WCP-Nets: A Weighted Extension to CP-Nets for Web Service Selection

Hongbing Wang[1,⋆], Jie Zhang[2], Wenlong Sun[1], Hongye Song[1], Guibing Guo[2], and Xiang Zhou[1]

[1] School of Computer Science and Engineering, Southeast University, Nanjing, China
[2] School of Computer Engineering, Nanyang Technological University, Singapore
hbw@seu.edu.cn

Abstract. User preference often plays a key role in personalized applications such as web service selection. CP-nets is a compact and intuitive formalism for representing and reasoning with conditional preferences. However, the original CP-nets does not support fine-grained preferences, which results in the inability to compare certain preference combinations (service patterns). In this paper, we propose a weighted extension to CP-nets called WCP-nets by allowing users to specify the relative importance (weights) between attribute values and between attributes. Both linear and nonlinear methods are proposed to adjust the attribute weights when conflicts between users' explicit preferences and their actual behaviors of service selection occur. Experimental results based on two real datasets show that our method can effectively enhance the expressiveness of user preference and select more accurate services than other counterparts.

1 Introduction

User preference often plays a key role in personalized and AI applications [5] such as web service selection [15,20] in order to support automatic decision making [11,10]. Basically, it can be represented in two ways: quantitative ("I prefer Thai Airline at the level of 0.7") or qualitative ("I prefer Qantas Airline to Thai Airline"). In addition, it can be expressed unconditionally ("No matter what time it is, I always prefer Qantas Airline to Thai Airline") or conditionally ("If time is late, I prefer Qantas Airline to Thai Airline"). However, it is generally agreed that users feel more comfortable and natural to express their preferences in a qualitative and conditional manner [4].

CP-nets [4] is a compact and intuitive formalism for representing and reasoning with conditional preferences under the *ceteris paribus* ("all else being equal") semantics. It has attracted much attention in the literature [3,6,9,7,16,17]. However, it suffers from two inherent issues: 1) users are unable to express their fine-grained preferences. More specifically, they cannot specify the level of their preferences between attribute values or between attributes; 2) due to the limited expressiveness [21], many *service patterns*, each of which is defined as a

⋆ This work is supported by NSFC project titled with "System of Systems Software Mechanisms and Methods based on Service Composition" and JSNSF(BK2010417).

C. Liu et al. (Eds.): ICSOC 2012, LNCS 7636, pp. 298–312, 2012.

combination of attribute values for all attributes, are incomparable. It is critical and essential for web service selection to effectively reason and represent user preference. Therefore, the expressiveness of CP-nets and the ability to compare service patterns have important impacts on web service selection. To solve these problems, some approaches attempt to enhance the expressiveness by relaxing the *ceteris paribus* semantics of CP-nets and defining a logical framework [21], or by adding relative importance in statements [6], etc. Some others focus on the comparison of service patterns such as using a utility function [3]. However, none of them can well solve these two problems at the same time.

In this paper, we propose a weighted extension to CP-nets called *WCP-nets* by allowing users to express their preferences in a fine-grained manner. More specifically, user preference can be delineated at multiple levels as described in [22]. For example, an attribute value x_0 can be preferred to another attribute value x_1 equally at level 1, or mildly at level 2, ..., or extremely at the maximum level. In addition, the relative importance (weight) between attributes can be explicitly specified as well, reflecting the extent to which an attribute is more important than another. We also propose a new measurement for the degree of preference of any given service pattern, which can be calculated both in linear or nonlinear methods. In this way, not only the expressiveness of WCP-nets is high, but all service patterns can also be compared. Experimental results based on two real datasets demonstrate that WCP-nets outperforms other approaches. In addition, the results also indicate that linear method can converge faster but nonlinear method achieves better accuracy.

2 Background and Related Work

To begin with, we will introduce the basic principles and the concerned problems of CP-nets. Then we will detail the major variants of CP-nets in the literature proposed to enhance the expressiveness of user preference and the effectiveness in comparisons between service patterns. Finally, we will review several recent studies based on CP-nets or its variants for web service selection.

2.1 CP-Nets

CP-nets [4] is a graphical model for representing and reasoning with conditional preference in a compact, intuitive and structural manner. 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 directed parent nodes $Pa(X_i)$. They are connected by arcs from $Pa(X_i)$ to X_i in the graph. Under the semantics of ceteris paribus ("all else being equal"), the values of X_i is only dependent on the values of $Pa(X_i)$. Thus, a *service pattern* can be defined as a combination of attribute values for all attributes represented in CP-nets, i.e. $sp = x_1x_2 \ldots x_n$, where $x_i \in D(X_i)$ for $i = 1 \ldots n$ represents a specific value on attribute X_i.

In addition, each node X_i is annotated with a CPT denoted by $CPT(X_i)$, which expresses user preferences over 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 $Pa(X_i)$. 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}.

A typical CP-nets is illustrated in Figure 1(a, b). It describes the data storage and access service of a company which consists of three attributes with respect to the quality of service (QoS), namely A : Platform, B : Location and C : Provider. Figure 1(e) shows that each attribute has two specific values. Specifically, data can be stored in either a file system a_1 or a database a_2 which can be located in New York b_1 or Beijing b_2 and can be accessed publicly c_1 or privately c_2. The user 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 the database is chosen for data storage, then the location in New York is preferred to Beijing. In that case, users prefer data to be accessed publicly rather than privately. Note that this example will be used throughout the rest of this paper.

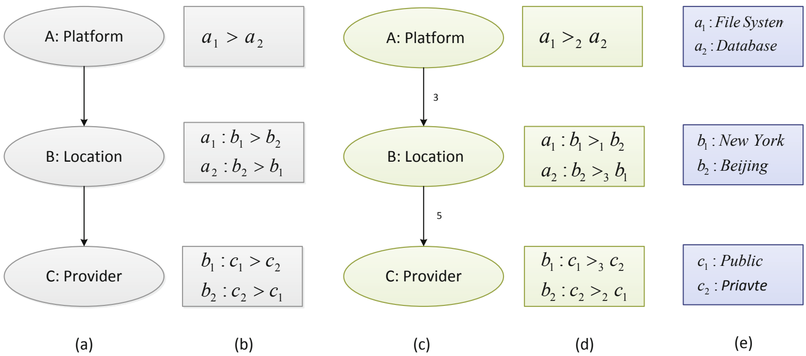


Fig. 1. (a, b) CP-nets; (c, d) WCP-nets; (e) Attribute Values

2.2 Problems of CP-Nets

Although CP-nets is an effective tool to represent and reason with conditional preference, it suffers from two inherent issues. The first is that users are unable to represent more fine-grained preference. For example, in Figure 1(b), the user can only specify that a_1 is preferred to a_2 (i.e. $a_1 \succ a_2$), but not able to indicate to what extent the preference would be. The user may indeed desire to express that a_1 is strongly preferred to a_2.

Another concern is regarding the relative importance between attributes. The dependence relationship in CP-nets merely indicates that parent nodes are more important than their children nodes. This fact usually results in that many service patterns are non-comparable. Take the service patterns $a_2b_2c_2$ and $a_1b_2c_1$ in

Figure 1(a, b) as an example. Although $a_2b_2c_2$ has preferred values on attributes B and C and only violates on attribute A, it is difficult to compare with $a_1b_2c_1$ that has only one preferred value on attribute A but two violations on attributes B and C. The reason is that multiple violations on lower priority (children) attributes may not be preferred to a single violation on a higher priority (parent) attribute.

2.3 Extensions and Variants of CP-Nets

In order to overcome the aforementioned issues, a number of approaches to date have been proposed to extend the original CP-nets in the literature. Some other variations for different purposes (e.g. cyclic preference [9], uncertainty in preference [7]) are not considered in this paper.

Wilson [21] relaxes the ceteris paribus semantics of the CP-nets and proposes a logical framework for expressing conditional preference statements. Specifically, it allows conditional preference statements on the values of an attribute and a set of attributes can vary while interpreting the preference statement. Although this method indeed increases the expressiveness of the CP-nets, it changes the basic reasoning foundations which is different from our proposal.

Boutilier et al. [3] propose a quantitative formalism UCP-nets by enabling users to apply a utility value to each value of attributes other than preference orderings. Service patterns can be compared based on the summed utility value of all attributes. The utility function is a GAI (Generalized Additive Independent) [2] which relies on the structure of CP-nets. Therefore, it is difficult for users to identify a well-suited utility function that can also guarantee the satisfaction of the principles of CP-nets.

Another noticeable extension to CP-nets is proposed by Brafman et al. [6], called Tradeoff CP-net (TCP-nets). It strengthens preference statements by adding relative (qualitative) importance to different attributes. In addition, it maintains the structures and ceteris paribus semantics of CP-nets. Although the expressiveness of TCP-nets is increased to some extent, it is demonstrated in [21] that the expressiveness of TCP-nets is still limited.

In this paper, we propose another significant extension to CP-nets by adding relative importance (weights) between attribute values and between attributes. We name this variant weighted CP-nets, or WCP-nets. The intuition behind is that users may have multiple levels of preference for one state over another [22]. Hence, users are able to express fine-grained preference at multiple preference levels. The concept of *Violation Degree* is introduced to measure the extent to which a service pattern is preferred. In this way, all service patterns can be compared based on this computed value. In addition, we provide two (one linear and another non-linear) methods to flexibly adjust attribute weights such that the measured violation degree well matches users' true preferences.

2.4 CP-Nets in Web Service Selection

There have been a number of studies that model users' qualitative preferences based on CP-nets in order to provide them with personalized web service recommendations. A typical problem for qualitative preference is that users may not express their preferences in detail and some of them could be unspecified. Wang et al. [18] propose that user's missing qualitative preferences could be complemented based on the preferences of similar users, that is, a collaborative filtering method. Wang et al. [19] also indicate that effective web service selection could be done in the absence of complete user preference. Our work does not attempt to elicit more preferences from users, but allow them to express more fine-grained preferences through which web services could be correctly selected.

In addition to CP-nets, other approaches or variants have also been applied to model qualitative preference for web service selection. For example, Santhanam et al. [15] represent user preference by means of TCP-nets through which a set of composite services can be returned. García et al. [8] propose an ontology-based method that transforms user preference modeling to an optimization problem. However, Wang et al. [17] point out that using a qualitative or quantitative approach alone cannot completely handle user preference of web services. Instead, they present user preference by adopting both qualitative and quantitative approaches. In particular, users' qualitative preferences are described by TCP-nets while quantitative preferences are specified by arbitrary positive numbers. Inspired by these studies, we propose to represent user preference in a qualitative way (WCP-nets) and base web service selection on a quantitative method to compute violation degree, i.e. the extent to which a service pattern is preferred.

3 WCP-Nets: A Weighted Extension

In this section, we first describe in detail how to extend CP-nets and how to compare service patterns using the relative importance (weights) based on the concept of violation degree. The weights can be further adjusted linearly or nonlinearly to resolve conflicts between user preference and user behavior. An intuitive example will be presented to exemplify the detailed procedure in the end of this section.

Table 1. The Level of Relative Importance

Level	Definition	Description
1	Equally important	Two values are equally preferred
2	Moderately importance	The first value is mildly preferred to the second
3	Quite importance	The first value is strongly preferred to the second
4	Demonstrably important	The first value is very strongly preferred to the second
5	Extremely important	The first value is extremely preferred to the second

3.1 Adding Weights to Conditional Preference

We adopt the concept of multiple levels of relative importance in [22] but use different level scales, as shown in Table 1. Five-level importance is utilized ranging from level 1 "Equally important" to level 5 "Extremely important". This semantics can be applied to both the relative importance between attribute values and the relative importance between attributes. Formally, we use $\succ_k$ instead of $\succ$ to represent the preference relations in the CPTs, where k refers to the level of relative importance. In addition, the importance level between attributes is assigned and tapped to the arcs in the DDG. The WCP-nets of the previous example is illustrated in Figure 1(c, d). It can be explained that value a_1 is mildly preferred to value a_2 on attribute A which is more important than attribute B at the level of 3. This explanation also holds for the rest.

Since the level of relative importance between attributes is known and the summation of all weights should be 1, the computation of the weights of attributes is trivial. In Figure 1(c), we know $w_B/w_A = 1/3$ and $w_C/w_B = 1/5$, where w_A, w_B, and w_C denote the weight of A, B and C, respectively. Since $w_A + w_B + w_C = 1$, we can easily yield the values of all attribute weights. These computed weights are regarded as the initial attribute weights in our work.

3.2 The Concept of Violation Degree

WCP-nets bases the comparison between service patterns on the concept of violation degree whose definition is given as follows.

Definition 1. ***Violation Degree of an Attribute*** *refers to the level of relative importance if the attribute value selected is not preferred for an attribute of a service pattern according to the corresponding CPT. Formally, it is denoted as $V_X(sp)$, where X represents an attribute of a service pattern sp.*

For the attribute A in Figure 1(d), given the preference $a_1 \succ_2 a_2$, if value a_2 is selected in a service pattern (e.g. $a_2b_2c_2$) rather than value a_1 (i.e. the preference is *violated*), then the violation degree V of attribute A is 2, denoted as $V_A(a_2b_2c_2) = 2$. Similarly, given the service pattern $a_1b_1c_2$, we can get the violation degree of attribute C is 3, i.e. $V_C(a_1b_1c_2) = 3$. Generally, the greater the violation degree of an attribute is, the less preferred is the attribute value.

Definition 2. ***Violation Degree of a Service Pattern*** *refers to the combination of violation degrees of all attributes for a service pattern, taking into consideration the weights of all attributes. Formally, it is denoted as $V(sp)$ and calculated by*

$$V(sp) = F(w_X, V_X(sp)) \tag{1}$$

where F is an aggregation function, taking into account attribute weights w_X and attribute violation degrees $V_X(sp)$.

There could be different methods to define specific functions for F to calculate the violation degree of a service pattern, linearly or nonlinearly. A simple linear method is a weighted summation of violation degrees of all attributes and the attribute weights. And artificial neuron network (ANN) is used as a nonlinear implementation for the aggregation function F. We will discuss it later in detail. Intuitively, a service pattern is more preferred if it has a smaller violation degree.

3.3 Adjusting Initial Attribute Weights

Although users express their preference explicitly, it does not guarantee that what services they select is consist with what are claimed they would like. Conflicts between user behavior and stated preference could occur in real life. O'Sullivan et al. [14] also confirm that explicit preferences are not always consistent with implicit user behaviors in the filed of recommendations. In this paper, we propose to flexibly adjust attribute weights such that the measured violation degree based on users' explicit preferences well matches users' actual behaviors. Two models are utilized to adjust the attribute weights in line with the two methods (linear and nonlinear) to calculate violation degree of a service pattern. In particular, the Lagrangian model [13] is used for linear adjustment and artificial neuron network (ANN) is for the nonlinear. The basic principle is to adjust the attribute weights of a service pattern when it is selected by a user rather than the one with the smallest violation degree. Given different applications and real constrains, other linear and nonlinear methods may be more suitable since we focus on how to adjust attribute weights in general rather than optimization.

3.3.1 Lagrangian Linear Weight Adjustment

We apply the Lagrangian model [13] to adjust the attribute weights of a selected service pattern if it does not have the smallest violation degree. Luenberger and Ye [13] contend that this method is clear in meaning, simple and practical to change the constrained optimization into unconstrained problems.

Assume that sp' is the best service pattern calculated using the initial attribute weights and sp is the service pattern that a user actually selects. Let w'_k be the initial weight of attribute k, and w_k be the adjusted weight of the same attribute. To be expected, users select the service pattern with the smallest violation degree, i.e. $V(sp) \leq V(sp')$. Ideally when the selected service pattern is exactly the computed best service pattern, it can be re-written as

$$\sum_{k=1}^{m} \Big(V_k(sp') - V_k(sp)\Big) w_k = 0 \tag{2}$$

where m is the number of attributes for the service and $\sum_{k=1}^{m} w_k = 1$. The weights that minimize the summed square of weight variations are the optimal. Aiming to obtain the optimal attribute weights, we construct a single objective optimization model as follows:

$$\begin{cases} \min F(w) = \sum_{k=1}^{m} (w_k - w'_k)^2; \\ \sum_{k=1}^{m} \big(V_k(sp') - V_k(sp)\big) w_k = 0; \\ \sum_{k=1}^{m} w_k = 1. \end{cases} \tag{3}$$

The Lagrangian model will linearly aggregate constrains in (3) as

$$\sum_{k=1}^{m}(w_k - w'_k)^2 + \lambda_1 \sum_{k=1}^{m}(V_k(sp') - V_k(sp))w_k + \lambda_2 \sum_{k=1}^{m} w_k = 1 \tag{4}$$

Then take the partial derivative of variables to obtain

$$\begin{cases} \frac{\partial L}{\partial w_1} = 2(w_1 - w'_1) + (V_1(sp\prime) - V_1(sp\prime)) + \lambda_2 = 0 \\ \vdots \\ \frac{\partial L}{\partial w_m} = 2(w_m - w'_m) + (V_m(sp\prime) - V_m(sp)) + \lambda_2 = 0 \\ \sum_{k=1}^{m} \big(V_k(sp') - V_k(sp)\big) w_k = 0 \\ \sum_{k=1}^{m} w_k = 0 \end{cases} \tag{5}$$

Finally, we can determine the optimal weights

$$w_k = w'_k - [\delta_k - \frac{\sum_{n=1}^{m} \delta_n}{m}] * \frac{\sum_{n=1}^{m} \delta_n * w'_n}{\sum_{n=1}^{m} \delta_n^2 - \frac{[\sum_{n=1}^{m} \delta_n]^2}{m}} \tag{6}$$

where $\delta_n = V_n(sp') - V_n(sp)$.

3.3.2 BP-ANN Nonlinear Weight Adjustment

Back propagation (BP) is a supervised learning algorithm which is often used to learn the best weights for the neurons (attributes) in the artificial neural networks (ANNs). ANN is well-suited for learning without a specific mathematical model and able to perform accurate non-linear fit [12]. A typical BP-ANN is shown in Figure 2. The first layer consists of n input units. Each input unit is fully connected to all h hidden units in the middle layer which results in m outputs in the last layer using the same connectivity. One of the advantages of ANN is that it can effectively approximate any complex nonlinear functions which correspond to the nonlinear combination functions in our case.

BP attempts to minimize the *Mean Square Error* (MSE) between the actual outputs and the desired ones. A set of training data is iteratively imported into the network until the learning is converged and the adjusted weights are kept stable. The training data is usually in the form of $< x_1, \ldots, x_n, d_1, \ldots, d_n >$, where x_i represents the initial input and d_i is the desired output.

In our case, the inputs of the network are the violation degrees of attributes of the selected service pattern, and the outputs are those of the calculated best service pattern. The strength of the links between the first and second layers is the initial attribute weights of the selected service pattern. We normalize the attribute weights to be in [0, 1].

Using the initial attribute weights, a number of service patterns, including the computed best service pattern and some random service patterns that are ordered by the violation degrees, will be returned to users and they may make a decision to select the most preferred one. If users choose some random service pattern instead of the expected best one, the adjustment process will be

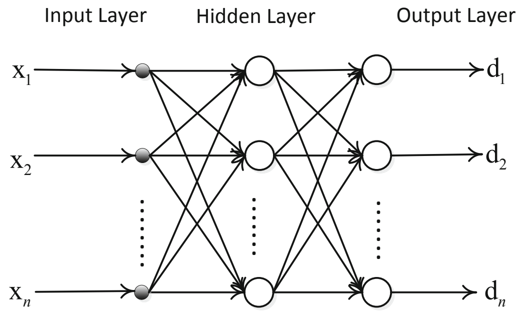


Fig. 2. A Typical BP Artificial Neural Network

activated. Then we assign the violation degree of the best service patten to the selected pattern. After the adjustment, BP algorithm is applied to retrieve another set of service patterns again and allow users to select again. This process will be continued until the best computed service pattern is selected by users, that is, the weight adjustment has converged.

3.4 Example

This section is introduced here to exemplify how WCP-nets works step by step. Since the relative importance between attributes is $w_A/w_B = 3, w_B/w_C = 5$ while the summation of all weights is $w_A + w_B + w_C = 1$, we can determine the initial attribute weights: $w_A = 0.7143, w_B = 0.2381, w_C = 0.0476$. There are eight service patterns, i.e. $(sp_1, sp_2, sp_3, sp_4, sp_5, sp_6, sp_7, sp_8) = (a_1b_1c_1, a_1b_1c_2, a_1b_2c_1, a_1b_2c_2, a_2b_1c_1, a_2b_1c_2, a_2b_2c_1, a_2b_2c_2)$. If we use linear method to calculate the violation degrees of service patterns, the values of them are 0, 0.1428, 0.1190, 0.2381, 2.1429, 2.2857, 1.5238, 1.4286, respectively. Thus the ranking sequence of service patterns is

$$sp_1 > sp_3 > sp_2 > sp_4 > sp_8 > sp_7 > sp_5 > sp_6 \tag{7}$$

To select the proper services, we first find out if there are any web services in the database that are in line with the best preferred service pattern sp_1. If so, we recommend these web services to a user Bill, otherwise we return those that satisfy the next best service pattern sp_3. This process repeats until some services are returned. However, the problem is that in practice the service Bill selects may conflict with his aforementioned preferences due to some reasons like preference drifting over time. Assume his real preference ranking of service patterns is

$$sp_1 > sp_6 > sp_2 > sp_3 > sp_4 > (sp_5 = sp_8) > sp_7 \tag{8}$$

To solve this issue, we need to adjust the initial attribute weights to obtain accurate violation degree of a service pattern. To show the adjustment process, we do not generate web services in the database that meet the description of

sp_1. Now we detail the adjustment process. Considering preference sequence (7), we return 4 web service patterns (here every service pattern responses to a unique web service) to Bill, including the computed best service pattern $sp3$ and three random service patterns (sp_6, sp_4, sp_7). He then selects the most preferred service pattern in the light of real sequence (8), i.e. sp_6 rather than sp_3. Since the conflict occurs, the adjustment process is subsequently activated. Applying the Lagrangian method described in Section 3.3.1 to adjust the weights, we can obtain a new set of attribute weights based on which a new preference sequence is produced as below.

$$sp_1 > sp_2 > sp_4 > sp_3 > sp_7 > sp_8 > sp_5 > sp_6 \tag{9}$$

At this iteration, Bill will select his most preferred service pattern among the retrieved 4 patters, say (sp_2, sp_4, sp_3, sp_7). According to Bill's real preference sequence (8), sp_2 will also be selected, hence there is no need to alter any attribute weights. We may continue this adjustment process to judge whether sp_2 is the most preferred available service pattern.

An alternative weight adjustment method is nonlinear and conducted by BP-ANN. The first step is to initialize a BP-ANN. Specifically, the violation degrees and weights of attributes of a service pattern will be imported into the network. And the eight outputs derived from the network will be ordered. Suppose after a few iterations of training, the violation degrees of eight service patterns are (113.5641, 117.3781, 92.0738, 108.8373, 49.4641, 87.3097, 58.6806, 143.2908). The best computed service pattern sp_5 together with three randomly selected patterns sp_6, sp_8, sp_2 are returned for further selection. According to sequence (8), Bill is likely to select sp_6 and hence the conflict happens. For weight adjustment, we assign the violation degree of sp_5 to sp_6. These new values of four service patterns will be regarded as the desired outputs for next iterative training. We continue this adjustment process until the attribute weights are stable.

4 Experimental Validation

The major concern we would like to verify for WCP-nets is threefold: 1) how good it is to distinguish service patterns compared with CP-nets and TCP-nets[1]? 2) how accurate the retrieved service patterns would be? 3) what is the distinction between linear and nonlinear weight adjustments?

4.1 Data Acquisition

For the experiments, we use two real datasets: *Adult*[2] and *QWS* [1]. The former is obtained form the UCI Machine Learning Repository, consisting of 32,561 records. Each record is regarded hereafter as a concrete dating service that contains 14 attributes. The latter contains 2507 real web services which stem from

[1] UCP-nets is excluded in our experiments because it is essentially a quantitative approach for CP-nets rather than a CP-nets extension.

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

public sources on the web including *Universal Description, Discovery, and Integration* (UDDI) registries, search engines, and service portals. Each web service contains nine attributes.All experiments are conducted using an IBM server with 8 CPUs of 2.13 GHz and a RAM of 16 GB.

4.2 Performance Analysis

The performance of WCP-nets is measured by the percent of comparable service patterns (CSPs) and the accuracy of returned web services. The difference between linear and nonlinear weight adjustment methods is also distinguished.

4.2.1 The Percent of Comparable Service Patterns

The first concern is the ability of WCP-nets in comparing different service patterns (CSPs) relative to CP-nets and TCP-nets. Specifically, for each experiment, we vary the number of attributes from 3 to 11 with step 2 and record the percents of service patterns that are comparable using three different methods. We conduct in total four experiments where the number of attribute values is taken from $\{2, 4, 6, 8\}$, respectively. In each case, 1000 CP-nets are randomly generated (for both dependency graphs and CPTs) to represent users' preferences. Based on CP-nets, TCP-nets is constructed by adding relative qualitative importance, and WCP-nets is built by adding relative quantitative importance between attributes. Each experiment is executed 1000 times and the average of the percent of CSPs is computed. The results are illustrated in Figure 3, where (a) - (d) represents the results when the number of attribute values is 2, 4, 6, 8 respectively.

Consistent results are obtained in four experiments, showing that WCP-nets outperforms CP-nets and TCP-nets in comparing different service patterns. In particular, since the relative quantitative importance between attributes is added to the CP-nets, the expressiveness of TCP-nets is better than CP-nets. But there are still a number of service patterns that cannot be comparable. On the contrary, WCP-nets are always able to compare all services patterns in terms of the computed violation degree.

4.2.2 The Accuracy of Retrieved Web Services

Another batch of experiments are conducted to investigate the accuracy of the web services (WSs) that are retrieved by three different methods, i.e. WCP-nets, TCP-nets and CP-nets. The experiments are based on the aforementioned two real datasets. Specifically, we randomly choose 3, 5, 7, 9, 11 attributes from the dataset and apply three methods to model user preference and retrieve the suitable WSs as required. A pre-processing is utilized to convert continuous attribute value to three sections. For example, the value of the attribute *Response time* in QWS is continuous from 0 to the maximum 4207.5ms which is then uniformly segmented into three parts. Finally, the three parts are symbolized as three attribute values of *Response time*. CP-nets is generated randomly as well as TCP-nets and WCP-nets. Generally, the best WS pattern is retrieved if it

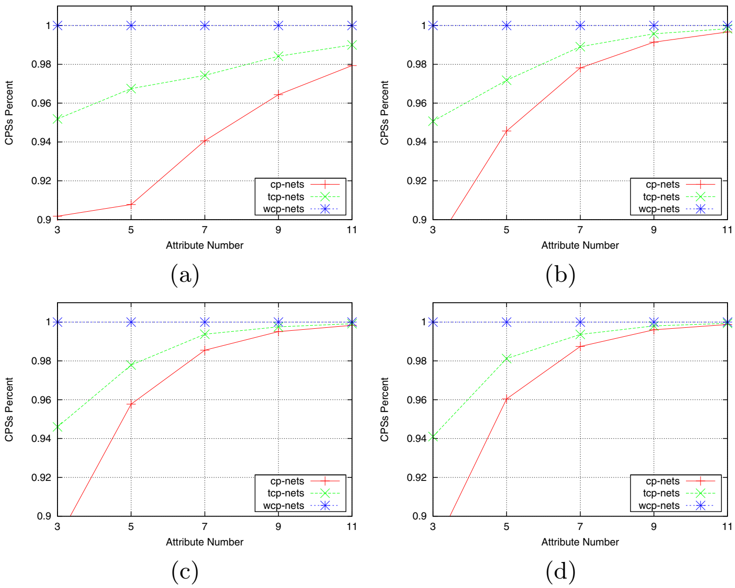


Fig. 3. The Percent of Comparable Service Patterns (CSPs)

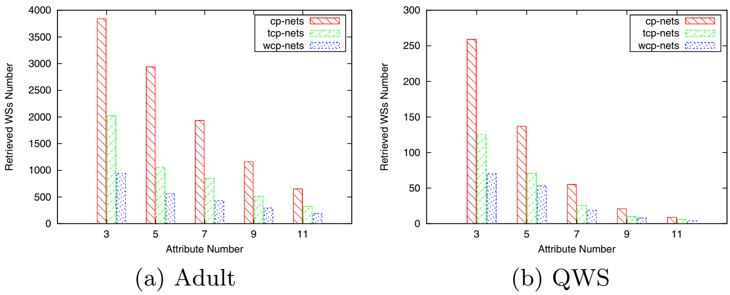


Fig. 4. The Accuracy of the Retrieved Web Services (WSs)

meets user preference expressed in (W, T)CP-nets. Otherwise, the second best WS pattern will be returned. All experiments are repeated 1000 times and the results of the average performance are delineated in Figure 4.

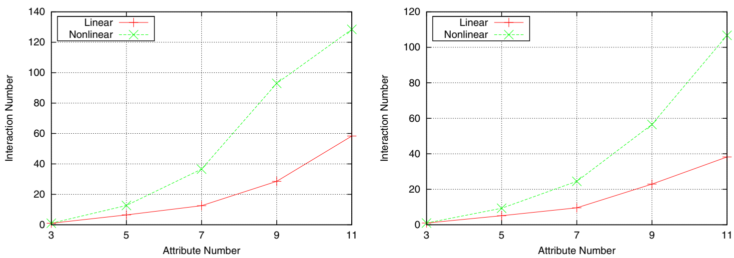
We count the number of retrieved WSs as the measurement of accuracy. The intuition is that the more accurate user preference is, the less number of retrieved WSs is. Clearly, TCP-nets show its strength relative to CP-nets whereas WCP-nets achieves the best accuracy. Theoretically, TCP-nets covers more aspects that are unknown to CP-nets, while WCP-nets expresses not only more but also fine-grained user preference that is not available in the others.

4.2.3 Linear vs. Nonlinear Weight Adjustments

The purpose of this subsection is to investigate the distinctions between linear and nonlinear weight adjustment methods and give readers guides on how to select suitable adjustment method for their own applications. The comparison will be focused on the efficiency and effectiveness for weight adjustments.

The experiments are based on the real datasets. A set of WCP-nets with 3, 5, 7, 9, 11 attributes are generated randomly, and 8 services are randomly selected in each time from the real datasets. The same preprocessing is used for continuous attribute value as Section 4.2.1. The adjustment of attribute weights is activated when conflicts occur, i.e. users select a random service rather than the expected one with minimum violation degree. We continue the adjustment process for at least 30 times even when users do select the computed best service. The purpose is to reduce the selection due to chance and avoid local optimal values. For the nonlinear method, three hidden layers are used in the BP-ANN. The violation degrees of attribute values are used as inputs and the weights of attributes as the weights of inputs. The number of neurons in hidden layer is empirically set 75% of the number of inputs. The training function we utilize is *traingd*, the transfer function between hidden layers is *tansig* and the transfer function of output layer is *purelin*. All experiments is repeated 1000 times and the average performance is adopted.

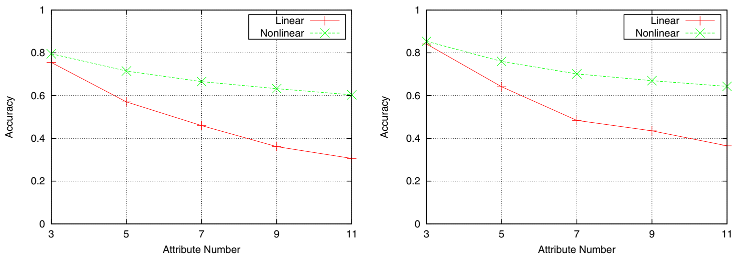
The efficiency is measured as the number of interactions required before convergence. The effectiveness is reflected by the accuracy of the retrieved WSs using different methods. A web service will be labeled *accurate* if it meets user preference which is simulated by a random function. A set of accurate WSs are selected as the benchmark. Hence the accuracy is computed as the percentage of retrieved WSs over the benchmark. The results shown in Figures 5 and 6 represent the efficiency and effectiveness, respectively.



(a) Adult (b) QWS

Fig. 5. The Efficiency of Linear and Nonlinear Methods

Figure 5 shows that the linear method consistently converges much faster than the nonlinear one as the number of interactions required by the former is greatly less than the latter. However, Figure 6 indicates that the latter achieves much better accuracy than the former. A conclusion can be drawn that, for those who require fast convergence, the linear method is more preferred and for those who seek best accuracy, the nonlinear method should be adopted.



(a) Adult (b) QWS

Fig. 6. The Accuracy of Linear and Nonlinear Methods

5 Conclusion

We proposed a weighted extension to CP-nets called WCP-nets, aiming to solve the issues of CP-nets described in Section 2.2. More specifically, the relative importance (weights) between attribute values and that between attributes were added to the original CP-nets to allow users to express more fine-grained preference. The concept of violation degree was introduced to measure the extent to which a service pattern is preferred. Both linear and nonlinear methods were presented to adjust users' initial weights when conflicts between stated preference and actual choices occur. Experiments on two real datasets were conducted and the results showed that our method can not only increase the expressiveness of user preference, but also select more accurate services than other counterparts.

References

1. Al-Masri, E., Mahmoud, Q.: Discovering the best web service. In: Proceedings of the 16th International Conference on World Wide Web, pp. 1257–1258 (2007)
2. Bacchus, F., Grove, A.: Graphical models for preference and utility. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 3–10. Morgan Kaufmann Publishers Inc. (1995)
3. Boutilier, C., Bacchus, F., Brafman, R.: Ucp-networks: A directed graphical representation of conditional utilities. In: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, pp. 56–64 (2001)
4. Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., Poole, D.: Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. J. Artif. Intell. Res. (JAIR) 21, 135–191 (2004)
5. Brafman, R., Domshlak, C.: Preference handling-an introductory tutorial. AI Magazine 30(1), 58 (2009)
6. Brafman, R., Domshlak, C., Shimony, S.: On graphical modeling of preference and importance. Journal of Artificial Intelligence Research 25(1), 389–424 (2006)
7. Châtel, P., Truck, I., Malenfant, J., et al.: Lcp-nets: A linguistic approach for non-functional preferences in a semantic soa environment. Journal of Universal Computer Science 16(1), 198–217 (2010)

8. García, J., Ruiz, D., Ruiz-Cortés, A., Parejo, J.: Qos-aware semantic service selection: An optimization problem. In: Proceedings of 2008 IEEE Congress on Services-Part I, pp. 384–388 (2008)
9. Gavanelli, M., Pini, M.: Fcp-nets: extending constrained cp-nets with objective functions. In: Constraint Solving and Constraint Logic Programming, ERCIM (2008)
10. Koriche, F., Zanuttini, B.: Learning conditional preference networks. Artificial Intelligence 174(11), 685–703 (2010)
11. Lamparter, S., Ankolekar, A., Studer, R., Grimm, S.: Preference-based selection of highly configurable web services. In: Proceedings of the 16th International Conference on World Wide Web, pp. 1013–1022. ACM (2007)
12. Liu, J., Chang, H., Hsu, T., Ruan, X.: Prediction of the flow stress of high-speed steel during hot deformation using a bp artificial neural network. Journal of Materials Processing Technology 103(2), 200–205 (2000)
13. Luenberger, D., Ye, Y.: Linear and nonlinear programming, vol. 116. Springer (2008)
14. O'Sullivan, D., Smyth, B., Wilson, D., Mcdonald, K., Smeaton, A.: Improving the quality of the personalized electronic program guide. User Modeling and User-Adapted Interaction 14(1), 5–36 (2004)
15. Santhanam, G.R., Basu, S., Honavar, V.G.: `TCP−Compose*` – A TCP-Net Based Algorithm for Efficient Composition of Web Services Using Qualitative Preferences. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 453–467. Springer, Heidelberg (2008)
16. Sun, X., Liu, J.: 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, pp. 1908–1911 (2010)
17. Wang, H., Liu, W.: 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, pp. 404–411 (2011)
18. Wang, H., Zhang, J., Tang, Y., Shao, S.: 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, pp. 51–58 (2011)
19. Wang, H., Zhang, J., Wan, C., Shao, S., Cohen, R., Xu, J., Li, P.: Web service selection for multiple agents with incomplete preferences. In: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT, pp. 565–572 (2010)
20. Wang, H., Zhou, X., Zhou, X., Liu, W., Li, W., Bouguettaya, A.: Adaptive Service Composition Based on Reinforcement Learning. In: Maglio, P.P., Weske, M., Yang, J., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6470, pp. 92–107. Springer, Heidelberg (2010)
21. Wilson, N.: Extending cp-nets with stronger conditional preference statements. In: Proceedings of the National Conference on Artificial Intelligence, pp. 735–741 (2004)
22. Xu, H., Hipel, K., Marc Kilgour, D.: Multiple levels of preference in interactive strategic decisions. Discrete Applied Mathematics 157(15), 3300–3313 (2009)