We address this problem by complementing agent preferences.

Input : CP-nets of m agents; CP-net of agent a_0; number of clusters n; threshold α
Output: A set of similar agents SA

1 Choose n CPTs as cluster centers randomly;
2 **while** *true* **do**
3   **foreach** *agent* a_i *(*$1 \leq i < m$*)* **do**
4   **foreach** *cluster center* c_j *(*$1 \leq i \leq n$*)* **do**
5   Calculate $Sim(a_i, c_j)$ by Equations 1 or 2;
6   Find the closest cluster, put a_i in it;
7   Calculate *precise* of each cluster as the sum of mean square deviation;
8   **if** *precise* $> \alpha$ **then**
9   Choose CP-net with smallest average distance to other CP-nets for each cluster as new center;
10   **else**
11   Exit while loop;
12 Calculate similarity between a_0 and each center;
13 Return SA $\Leftarrow$ agents in the closest cluster;

Algorithm 1: The Clustering-based Approach

IV. Complementing Preferences

In the system, each agent holds a CP-net representing its human user's preferences, which may be incomplete. Given an agent a_0 with an incomplete CP-net, in this section, we present a set of collaborative approaches for complementing agent a_0's preferences, including a clustering-based approach, a threshold-based approach and a behavior-based approach. The commonality among them is that they all first find similar agents as agent a_0 and then use these agents' CP-nets to complement a_0's. But, they are different in their ways of finding similar agents. We will also analyze and compare the advantages and disadvantages of these approaches in terms of both effectiveness and computation complexity.

A. Finding Similar Agents

Three different ways of finding similar agents as agent a_0 are used in the respective collaborative approaches.

1) A Clustering-based Approach: Our clustering-based approach and the threshold-based approach both are based on the assumption that if two agents have similar preferences on some service attributes, their preferences on other service attributes should also be similar. The clustering-based approach finds similar agents by first applying the *K*-means algorithm to cluster the agents according to their preferences (*K* is set to 5 in experiments, to achieve the best performance). It then compares agent a_0's CP-net representation with each cluster center and finds the cluster whose center is the most similar. The agents in this cluster will be considered as similar agents. A pseudo code summary in Algorithm 1 details this process.

In Algorithm 1, by calculating the similarity of agents, the agents can be grouped into several clusters (Line 5). Our clustering-based approach calculates the similarity of two agents based on their specified CP-nets. More specifically, it is calculated as the size of overlapping between two induced preference graphs divided by the size of the overall graph (e.g. the merger of the two graphs).

Definition 2. Let A and B be two CP-nets of an abstract service, $G(A)$ and $G(B)$ be the induced preference graphs of A and B respectively, and *e* denote an edge in a preference graph. Similarity between A and B can be calculated as:

$$Sim(A:B) = \frac{|\{e : e \in G(A) \wedge G(B)\}|}{|\{e : e \in G(A) \vee G(B)\}|} \tag{1}$$

It is an intuitive way to calculate the similarity in Equation 1, but the size of an induced preference graph grows exponentially with the number of service attributes. It will be infeasible to use Equation 1 to compute the agents' similarity when a large number of attributes are considered in service selection. Fortunately, we have an equivalent but simpler method according to characteristics of CP-net. Given a particular abstract service, we assume that different agents' CP-nets share the same dependency graph (e.g., Figure 1(a)). This assumption is based on two facts. First, the dependencies among the attributes of a certain service type are usually determined by the inherent characteristics of these attributes themselves. Second, even when agents specify different dependency graphs in their CP-nets, we can create a common dependency graph for them by combining their dependency graphs into one. The agents' CP-nets can be adjusted accordingly to the more complex common dependency graph, without varying their semantics. When CP-nets share a common dependency graph, their similarity can be directly calculated from their CPTs.

Definition 3. Let X_1,...,X_n be the attributes of an abstract service S. Let A and B be two CP-nets of S which share the same dependency graph. Let $D(X_i)$ denote the set of attributes which X_i depends on. Let $R(X_i)$ be the set of values that can be assigned to X_i. Then the similarity between A and B can be calculated as follows:

$$Sim(A:B) = \frac{\sum_{X_i}(|T_A(X_i) \cap T_B(X_i)| \times \prod_{X_j \notin D(X_i)}|R(X_j)|)}{\sum_{X_i}(|T_A(X_i) \cup T_B(X_i)| \times \prod_{X_j \notin D(X_i)}|R(X_j)|)} \quad (2)$$

where we substitute CPT by T for the purpose of simplicity.

To explain the equality of Equations 1 and 2, we revisit the dating example. Each preference specified by Jane determines certain edges in the induced preference graph. The number of edges corresponding to each preference regarding attribute X can be calculated by $\prod_{Y \notin D(X)}|R(Y)|$. For example, a preference China $>$ America corresponds to some edges, the number of which is computed by $|R(B)| \times |R(C)| = 2 \times 2 = 4$. In fact, the number of common edges of two induced preference graphes equals to the number of edges computed by the common preferences.

2) A Threshold-based Approach: Our threshold-based approach is similar to the clustering-based approach but with some variation. It calculates the similarity between agent a_0 and every other agent using Equations 1 and 2. An agent is similar to agent a_0 if its similarity value exceeds a pre-defined threshold. Algorithm 2 details this process.

```
Input : CP-nets of agents; the total number of
        agents m; CP-net of agent a_0; threshold β
Output: A set of similar agents SA
foreach agent a_i(1 ≤ i < m) do
    Calculate similarity between a_i and a_0 using
    Equations 1 and 2;
    if Sim(a_0, a_i) > β then
        Put a_i into similar agents set SA;
Return SA;
```

Algorithm 2: The Threshold-based Approach

3) A Behavior-based Approach: The two approaches mentioned above are based on comparing agents' explicit preferences represented by CP-nets to find similar agents and using the similar agents' available preferences to complement agent a_0's incomplete preferences. These two approaches are particularly useful when the system does not have any other extra information about agent a_0 (i.e. agent a_0 is new to the system). However, this type of approaches does not work well when agents share only partially similar preferences.

In this section, we introduce our third approach, a behavior-based approach. This approach is based on agents' behavior of choosing best services, which is a reliable information source and implicitly reveals the agents' preferences [9]. Thus, if two agents have similar recorded behavior, they are also likely to share similar preferences. It works for the situation where the system records agents' historical behavior of choosing their satisfactory services, and a sufficient amount of such information of agent a_0 also exists. Our behavior-based approach uses recorded behavior of agents to find a set of similar agents with agent a_0. For example, agent A and agent B have corresponding service sets which respectively contain the services chosen by agents A and B. The similarity of the two service sets can be calculated to judge whether these two agents have the similar preferences. As the process of calculating the similarity of behavior needs to calculate the similarity between services. Thus, we first define the similarity between a pair of services in Definition 4.

Definition 4. Let a and b be two concrete services in the same domain (such as *dating service*). Each can be described as a vector, of which its items are the values to corresponding service attributes. q_a^k represents the value of the kth attribute of service a. Here, the values of the former m attributes among a total of n attributes are continuous, and that of the remaining attributes are discrete. And, w_k is the weight of the kth attribute which can be preprocessed. The similarity of two services can be calculated as follows:

$$Sim(a,b) = \sum_{k=1}^{m} w_k \times (1 - |q_a^k - q_b^k|) + \sum_{k=m+1}^{n} w_k \times f(q_a^k, q_b^k) \quad (3)$$

where $f()$ for attributes with discrete values is defined as:

$$f(x,y) = \begin{cases} 1 & \text{if } x = y; \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Note that when computing the similarity, the vectors of services a and b should be preprocessed to map the continuous value of each attribute to a value between 0 and 1. The course of this preprocessing is called standardization. We apply the Gaussian Distribution [10] to complete this standardization process. The weight of an attribute can be determined by the number of agents that have specified preferences on this attribute, indicating the importance of the attribute to all agents. These processes can be done off-line.

```
Input : CP-nets and behavior sets of m agents;
        CP-net and behavior set (size k) of a_0;
        threshold γ
Output: A set of similar agents SA
foreach agent a_i(1 ≤ i < m) do
    Form a r_{a_i} × k Similarity Matrix M;
    //r_{a_i} is the size of a_i's behavior set
    Set behavior similarity Sim = 0;
    while M is not empty do
        Find highest similarity s_h and add to Sim;
        Eliminate the column and row of s_h;
    Sim = Sim/min(r_{a_i}, k);
    if Sim > γ then
        Put a_i into SA;
Return SA;
```

Algorithm 3: The Behavior-based Approach

The behavior similarity of two agents can be attributed to computing the similarity between service sets composed of the services which have been chosen or used by them. For this purpose, we create a *Similarity Matrix* defined below.

Definition 5. Let service set S^a=$\{S_1^a$,...,$S_n^a\}$ be the behavior records of agent a, S^b=$\{S_1^b$,...,$S_m^b\}$ be the behavior records of agent b. S_i^a and S_j^b are two services once used by agents a and b respectively. An $n{\times}m$ matrix can be created with the element $M[i][j]$ assigned by $Sim(S_i^a,S_j^b)$.

With the help of the *Similarity Matrix*, we can easily compute the behavior similarity of agents, as detailed in Algorithm 3 within the process of the behavior-based approach. For example, assume that agent a has accessed two services: S_3^a and S_5^a and agent b has accessed three services: S_6^b, S_{13}^b and S_3^b. Then we build a 2×3 *Similarity Matrix*. The process of computing behavior similarity is illustrated as:

$$\begin{bmatrix} & S_6^b & S_{13}^b & S_3^b \\ S_3^a & 0.9 & 0.6 & \mathbf{1} \\ S_5^a & 0.7 & 0.8 & 0.6 \end{bmatrix} \Rightarrow \begin{bmatrix} & S_6^b & S_{13}^b \\ S_5^a & 0.7 & \mathbf{0.8} \end{bmatrix}$$

The behavior similarity is then equal to (1+0.8)/2 = 0.9.

4) Analysis and Comparison of the Approaches: Let us revisit our three approaches, analyze and compare their effectiveness and computation complexity.

The clustering-based and threshold-based approaches both are based on comparing agents' explicit preferences represented by CP-nets to find similar agents and using these similar agents' available preferences to complement the current agent a_0's incomplete preferences. These two approaches are particularly useful when the system does not have any other extra information about a_0 (i.e. a_0 is new to the system). However, this type of approaches does not work well when agents share only partially similar preferences. The behavior-based approach, on another hand, is based on agents' behavior of choosing best services. Behavior is considered to be a reliable information source and implicitly but accurately reveals the agents' preferences. Thus, the behavior-based approach should be more effective than the other two approaches. We also argue that the threshold-based approach is more effective than the clustering-based approach. The former calculates similarity between agent a_0 with each other agent, but the latter calculates the closeness with a cluster of agents and treats the (similar) agents in the closest cluster the same, ignoring the differences among them.

In terms of computation complexity, we argue that the clustering-based approach is the best, and the behavior-based approach is the worst in most of the cases. More specifically, for the clustering-based approach, the process of clustering agents could be performed off-line. The time is then mainly spent on comparing the current agent's CP-net with the center of every cluster. For Algorithm 2, computation is mainly caused by Equation 2. For Algorithm 3, much computation is spent on building Similarity Matrices. The time complexity of Algorithm 3 is the largest among the three approaches.

B. Complementing Agent Preferences

Using the three approaches mentioned above, we can identify a set of agents with similar preferences as the current agent a_0. The missing preferences of a_0 can then be complemented by the preferences of those similar agents. We adopt the concept of collaborative filtering. If a preference is supported by a larger number of similar agents, the likelihood of this preference matching a_0's true preference is higher. Thus, for each missing preference of a_0, similar agents give a vote if this preference is supported by their CP-nets. Finally, the preference with the largest number of votes will be used to supplement a_0's respective missing preference.

The votes for candidate preferences affect the result of service pattern ranking (part of service selection process, see Section III-A) for agent a_0. Thus, it is important to also consider the ability for distinguishing service patterns. For instance, if one preference has received many supporting votes from similar agents but accepting this preference cannot better distinguish service patterns, this preference has no use to a_0. In our complementing preference scheme, we take both the two factors into consideration and propose a scoring function for candidate preferences, as follows:

$$score(P) = \lambda{\times}likelihood(P){+}(1{-}\lambda){\times}selectivity(P) \quad (5)$$

where, $likelihood(P)$ is used to measure how much it is supported by similar agents, and is computed as follows:

$$likelihood(P) = (V_P - V_{\min})/(V_{\max} - V_{\min}) \quad (6)$$

where V_P is the total votes for candidate preference P, $V_{\max}$ is the maximum total votes among all candidate preferences, and $V_{\min}$ is the minimum total votes among all candidate preferences. Here, a vote for a preference is a similarity value between a_0 and the agent having the preference. $selectivity(P)$ is the ability for distinguishing service patterns, calculated as:

$$selectivity(P) = N_P/N_{total} \quad (7)$$

where N_p is the number of service patterns that can be distinguished by adding P, and N_{total} is the total number of service patterns. Besides, λ is a balance factor between $likelihood(P)$ and $selectivity(P)$. It is an empirical value which is usually set as 0.5. Overall, the candidate preference with the highest score will be chosen as a_0's preference.

V. EXPERIMENTATION

In this section, we begin with a set of experiments to verify the effectiveness and scalability of the three approaches mentioned above, and then carry out experiments to compare with a collaborative filtering approach.

We use the *Adult Data Set* obtained from the University of California Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/Adult) for conducting our experiments. The *Adult Data Set* consists of 32,561 records. Each record is used by us to represent a concrete dating service and is added by a QoS attribute *Annual Salary* whose values are produced in a random way. Together, each record has 15 attributes in total. We generate a preference dependency graph shared by all agents. In order to simulate real situations, a set of 1000 CPTs are generated, each of which represents an agent's preferences according to preference dependency graph. The completeness degree of agents' CPTs is a parameter to control the generation of the CPTs. Besides, for evaluating the behavior-based approach, we generate for each agent (called historical agents) a number of (varying

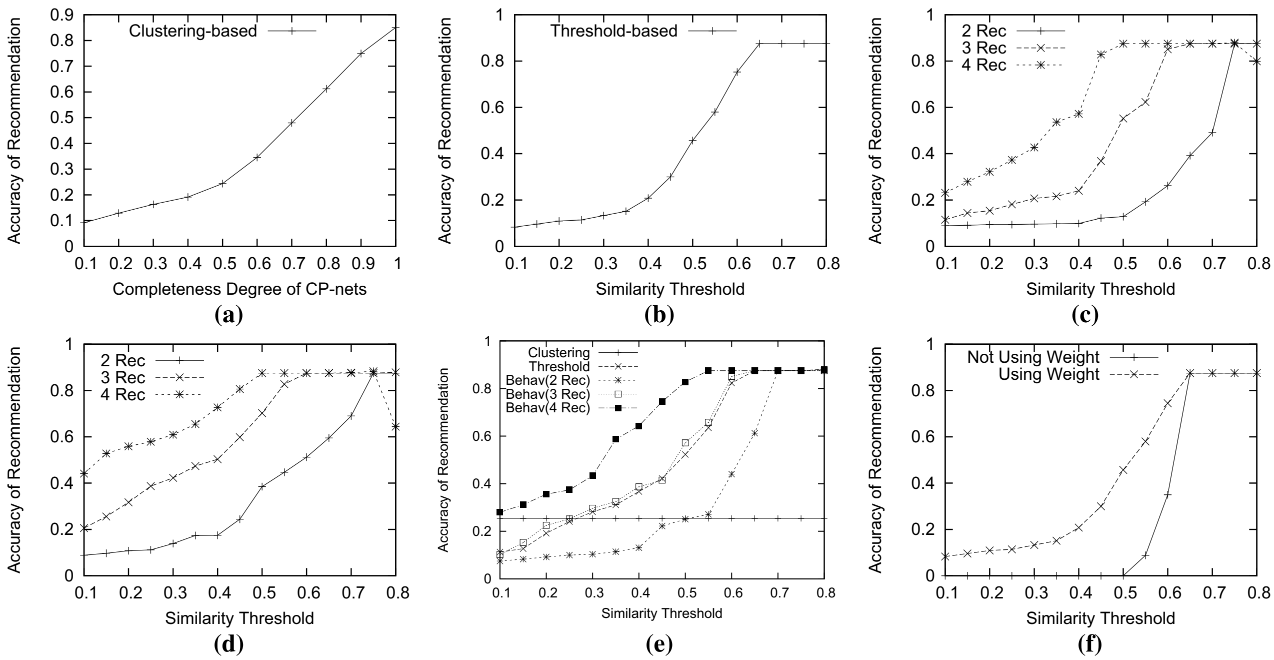

Fig. 3. (a) Effectiveness of Clustering-based Approach; (b) Effectiveness of Threshold-based Approach; (c) Effectiveness of Behavior-based Approach (40% of Completeness);(d) Effectiveness of Behavior-based Approach (70% of Completeness); (e) Comparison of the Approaches; (f) Weight of Similar Agents

from 0 to 5 uniformly) records of their behavior on choosing their best dating services. These records are generated based on the complete preferences of these historical agents. Note that all experiments implemented in Java are conducted in an IBM Server installed with Windows Server 2003 OS and with 8CPUs of 2.13GHz and a RAM of 16G.

A. Effectiveness of Our Approaches

In this set of experiments, we evaluate and compare the effectiveness of our three approaches. The accuracy of an approach for recommending services to an agent is evaluated by the agent's satisfaction with the recommended services. It is computed as the number of desirable services divided by the number of recommended services. The desirable services are generated based on the complete preferences of the agent. Each experiment is run for 1000 times.

The first experiment is to evaluate the accuracy of the clustering-based approach. The current agent's CP-net is 20% complete. In Figure 3(a), the accuracy of service selection is much improved after complementing incomplete preferences (compared to 1.5% when not complementing). The accuracy also depends on the completeness degree of historical agents' preferences. When historical agents' preferences are more complete, the accuracy of this approach is higher. In fact, this trend is also true for other two approaches.

We then explore the performance of the threshold-based approach. In this experiment, the completeness degree of historical agents' preferences is 50%. We vary the similarity threshold (β in Algorithm 2) from 0.1 to 0.8. In Figure 3(b), when the threshold increases, the accuracy of this approach also increases. When the threshold reaches between 0.65 and 0.8, the accuracy becomes stable (about 87.5%). When the threshold exceeds 0.8, the accuracy becomes erratic because only a very few similar agents could be found for complementing preferences. This kind of trend is also true for the behavior-based approach (see Figures 3(c) and 3(d)).

We demonstrate the effectiveness of the behavior-based approach when the completeness degree of historical agents' preferences is 40%. The similarity threshold γ changes from 0.1 to 0.8. The number of the current agent's behavior records also varies from 2 to 4. In Figure 3(c), the accuracy of the behavior-based approach increases with the increasing number of the current agent's behavior records. We run this experiment again for the case where 70% of preferences of the historical agents are complete. The performance becomes better in this case, which confirms our argument that the performance of the behavior-based approach is also affected by the completeness degree of the historical agents' preferences.

We also compare the performance of our three approaches for service selection. In this experiment, 50% of preferences of historical agents are complete. Results are shown in Figure 3(e). Because the clustering-based approach does not rely on the similarity threshold, its performance remains constant over different similarity threshold values. We can also see that if an appropriate value is chosen for the similarity threshold, both the threshold-based and the behavior-based approaches are better than the clustering-based approach. The accuracy of the threshold-based approach has a good match with the behavior-based approach when the current agent has only 3

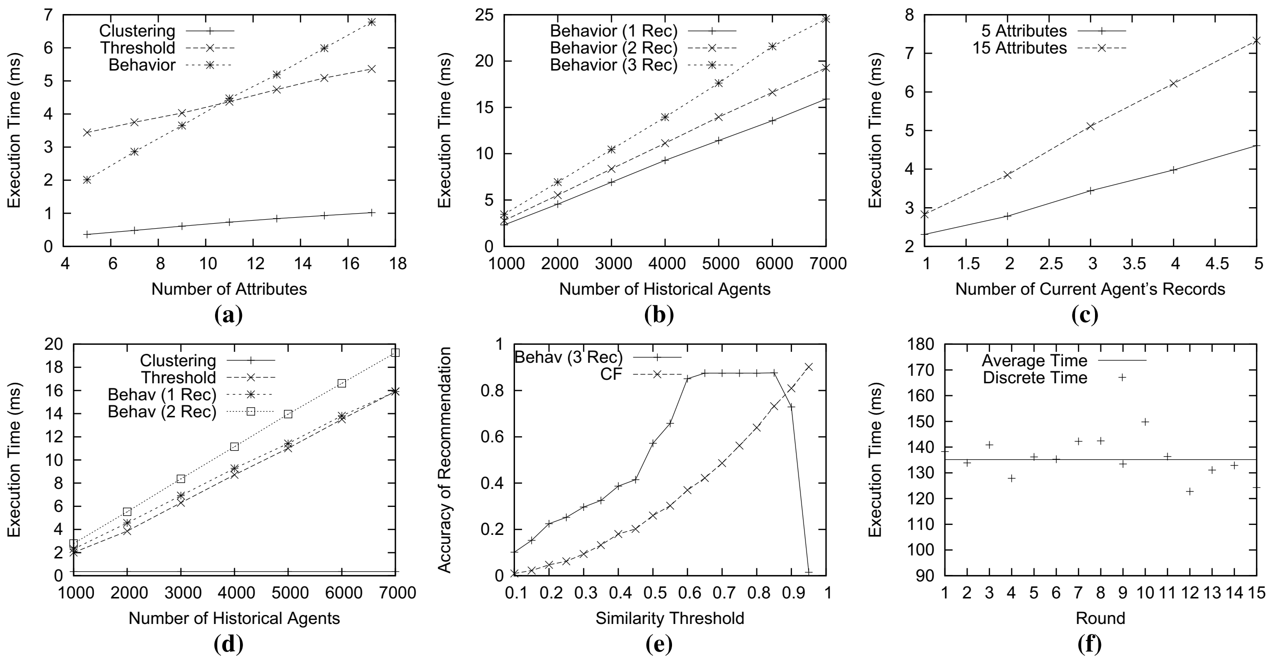

Fig. 4. (a) Runtime with Numbers of Service Attributes;(b) Runtime of Behavior-based Approach with Numbers of Historical Agents;(c) Runtime of Behavior-based Approach with Numbers of Current Agent's Behavior Records; (d) Runtime Comparison between Three Approaches; (e) Comparison between Behavior-based Approach and Collaborative Filtering; (f) Runtime of Collaborative Filtering

behavior records. However, when the number of the current agent's behavior records increases (i.e. to 4), the behavior-based approach dominates the threshold-based approach.

In general, the behavior-based approach outperforms the other two approaches, and the threshold-based approach is more effective than the clustering-based approach. This confirms our arguments in Section IV-A4. Both the threshold-based and behavior-based approaches are affected by the setting of the similarity threshold value. For the behavior-based approach, the threshold value can be chosen from a wide range ($0.5 \sim 0.8$). The behavior-based approach is also affected by the number of behavior records of the current agent. When the number of records is below 3, the behavior-based approach may not show good performance (known as the *Cold Start* problem). This suggests that in this case, we should rely on the threshold-based approach to complement agent preferences.

After using the three approaches to find a set of similar agents for the current agent, each missing preference of the current agent will be complemented by the votes from other similar agents that support this preference. One simple way is that every similar agent is considered to have the same voting weight for its supported preference. Our approach actually assigns different weight to the similar agents according to how similar they are with the current agent (see Section IV-B). We conduct an additional experiment to compare these two methods. As shown in Figure 3(f), the method without considering similar agents' weight results in more failures when a larger number of similar agents are considered. While in our method, such effect is much reduced. Thus, it is better to consider similar agents differently according to their similarity values, when complementing preferences.

B. Scalability of Our Approaches

We then evaluate the scalability of our approaches. This experiment involves 1000 historical agents. And, the number of service attributes varies from 5 to 17 (2 more random ones). As can be seen from Figure 4(a), the runtime of all three approaches increases linearly with the number of service attributes. The runtime of the behavior-based approach increases faster than the other approaches. The clustering-based approaches has the slowest increment in runtime.

We also carry out experiments to observe the runtime of the behavior-based approach with different numbers of historical agents. We run the experiments in three cases where the number of the current agent's behavior records is 1, 2, and 3 respectively. Results are shown in Figure 4(b). We can see that the runtime of the behavior-based approach increases linearly with the number of historical agents when the number of the current agent's behavior records is fixed. It can also be seen that the runtime of this approach also increases with the number of the current agent's records.

In the next experiment, we then evaluate the runtime of the behavior-based approach with the number of the current agent's behavior records ranging from 1 to 5. We conduct this experiment in both the cases of 5 and 15 service attributes respectively. From the results shown in Figure 4(c), the runtime of this approach also increases linearly with the number of the current agent's behavior records.

In the following experiment, we aim to compare the scalability of the three approaches when the number of historical agents increases from 1000 to 7000. Results are shown in Figure 4(d). The clustering-based approach consists of the process of clustering similar agents together. Because this process can be performed off-line, the runtime of this approach is then mainly caused by comparing the CP-net of the current agent with that of every cluster center. Obviously, the runtime of this is nothing compared to the process of computing similarity with every historical agents (possibly a large number). Thus, it can be seen from Figure 4(d) that the runtime of the clustering-based approach is much less that the other two approaches. Also, we can see that the behavior-based approach takes more time than the threshold-based approach.

To sum up, the runtime of our three approaches increases linearly with different varying factors, and thus are scalable. The clustering-based approach is the most scalable. It is affected by the number of service attributes, but not the number of historical agents. The threshold-based approach and the behavior-based approach are affected by both the two factors. The behavior-based approach is also affected by the number of the current agent's behavior records. And, the threshold-based approach is more scalable than the behavior-based approach, which also confirms the arguments in Section IV-A4.

C. Comparison with Collaborative Filtering

Our experimental results in Section V-A confirm that the behavior-based approach is the most effective. In this section, we compare this approach with a classic collaborative filtering approach [9]. As discussed in Section II, our behavior-based approach has many differences with the collaborative filtering approach. However, the aim of both approaches is the same in the sense that they all provide recommendations of services to the current agent, when the behavior records of other historical agents as well as the current agent exist.

In this experiment, we implement the collaborative filtering approach to provide recommendations of the top-k services. The process of finding a set of similar agents as the current agent is the same for both the two approaches. After having the set of similar agents and based on their behavior records, the collaborative filtering approach will just directly generate recommendations of services to the current agent. Our behavior-based approach however will first complement the current agent's missing preferences using the preferences of the similar agents, then use the complemented CP-net of the current agent to generate service recommendations.

Figure 4(e) shows the results of performance comparison. The accuracy of collaborative filtering is less than our behavior-based approach when the similarity threshold varies from 0 to 0.85. When the threshold value exceeds 0.85, the accuracy of our approach appears unusual because only a very few similar agents will be found in this case. The collaborative filtering approach, on another hand, is more robust with respect to the similarity threshold as long as some similar agents can be found. We also test the runtime of the collaborative filtering approach. As can be seen from Figure 4(f), the runtime required for this approach is much longer than the behavior-based approach (see Figures 3(b$\sim$d)). To conclude, our behavior-based approach has the comparable performance as the collaborative filtering approach when the similarity threshold value is set properly. The behavior-based approach also requires much less runtime and thus is more scalable.

VI. Conclusion and Future Work

Our work was focused on complementing agent preferences by proposing a set of approaches. Our approaches have several properties. They are qualitative by representing agent preferences using CP-nets. They are collaborative in the sense that they make use of available preferences of other agents in the system. They are also explicit since they are able to show the complete representation of agent preferences after complementing the missing ones. This property is important as agents have the flexibility in modifying the complemented preferences if they want. Especially, our behavior-based approach transfers implicit agent preferences embedded in agents' behavior of service selection to explicit preferences. We also carried out experiments using a real dataset to evaluate these approaches. Our approaches have been shown to be effective and hold competitive advantages over the clustering filtering approach in terms of the effectiveness of service selection and computation complexity.

For future work, we will investigate how data mining techniques can be applied to complementing agents' incomplete preferences from an available database of agents' behavior, without relying on other agents' preferences (This work is partially supported by JSNSF of China(No.BK2010417)).

References

[1] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient qos-aware service composition," in *Proceedings of the 18th ACM International Conference on World Wide Web*, 2009.

[2] P. Xiong and Y. Fan, "Qos-aware web service selection by a synthetic weight," in *Proceedings of the 4th IEEE International Conference on Fuzzy Systems and Knowledge Discovery*, 2007.

[3] H. Wang, J. Zhang, C. Wan, S. Shao, R. Cohen, J. Xu, and P. Li, "Web service selection for multiple agents with incomplete preferences," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 2010.

[4] C. Boutilier, R. I. Brafman, H. H. Hoos, and D. Poole, "Preference-based constrained optimization with cp-nets," *Computational Intelligence*, vol. 20, no. 2, pp. 137–157, 2004.

[5] S. Lamparter, A. Ankolekar, R. Studer, and S. Grimm, "Preference-based selection of highly configurable web services," in *Proceedings of the 16th ACM International Conference on World Wide Web*, 2007.

[6] J. Doyle and R. H. Thomason, "Background to qualitative decision theory," *AI Magazine*, vol. 20, no. 1, pp. 55–68, 1999.

[7] G. Hines and K. Larson, "Preference elicitation for risky prospects," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent System (AAMAS)*, 2010.

[8] J. M. García, D. Ruiz, A. R. Cortés, and J. A. Parejo, "Qos-aware semantic service selection: An optimization problem," in *Proceedings of the 2008 IEEE Congress on Services*, 2008.

[9] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[10] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. S. Huang, "Supporting similarity queries in mars," in *Proceedings of the 5th ACM International Conference on Multimedia*, 1997.