International Journal of Innovative Computing, Information and Control
Volume 13, Number 4, August 2017
 ISSN 1349-4198
pp. 1169–1185

A HYBRID RECOMMENDATION METHOD AND DEVELOPMENT FRAMEWORK OF USER INTERFACE PATTERNS BASED ON HYPERGRAPH THEORY

Rui Chen[1], Qingyi Hua[1,*], Wei Jia[1], Minjun Zhang[1], Xiang Ji[1]
Juanni Li[1], Bo Wang[2] and Jianxin Liu[3]

[1]School of Information Science and Technology
Northwest University
Guodu University Avenue, Chang'an District, Xi'an 710127, P. R. China
nwuchenrui@126.com; *Corresponding author: huaqy@nwu.edu.cn

[2]School of Computer
Xi'an University of Posts and Telecommunications
No. 563, Chang'an Street, Xi'an 710121, P. R. China
xiyouwangbo@163.com

[3]Department of Mathematics
Simon Fraser University
V5A 1S6, BC, Canada
jianxinl@sfu.ca

Received January 2017; revised April 2017

Abstract. *User interface patterns provide good support for the development of usable user interface. However, with the increase of patterns, it is difficult for users to find suitable patterns for usable interface design. Therefore, it is an urgent problem for users to recommend appropriate patterns in the process of designing interface. In this paper, we firstly analyze the advantages and disadvantages of various recommendation algorithms, study the characteristics of task models and interface patterns, and then propose a hybrid recommendation method and development framework of the interface patterns based on hypergraph theory. Afterwards, according to task analysis, we tag the task attributes, and use hypergraph theory to construct users, tasks and patterns as hypergraph models with the historical behavioral data of users' using patterns. Then we divide and cluster the hypergraph models by weight similarity method proposed in this paper, and calculate the similarity between each pattern and task in the cluster to recommend a list of patterns so that the users can quickly and accurately select the appropriate patterns for interface development. Finally, the prototype system is developed and evaluated. The results show that the proposed method can improve not only the usability of the system, but also the development efficiency.*
Keywords: User interface pattern, Hypergraph theory, Task model, Spectral clustering, Collaborative filtering, Vector space model

1. **Introduction.** With the emergence of a variety of interactive devices, more and more people begin to communicate via mobile devices. Shortening the development circle of application software and improving the usability of interface are major challenges for developers. Statistics suggest that the workload of the interface development accounts for 40% $\sim$ 60% of the entire system development [1,2]. In order to develop usable interface, scholars have carried out exploration for many years, proving that the user interface pattern is a feasible scheme for fast interface design. The user interface pattern includes functionality and usability features, and can greatly improve the efficiency of interface

development. The interface pattern can be used to guide the developers to design interface, and make up for the deficiency of reusability based on model-driven development.

In order to generate user interface automatically or semiautomatically, many researches have been done on the application of interface patterns and the structural definition of interface patterns in recent years. Ahmed and Ashraf [2] applied the idea of interface patterns to the process of task modeling, dialog control modeling, presentation modeling and layout management modeling. Nilsson [44] provided a detailed interface design pattern for the screen space, interaction mechanism and other issues in the mobile user interface, and validated the effectiveness of the proposed patterns. Jenifer Tidwell, Martijn van Welie, and Douglas van Duyne respectively classified the patterns, following a variety of pattern language markup languages (PLML) such as PLML, Extended Pattern Language Markup Language (PLMLx), and eXtended Pattern Language Markup Language (XPLML) [4-6].

However, these pattern libraries and description languages are all described by nature language, which are the summary of experiences obtained from expert; thus the user interface is only developed manually by programmers. For those who are not familiar with interface patterns, it is undoubtedly a waste of time to search them in a large number of patterns, which will reduce the efficiency of interface development. In the task modeling phase, selecting the appropriate pattern for each task by automatic system can reduce the user's searching time, so as to improve the efficiency of interface development.

It is a great challenge for the interface developers to find out an appropriate pattern from large numbers of patterns to satisfy the user's preference. Recommendation algorithm is one of the most important ways to solve this problem by modeling the user's historical behavior, which can provide the user with the patterns matching the tasks.

The current recommendation methods include content-based recommendation, collaborative filtering and hybrid recommendation, in which term frequency-inverse document frequency (TF-IDF), k-means clustering, k-Nearest Neighbor (KNN), and Support Vector Machine (SVM) are common techniques. However, these techniques only focus on the relationship among items, without taking user relationship into consideration or only consider the relationship between the user and the item, ignoring the ability to describe the relationship among three or more objects [8,9]. Lin presented a complete three-part diagram model that accurately reflected the user's interest and implemented recommendations for items and tags. However, this model only considered a single relationship between every two nodes but not other relational information, resulting in a loss of information, and low recommended accuracy [11]. Gu introduced the attribute objects of users, news articles and news content into the hypergraph model, and proposed a news recommendation method based on hypergraph model [12].

The hypergraph, which is the generalization of general graph, is used to study the multivariate relations among the elements in a finite set [10]. Hypergraph is widely used in image processing, machine learning, data mining, system modeling and other fields, in which good results have been achieved [10-12].

With the characteristics of hypergraph, this paper proposes a hybrid recommendation method and development framework of interface patterns by hypergraph describing the relationship among user, task and pattern. The combination of vector space model and hypergraphs clustering is used to recommend their preferred patterns for the users, which improves the efficiency of interface development. Finally, a prototype system is provided to verify the feasibility and effectiveness of the proposed method in this paper. Experiments show that the system prototype improves the efficiency of interface development and achieves the interface usability.

2. Related Work.

2.1. User interface patterns. The concept of user interface patterns, also known as HCI patterns, was originated from concept in architecture proposed by Christopher Alexander, and later was applied to the fields of software development and human-computer interaction. It is a summary and knowledge representation of a long-term practical experience of developers, a structured description of an invariant solution to recurring problems in the context, and a more mature solution to the particular problems [15].

After a long-term practice, people have summarized a variety of valuable user interface pattern libraries, such as Jenifer Tidwell, Martijn van Welie, and Yahoo, which provided a sketch, an example and an effective guidance for developers to develop a usable interface, and speed up the interface development efficiency [16,17]. In order to facilitate the selection and retrieval of patterns, Computer-Human Interaction (CHI) Conference held in 2003 proposed a unified model description and instruction. Then, Pattern Language Markup Language (PLML) was produced. Initially, 16 attribute elements were used to describe the patterns, followed by the emergence of PLMLx, XPLML [4-6]. These pattern markup languages divided the patterns according to their name, problem, solution, use when, how, and so on, which were described in unstructured form, such as text, and sketches. These non-formal expressions are difficult to apply interface patterns to interface development tools.

Jenifer Tidwell divided 125 patterns into 11 categories and every pattern was described with 8 attributes. Martijn van Welie divided 131 patterns into 15 categories and described them with 9 attributes. Douglas van Duyne devided 107 kinds of web-related patterns into 13 categories and described them with seven attributes [16-19].

2.2. Recommendation method. The traditional recommendation algorithms can be classified into three categories: content-based recommendation algorithm, collaborative filtering recommendation algorithm and hybrid recommendation algorithm.

The content-based recommendation algorithm generated recommendation results by matching the similarity between the user's preference and the item description document to be predicted, which results in over-concentration of recommendation results, a lack of diversity, and a cold start problem for new items. Users are easily confined to the previous similar recommendation results, and impossible to find out the potential interests of users.

Collaborative filtering recommendation algorithm is classified into memory-based method and model-based method. Memory-based method is classified into user-based collaborative filtering and item-based collaborative filtering. User-based collaborative filtering algorithm recommends items to current users based on the interest records of users sharing the same interests with current users. Item-based collaborative filtering algorithm uses items similar to the current items to recommend them to current users. The former needs to calculate the similarity between the users, and the latter needs to calculate the similarity between items and recommend to current users. Model-based collaborative filtering algorithm regards the scoring matrix as the training data, and uses the method of machine learning to obtain the complex pattern to build the user and the item model. Model-based collaborative filtering methods include Bayesian networks, clustering-based, semantic-based and graph-based collaborative filtering algorithms [12,26-29].

Although the basic idea of memory-based collaborative filtering algorithm is clear and easy to achieve, and the effect is good, it will cause a huge burden to the system when the number of users and items increase dramatically. Memory-based collaborative filtering algorithm shows scalability and cold start problems. Model-based method can solve the problem of data sparsity and scalability to some extent, and it has high recommendation precision and strong interpretability, but the model is expensive to build, its accuracy and scalability are difficult to coexist with each other, and reducing dimension will sacrifice some of the data.

Hybrid recommendation is a combination of a number of algorithms, through mixing to achieve the purpose of avoiding weaknesses, so that the recommended results are more accurate to meet the user's interests. The hybrid method solves the cold start problem, improves the quality of recommendation, and reduces the limitations of different algorithms, while its disadvantage is that it is more complex, costly, and requires additional data.

2.3. **Hypergraph theory.** Hypergraph is the generalization of general graph, and its edges can connect multiple vertices. The related concepts of hypergraph are defined as follows [10-14].

Definition 2.1. *A hypergraph is represented as a pair: $G = (V, E)$, where $V = \{x_1, x_2, \dots, x_n\}$ is a finite set of objects, called the vertex set; $E = \{e_1, e_2, \dots, e_m\}$ is the set of hyperedges, and e_i is called the hyperedge.*

The set E of the hyperedges is also denoted by $E = (e_i)_{i\in I}$, where e_i is the first i hyperedge of the set E, and I is the index of edges in E.

Definition 2.2. *The number of vertices in hypergraph $H = (V, E)$ is called cardinality of vertices, denoted as $|V|$. The number of vertices contained in each hyperedge, is called the degree of hyperedge, and is denoted by $d(e)$ or $|e|$.*

Definition 2.3. *$\forall i \in I$, if $|e_i| = k$, then H is called homogeneous hypergraph. When $k = 2$, H is degraded to a general graph.*

Definition 2.4. *Let $H = (V, E)$ be a hypergraph, if $e_i \subseteq e_j => i = j$, that is $\forall e$, $e' \in E$, $e \neq e'$, there is $e \backslash e' \neq \emptyset$, then H is a simple hypergraph or an irreducible hypergraph. There are no duplicate hyperedges in simple hypergraphs. Let $H = (V, E)$ and $H' = (V', E')$ be hypergraphs, if $E' \subset E$, then H' is a partial hypergraph.*

The hypergraphs generally refer to simple hypergraphs in this paper.

Definition 2.5. *If $V' \subseteq V$, and $\forall e_j \in E'$, then $H = (V', E' = (e_j)_{j\in J})$, $J \subseteq I$, H' is called a subhypergraph of H.*

Definition 2.6. *Let $A = a_{ij}$ $(i = 1, 2, \dots, n;\ j = 1, 2, \dots, m)$ be adjacency matrix of a hypergraph, where n is the number of vertices and m is the number of hyperedges, that is (see Equation (1))*

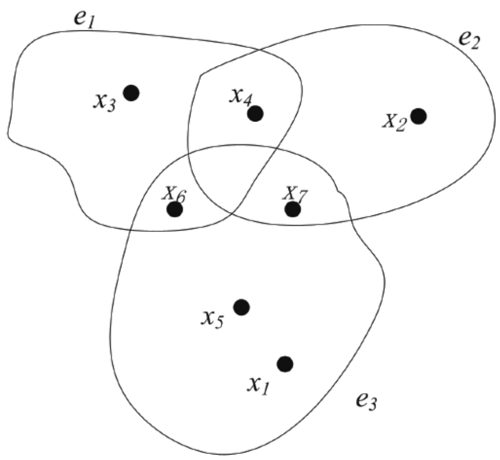
$$a_{ij} = \begin{cases} 1 & v_i \in e_j \\ 0 & v_i \notin e_j \end{cases} \tag{1}$$

The hypergraph $H = (V, E)$ and the adjacency matrix are shown in Figure 1.

The hypergraph H contains seven vertices $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ and three hyperedges $\{e_1, e_2, e_3\}$, where x_3, x_4, $x_6 \in e_1$, x_2, x_4, $x_7 \in e_2$, x_1, x_5, x_6, $x_7 \in e_3$.

Definition 2.7. *The degree of the hyperedge is defined as the number of elements in the hyperedge, that is, for any edge e, its degree $\delta(e) = |e| = \sum_{v\in V} h(v, e)$, where in the adjacency matrix A, if $v \in e$, then $h(v, e) = 1$. Otherwise, $h(v, e) = 0$.*

3. **Recommendation Method and Framework of Interface Patterns Based on VSM-HC.** Ahmed Seffah proposed the method of constructing task model, dialog model and layout management model by using the interface patterns in Concur Task Trees (CTT) model. However, the selection of patterns also requires users with rich interface



$$\begin{array}{c c} & \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{array} & \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

FIGURE 1. Hypergraph and adjacency matrix

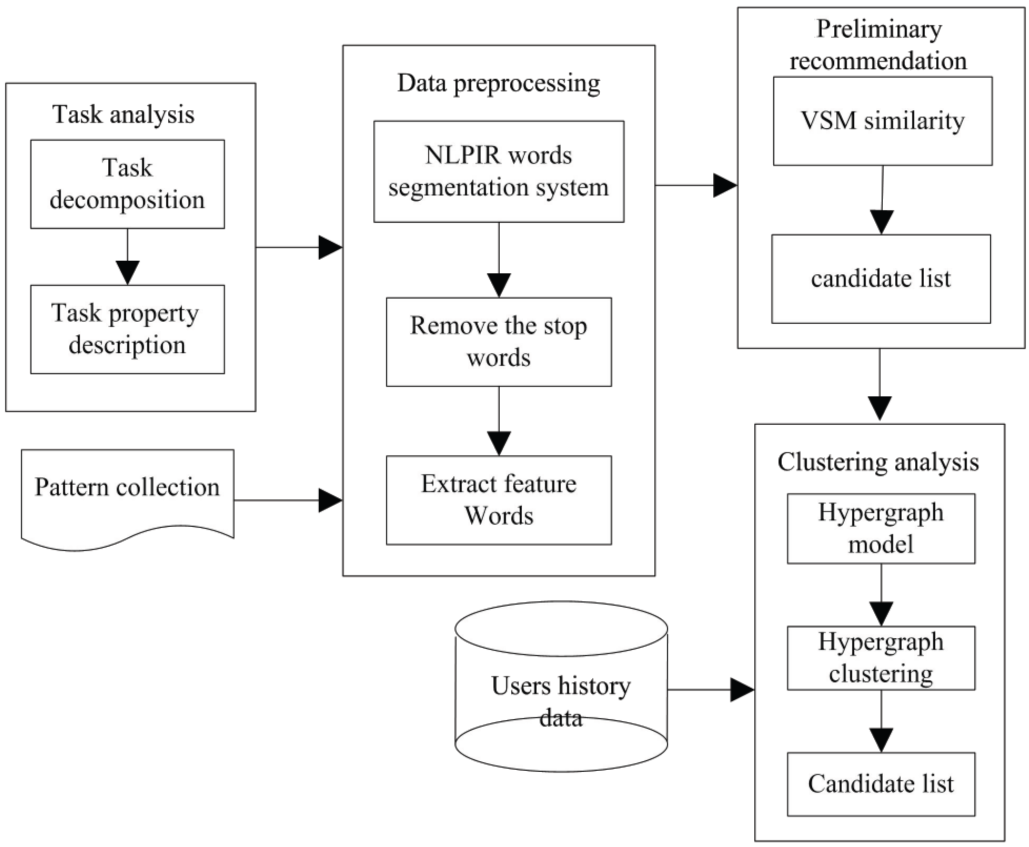


FIGURE 2. Recommendation framework of interface patterns based on VSM-HC

development experience to find manually, which greatly reduces the efficiency of interface development. In order to improve the efficiency of interface development and make the developers get rid of tedious work, we propose the recommendation method of interface patterns based on Vector Space Model-Hypergraph Clustering (VSM-HC) in this paper, not only to match patterns with the task, but also to meet the needs of different users.

3.1. **Recommendation framework of interface patterns.** The recommendation framework of interface patterns is shown in Figure 2.

In Figure 2, after describing the tasks with tools such as CTT, the system respectively preprocesses the attribute sets of tasks and patterns, such as segment words, removes the stop words and extracts the characteristic words; the similarity between tasks and patterns is calculated according to the vector space model, and the preliminary candidate list is obtained according to the similarity degree from large to small. This recommendation list can solve the user's cold start problem without his historical data records and personal needs. Furthermore, the hypergraphs model among users, tasks and patterns are established based on hypergraph theory according to the users' history of selecting patterns for tasks, then the hypergraph is separated by spectral clustering algorithm, and each hypergraph is clustered by k-means. Finally, the similarity between tasks and patterns is calculated using the vector space model, and the list of recommendation is got which can meet the users' interest according to the similarity.

3.2. **Task attribute annotation.** In order to recommend the appropriate patterns to the appropriate task and facilitate the transition of the task model to the abstract user interface, users need to label each task attribute in the process of task analysis.

The labeling of task attributes needs to consider the characteristics of the pattern to improve the accuracy of the recommendation. The pattern sets were classified by the scholars such as Jenifer Tidwell, Martijn van Welie and Douglas van Duyne, followed by pattern markup languages, such as PLML, PLMLx, and XPLML, most of which have described patterns from pattern ID, name, alias, illustration, problem, context, forces, and so on [20]. In order to quickly and accurately recommend the appropriate patterns to the users, the description of the tasks is also according to the methods of patterns classification, and the task will be marked in accordance with the method of labeling. In this paper, each task is described as (task ID, name, type, function, context, program, solution).

3.3. **The initial recommendation of interface patterns.** After data preprocessing, this paper firstly calculates the similarity between patterns and tasks, so as to recommend appropriate patterns to the user.

The attribute set $(t_{q1}, t_{q2}, t_{q3}, \ldots, t_{qn})$ of each task constitutes the query vector Q, and the pattern set $(p_{j1}, p_{j2}, p_{j3}, \ldots, p_{jn})$ constitutes the document vector D_i, the similarity [30,31] of which is defined as follows (see Equation (2)):

$$sim(Q_q, D_i) = \frac{\sum\limits_{j=1}^{n} w_{qj} \bullet w_{ij}}{\sqrt{\sum\limits_{j=1}^{n} (w_{qj})^2} \bullet \sqrt{\sum\limits_{j=1}^{n} (w_{ij})^2}} \tag{2}$$

where n is the dimension of the query vector and the document vector, and w_{ij} and w_{qj} are the weights of the items.

Here, the weight w_{ij} is defined as follows (see Equation (3)):

$$w_{ij} = tf_{ij} * idf_i = \frac{tf_{ij} \bullet \lg\left(\frac{N}{df_i} + 0.01\right)}{\sum\limits_{k_i \in D_j} \left[tf_{ij} \bullet \lg\left(\frac{N}{df_i} + 0.01\right)\right]^2} \tag{3}$$

The term frequency tf_{ij} indicates the times of the index entry k_i appearing in the document D_j. The larger tf_{ij} is, the more important the k_i is to the D_j. The document frequency df_i represents the number of documents that contain the index item k_i in the whole document set, and the higher the df_i is, the greater universality of k_i is, the lower to measure the similarity between documents according to index terms.

3.4. **Hypergraph model and weight calculation.** Gu et al. [12,34] put hypergraph theory into the news recommendation process, and the precision of the recommendation has been greatly improved. This paper is inspired by hypergraph theory, spectral clustering and vector space model, and establishes the hypergraph model of interface patterns, task model and user relationship. The hypergraph theory and vector space model are employed in the recommended process of interface patterns, and a calculation method of hyperedge weight is proposed. Interface patterns recommendation mainly involves three objects: user, pattern and task. In this paper, u is used to denote the user, p is the pattern, and t is the task.

Definition 3.1. *Let V be the set of nodes, U be the set of users, P be the set of patterns, T be the set of tasks, and node set V be the set of nodes of the proposed hypergraphic model, denoted by $V = U \cup P \cup T$.*

Definition 3.2. *The hypergraph model is defined as $G = (V, E, w)$, and w is the weight of the hyperedge. The hyperedge weight is used to express the hyperedge strength of T, P and U. The larger the value is, the stronger the relationship is. The weights need to consider the relationships between T and P, U and P. The hyperedge weight is denoted as (see Equation (4)):*

$$w = sim(p_j, t_k) + \frac{|sel(u_i, p_j, t_k)|}{|sel(u_i, p_j)} \tag{4}$$

where $sim(p_j, t_k)$ is the similarity between the pattern and the task, $sel(u_i, p_j, t_k)$ denotes the times that the user u_i selects the pattern p_j in describing the task t_k and $sel(u_i, p_j)$ represents the total times that user selects the patterns p_j for describing all tasks.

In this way, we can get the adjacency matrix H and the diagonal matrix D_v of the vertex degree and the diagonal matrix D_e of the super edge.

The hypergraph model among users, tasks, and patterns is shown in Figure 3.

As can be seen from Figure 3, the nodes associated with the hyperedge are not confined to two nodes, for example, e_1 associates with 3 nodes and e_2 associates with 4 nodes.

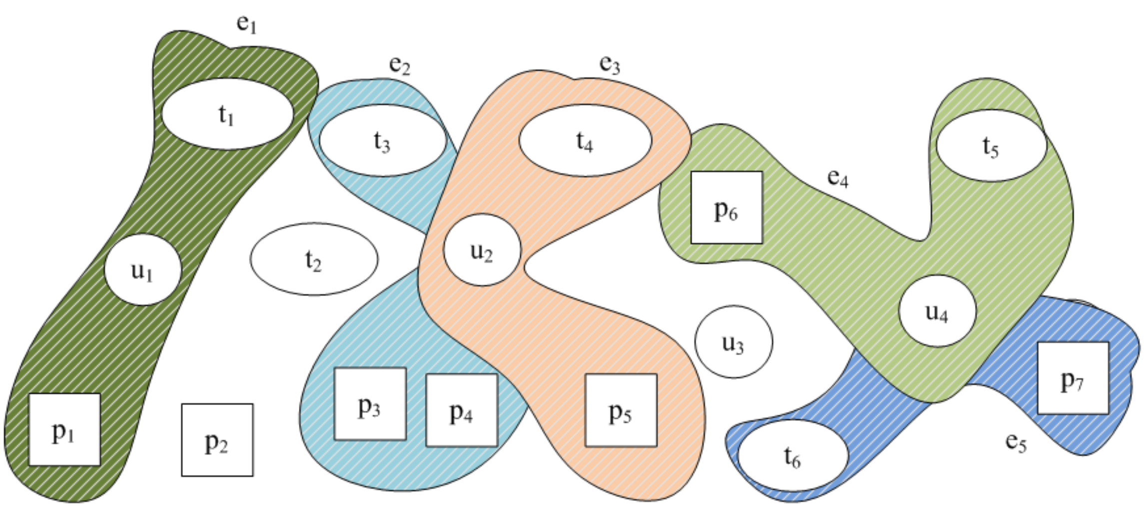


FIGURE 3. A hypergraph model consisting of users, tasks, and pattern

3.5. **The algorithm of vector space model-hypergraph clustering.** Hypergraphs are also represented by matrices like general graphs, so that the operations of the hypergraphs become the operations of the general graphs. It is a kind of clustering problem to recommend appropriate patterns for tasks and users. The problem of graphs clustering is to find an optimal segmentation, which divides an undirected weighted graph into two or more optimal subgraphs. Therefore, the clustering problem of graphs is to find the optimal segmentation.

Let X and Y be two subsets of a hypergraph, and the optimal segmentation is to minimize the sum of the weights of the edge to be cut. The edge with the larger weight will not be cut off, which means that the following objective function needs to be minimized (see Equation (5)):

$$cut(X,Y) = \sum_{i \in X, j \in Y} e_{ij} \tag{5}$$

However, the minimizing cut usually leads to a poor segmentation [34]. In order to obtain a reasonable size for each hypergraph, the objective function is as large as possible for $A_1, A_2, \ldots, A_k$. The improved objective function is (see Equation (6)):

$$RatioCut(A_1, A_2, A_3, \ldots, A_k) = \sum_{i=1}^{k} \frac{cut\left(A_i, \overline{A}_i\right)}{vol|A_i|} \tag{6}$$

where $vol(A) = \sum_{i \in A} w_{ij}$. According to the properties of the Laplace matrix [35], the following inference can be made (see Equation (7)):

$$f'Lf = |V| RatioCut(X,Y) \tag{7}$$

where $|V|$ is a constant, which means that the Laplacian matrix L is equivalent to the optimized objective function *RatioCut*, and minimizing *RatioCut* is minimizing $f'Lf$.

Based on [12,33-35], we use vector space model and hypergraph clustering hybrid method (VSM-HC) to make the recommendation in this paper. Firstly we get the initial patterns recommendation list by vector space model, and then use the clustering algorithm to segment and cluster the hypergraphs. Finally, we calculated the nodes similarity of each hypergraph, and formed the final recommendation list according to the similarity. The improved algorithm is described as follows.

Step 1. Using the vector space model given in Section 3.3 to calculate the similarity between t and p, and the recommended list is constituted according to the similarity value from large to small.

Step 2. According to the user history behavior data and the task-model similarity measure, hypergraph model is constructed using the definition of hypergraph, that is $G = (V, E, w)$, where $V = U \cup T \cup P$, which is the union of all types of nodes, and w denotes the weight vector of each hyperedge. Each item weight of the feature vectors formed by the task and the pattern is different, and the calculating of weight is given in Equation (3).

Step 3. The similarity matrix A is constructed according to the hyperedges association and the weight between the nodes, where the value of the element a_{ij} is the weight of the nodes v_i and v_j through the hyperedges association. If there is no hyperedge between the two nodes, the associated weight is 0, where the constraint $a_{ij} = 0$. The calculating of weight is given in Definition 3.2.

Step 4. The value of each column of the similarity matrix A is added up to get the number of N, and put them on the diagonal to form an $N * N$ diagonal matrix D, that is a degree matrix. The diagonal element value is $d_{jj} = \sum_{j}^{N} a_{jj}$.

Step 5. The results of $D - A$ are denoted by the Laplacian matrix, i.e., $L = D - A$.

Step 6. Normalize the matrix L, i.e., $L = D^{-\frac{1}{2}} \bullet L \bullet D^{-\frac{1}{2}} = D^{-\frac{1}{2}} \bullet (D - A) \bullet D^{-\frac{1}{2}} = D^{-\frac{1}{2}} \bullet D \bullet D^{-\frac{1}{2}}$- $D^{-\frac{1}{2}} \bullet A \bullet D^{-\frac{1}{2}} = E - D^{-\frac{1}{2}} \bullet A \bullet D^{-\frac{1}{2}}$, so the normalization of L becomes the normalization of A.

Step 7. Find the first k feature values of L, which are $\lambda_1, \lambda_2, \ldots, \lambda_k$, where $\lambda_i \geq \lambda_j$ and $i < j$, i.e., the feature values are arranged from large to small, and the corresponding feature vectors $v_1, v_2, v_3, \ldots, v_k$ are obtained.

Step 8. These k feature vectors are put together to form the matrix of $N * k$, the partitioning of each node in the hypergraph is completed, and the elements in the set are clustered by k-means algorithm. The result of clustering completes a classification of N nodes in the original hypergraph.

Step 9. The vector space model is used to calculate the similarity between t_i and p_j, that is, $\frac{\sum_{p_j \in C} sel(u,p_j)}{\sum_{p_i \in C} p_i} * sim(t_i, p_j)$, where $\sum_{p_j \in C} sel(u, p_j)$ denotes the times which u selects p_j in hypergraph C, and $\sum_{p_i \in C} p_i$ denotes the total number of historical patterns selected in hypergraph C, $\frac{\sum_{p_j \in C} sel(u,p_j)}{\sum_{p_i \in C} p_i}$ indicates that u selects p_j for the current t_i as the proportion of all u selection patterns.

Step 10. The final pattern recommendation list is obtained from large to small according to similarity.

4. **Case Study.** In this paper, an example is given to illustrate patterns recommendation using VSM-HC method, and the feasibility of Laplacian decomposition and clustering method in patterns recommendation is also verified.

In the field of interface design, a variety of patterns emerge endlessly, and there are dozens of common search patterns in the mobile platform. Figure 4 shows several common search patterns [3,37].

Which pattern to use depends on the user task function, type and platform features. In order to accurately and quickly recommend an appropriate pattern for adapting to the task model to improve interface development efficiency. The attributes of each subtask in the process of task modeling are labeled, as shown in Figure 5.

Figure 5 shows several basic properties of "Multi-value input" task, and the description of the task may be described by one or two keywords or a paragraph, where the more information is provided, the higher accuracy of the search will be. Then we use the words segmentation system of Chinese Academy of Sciences to segment the attribute description information automatically, afterwards remove the stop words and extract the characteristic words; thus the task is expressed as a vector. Likewise, each pattern in the pattern library is also expressed as a vector of character value weights, and similarity is calculated using cosine similarity, so a similar pattern is recommended for each task.

Taking account of different user preferences, hypergraph weight can be obtained through Definition 3.2 according to empirical data of selecting patterns for different users. For example, u_1, u_2 and u_3 select the appropriate patterns for t_1, t_2 and t_3 to form a hypergraph, which is shown in Figure 6.

There are four hyperedges in the hypergraph, whose hyperedges are e_1, e_2, e_3 and e_4 respectively, where $e_1 = (t_1, u_1, p_1)$, $e_2 = (t_2, u_1, p_2)$, $e_3 = (t_3, u_2, p_2)$, $e_4 = (t_3, u_3, p_3, p_4)$, and the weight of each hyperedge is $w = (1.5, 0.8, 1.1, 1.2)$.

In order to recommend corresponding patterns, we need to divide hypergraph into two categories and establish corresponding similarity matrix and degree matrix, as shown in Figure 6.

Navigating around
- .Accordion
- .Headerless Menu
- .Breadcrumbs
- .Directory Navigation
- .Doormat Navigation
- .Double Tab Navigation
- .Faceted Navigation
- .Fly-out Menu
- .Home Link
- .Icon Menu
- .Main Navigation
- .Map Navigator

Serching
- .Advanced Serch
- .Autocomplete
- .Frequently Asked Questions(FAQ)
- .Help Wizard
- .Search Box
- .Search Area
- .Serch Results
- .Serch Tips
- .Site Index
- .Site Map
- .Footer Sitemap
- .Tag Cloud

Shopping
- .Booking
- .Product Comparison
- .Product Advisior
- .Product Configurator
- .Purchase Process
- .Shopping Cart
- .Store Locator
- .Testimonials
- .Virtual Product Display

Making choices
- .Country Selector

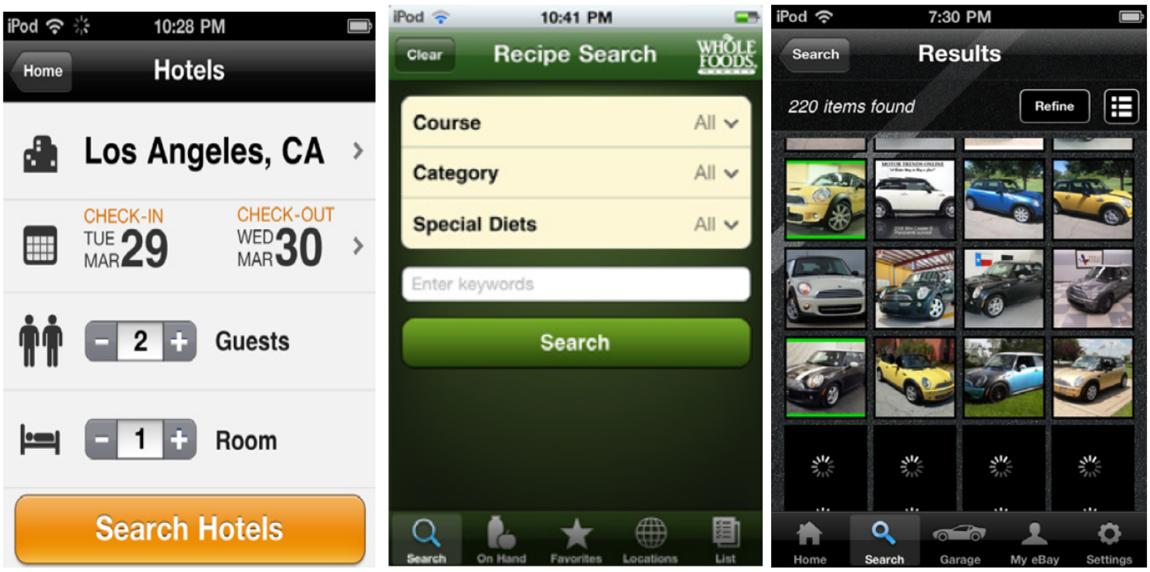
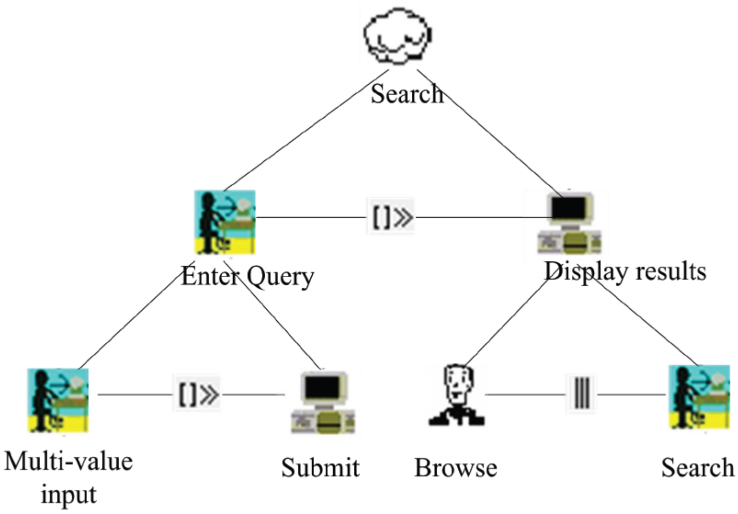


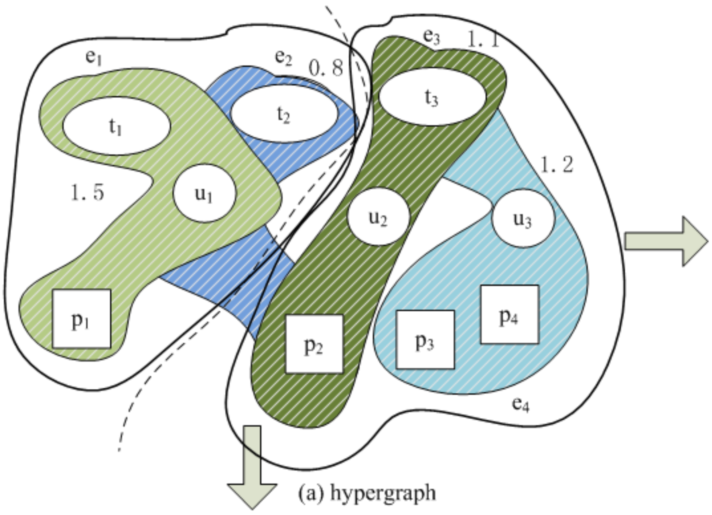
FIGURE 4. Several common search patterns



Attribute	name	type	function	how	platform

Attribute value (multi-value input, search, input text message,···, android)

FIGURE 5. The "Search" task model of CTT



(a) hypergraph

	e_1	e_2	e_3	e_4
t_1	1	0	0	0
t_2	0	1	0	0
t_3	0	0	1	1
u_1	1	1	0	0
u_2	0	0	1	0
u_3	0	0	0	1
p_1	1	0	0	0
p_2	0	1	1	0
p_3	0	0	0	1
p_4	0	0	0	1

(b) incidence matrix

	t_1	t_2	t_3	u_1	u_2	u_3	p_1	p_2	p_3	p_4
t_1	0	0	0	1.5	0	0	1.5	0	0	0
t_2	0	0	0	0.8	0	0	0	0.8	0	0
t_3	0	0	0	0	1.1	1.2	0	1.1	1.2	1.2
u_1	1.5	0.8	0	0	0	0	1.5	0.8	0	0
u_2	0	0	1.1	0	0	0	0	1.1	0	0
u_3	0	0	1.2	0	0	0	0	0	1.2	1.2
p_1	1.5	0	0	1.5	0	0	0	0	0	0
p_2	0	0.8	1.1	0.8	1.1	0	0	0	0	0
p_3	0	0	1.2	0	0	1.2	0	0	0	1.2
p_4	0	0	1.2	0	0	1.2	0	0	1.2	0

(c) similarity matrix

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
d_1	3	0	0	0	0	0	0	0	0	0
d_2	0	1.6	0	0	0	0	0	0	0	0
d_3	0	0	5.8	0	0	0	0	0	0	0
d_4	0	0	0	4.6	0	0	0	0	0	0
d_5	0	0	0	0	2.2	0	0	0	0	0
d_6	0	0	0	0	0	3.6	0	0	0	0
d_7	0	0	0	0	0	0	3	0	0	0
d_8	0	0	0	0	0	0	0	3.6	0	0
d_9	0	0	0	0	0	0	0	0	3.6	0
d_{10}	0	0	0	0	0	0	0	0	0	3.6

(d) diagonal matrix

FIGURE 6. The Laplacian matrix of the hypergraph

After normalizing the similarity matrix, the first k largest character values and the character vectors ($k = 2$) are obtained. The k character vectors are put together to form the characteristic matrix of $N * k$. Finally, the k-means clustering algorithm is used to obtain an N-dimensional vector $C = (2, 2, 1, 2, 1, 1, 2, 1, 1, 1)$, and the nodes in Figure 6 are divided into two categories: (t_1, t_2, u_1, p_1) and $(t_3, u_2, u_3, p_2, p_3, p_4)$, achieving the segmentation. For example, the recommended pattern for task t_1 under current user u_1 is p_1. If the partitioned hypergraph has several patterns, we need to calculate the similarity according to the recommendation algorithm step (9) to form a recommendation list for the user recommendation.

5. **Recommended System Prototype of Interface Patterns.** In this paper, based on the analysis of CTT tasks, VSM-HC method is used to recommend the interface patterns matched with the task from the patterns library, which improves the development efficiency. In order to further verify the practicability of the method, this paper develops a prototype system of interface design. The interface patterns recommendation system prototype is shown in Figure 7.

The prototype system consists of three views: The middle is the task modeling area, which is presented in two forms of design view and XML view, among them design view is presented in a graphical way for users to task modeling and design, and XML view is

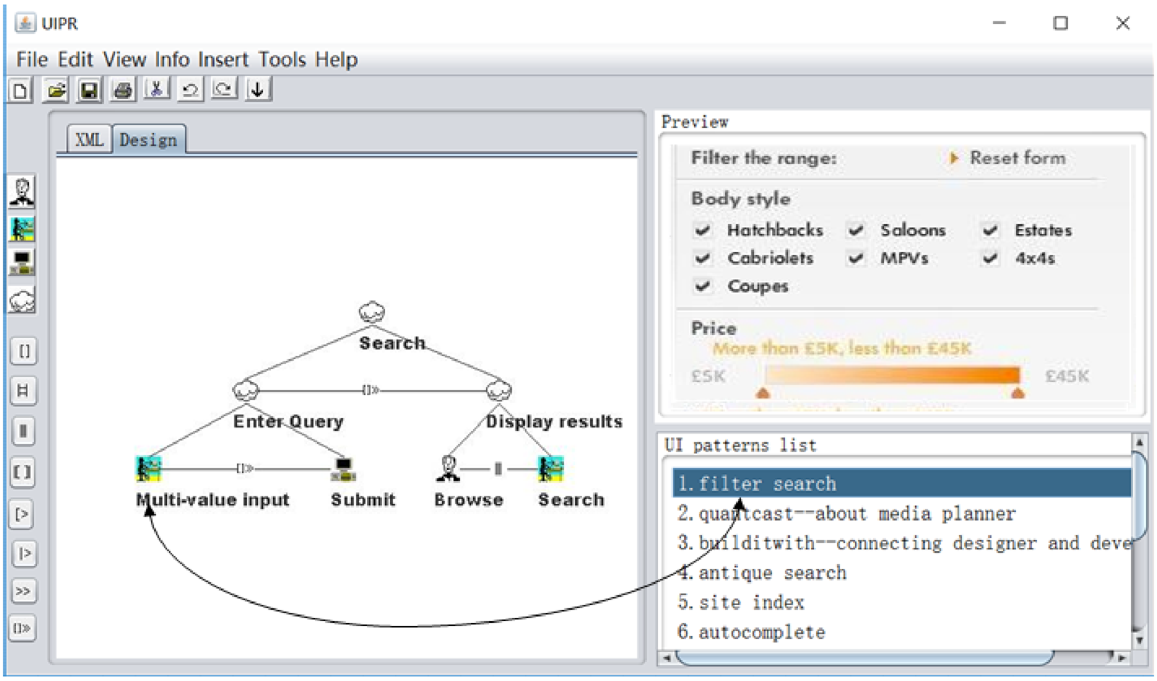


FIGURE 7. Recommendation system prototype of interface patterns

presented in the form of XML tags storing various attribute definitions and relationships of sub-tasks; The upper right corner is the currently selected interface pattern profile; The lower right corner is the list of patterns recommended by the system for the task and the user, when the user selects one of them, it is displayed in the upper right in the form of screenshot, and the user can also double-click to view the interface patterns definition and details.

The user can drag and drop the selected pattern to the corresponding task in the CTT model, which establishes an association with the pattern. After the task model is expressed in the form of a pattern, the task model forms a sequence in the form of patterns for the next transition from the task model to the abstract interface.

6. Systematic Assessment.

This paper will evaluate the effectiveness of the algorithm and the overall performance of the prototype system respectively.

6.1. Evaluation of recommendation effect based on hypergraph theory.

This paper compares TF*IDF and VSM-HC respectively with the precision and the recall rate as the criterion based on the Martijn van Welie pattern set.

Precision and recall are defined as (see Equations (8) and (9)):

$$\text{precision} = \frac{\text{Number of patterns that match with the task}}{\text{Length of recommendation list}} \tag{8}$$

$$\text{recall} = \frac{\text{Number of patterns retrieved}}{\text{The total number of patterns in the pattern library}} \tag{9}$$

Results in Figure 8 show that, as the number of recommended patterns increases, the list of recommended patterns is more closely matched with the task, and the precision is improved significantly. However, when the number of patterns exceeds 13 in the list, the matching degree between patterns and tasks will gradually decline; therefore, the length of the list is recommended to set to 13 or so.

As can be seen from Figure 9, with the number of task attribute tags increasing, the number of retrieved patterns increases. When the number of attribute tags reaches 7, the

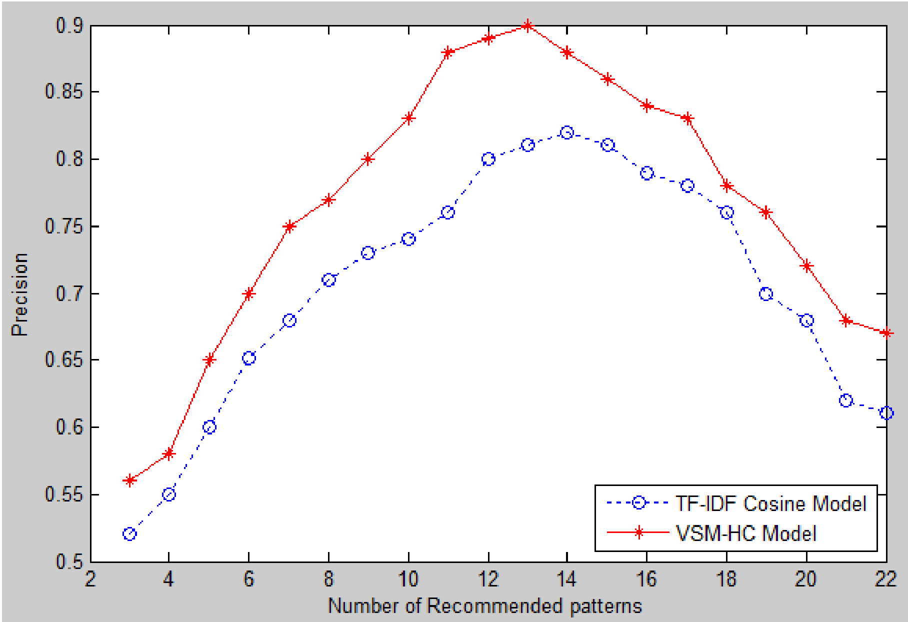


FIGURE 8. Comparison of the precision of different recommended methods

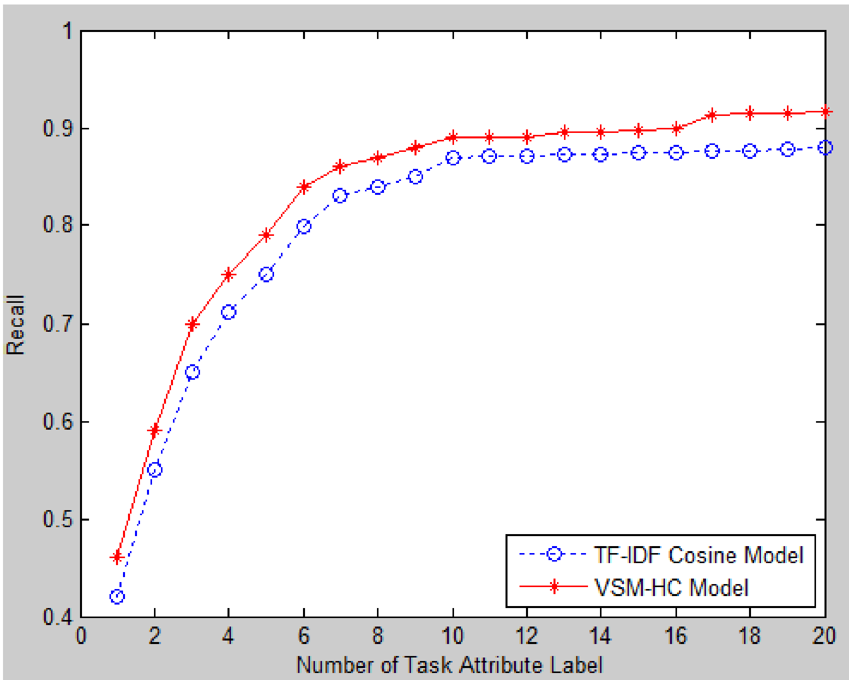


FIGURE 9. Comparison of recall rate of different recommended methods

improvement of recall rate becomes less obvious, so this paper will set the number of task attributes label to 7. The more detailed the description of the task attributes is, the more accurate the proposed patterns will be. The precision and recall rate will increase with the increase of k, when the hypergraph is divided into a reasonable range, the precision of recommended system is relatively high.

6.2. **System prototype evaluation.** To evaluate the system prototype, we mainly evaluate the usability of the system (interface friendly, learnability, operability), the recommendation effect and the efficiency. The evaluation scale of the system has 5 grades [39]:

Very important (0.9), Important (0.75), General (0.6), Unimportant (0.3) and Very insignificant (0.1). The overall performance evaluation scale of the system also has 5 grades: Very satisfying, Satisfying, General and Unsatisfactory and Not at all satisfying.

20 interface developers are selected as test objects. Before test, the weight of each index is determined by the scoring of each person. The weight analysis of the system performance evaluation comprehensive is shown in Table 1.

Table 1. The weight analysis of system performance evaluation comprehensive

score / significance index — evaluation index	Very important (0.9)	Important (0.75)	General (0.6)	Unimportant (0.3)	Very insignificant (0.1)
Interface friendly	1	6	8	5	0
Learnability	9	8	2	1	0
Operability	12	8	0	0	0
Recommendation effect	16	4	0	0	0
Efficiency	14	5	1	0	0

Table 2. The system satisfaction score

score / satisfaction index — evaluation index	Very satisfying	Satisfying	Ordinary	Unsatisfactory	Not at all satisfying
Interface friendly	6	10	2	1	1
Learnability	7	9	3	1	0
Operability	4	8	5	2	1
Recommendation effect	8	9	3	0	0
Efficiency	7	9	4	1	0

Then let each person use the prototype system proposed in this paper to design a car rental system interface, and evaluate the satisfaction of using the system. The satisfaction evaluation of the situation is shown in Table 2.

According to the fuzzy comprehensive evaluation method [39,40], the comprehensive evaluation weight and satisfaction membership matrix of the system are established, as shown in Figure 10.

Weight evaluation vector $W = W^s \bullet R'_{weight} = (0.9, 0.7, 0.6, 0.3, 0.1) \bullet R' = (0.57, 0.76, 0.82, 0.86, 0.835)$, the unitized weight matrix $W = (0.1482, 0.1977, 0.2133, 0.2237, 0.2172)$. The membership score $B = W \bullet R'_{satisfaction} = (1.2373, 1.7178, 0.6720, 0.1903, 0.0695)$, and the unitized membership degree B is $(0.3183, 0.4420, 0.1729, 0.0489, 0.0179)$, and finally the comprehensive score is 2.0061. It can be seen that the overall evaluation of the system by the user is "Satisfying".

7. **Conclusion and the Future Work.** In order to improve the efficiency of user interface development, a recommendation method and a development framework of interface patterns based on VSM-HC model are proposed. The hypergraph is used to describe the

$$\begin{bmatrix} 0.05 & 0.3 & 0.4 & 0.25 & 0 \\ 0.45 & 0.4 & 0.1 & 0.05 & 0 \\ 0.6 & 0.4 & 0 & 0 & 0 \\ 0.8 & 0.2 & 0 & 0 & 0 \\ 0.7 & 0.25 & 0.05 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0.3 & 0.5 & 0.1 & 0.05 & 0.05 \\ 0.35 & 0.45 & 0.15 & 0.05 & 0 \\ 0.2 & 0.4 & 0.25 & 0.1 & 0.05 \\ 0.4 & 0.45 & 0.1 & 0 & 0 \\ 0.35 & 0.45 & 0.05 & 0.05 & 0 \end{bmatrix}$$

(a) Weighted membership (b) Satisfaction membership

FIGURE 10. Weight and satisfaction membership matrix

user-pattern-task relationship, the hypergraph is segmented by spectral clustering, and the vector space model is used to calculate the similarity between the tasks and the patterns in the same hypergraph, so as to recommend satisfied patterns to users and tasks, and improve the efficiency of interface development.

In the future, we will collect more interface patterns to enrich interface pattern library, perfect the recommendation system, formalize the task model, and thereby speed up the transition from the task model to the abstract interface and specific interface model, to improve the degree of automation in the interface development process.

Acknowledgment. This research is funded by the National Natural Science Foundation of China (No. 61272286), and Joint Funded Projects of the Special Scientific Research Fund for Doctoral Program of Higher Education (No. 20126101110006), and Science Research Foundation of Northwest University (No. 15NW31), and Industrial Science and Technology Research Project of Shaanxi Province (No. 2016GY-123).

REFERENCES

[1] A. Seffah, *Patterns of HCI Design and HCI Design of Patterns*, Springer International Publishing, 2015.
[2] S. Ahmed and G. Ashraf, Model-based user interface engineering with design patterns, *Journal of Systems & Software*, vol.80, no.8, pp.1408-1422, 2007.
[3] T. Neil, *Mobile Design Pattern Gallery*, O'Reilly Media Incorporation, 2012.
[4] J. Engel, C. Herdin and C. Maertin, Exploiting HCI pattern collections for user interface generation, *Proc. of Patterns 2012*, Nice, France, pp.36-44, 2012.
[5] J. Janeiro and S. D. J. Barbosa, Enhancing user interface design patterns with design rationale structures, *International Conference on Design of Communication*, Bloomington, IN, USA, pp.9-16, 2009.
[6] C. Kruschitz and M. Hitz, Analyzing the HCI design pattern variety, *Proc. of the 1st Asian Conference on Pattern Languages of Programs*, 2010.
[7] F. Moussa and M. Riahi, Use of formalized guidelines for semi-automated generation of GUI: The Ergo-conceptor tool, *Tools for Working with Guidelines*, Springer, London, pp.237-246, 2001.
[8] Y. Pan, F. He and H. Yu, A socialized recommendation algorithm based on implicit similarity of trust relation, *Chinese Journal of Computers*, 2016.
[9] M. Wang, *Research on Socialized Recommendation Algorithm Based on Weighted Trust Relationship and User Similarity*, Shandong University, 2015.
[10] J. Wang, *Theoretical Basis of Hypergraph*, Higher Education Press, 2006.
[11] J. Lin, *Research on Personalized Recommendation Technology Based on Complete Ternary Graph Model*, Shanghai University, 2015.
[12] W. Gu, *Research on News Recommendation Based on Hypergraph Model*, South China University of Technology, 2015.

[13] I. Bloch, A. Bretto and A. Leborgne, *Similarity between Hypergraphs Based on Mathematical Morphology, Mathematical Morphology and Its Applications to Signal and Image Processing*, Springer Berlin Heidelberg, pp.1-12, 2013.

[14] X. Xu and R. Huang, Hypergraph theory and its application, *Journal of Electronics*, no.8, pp.65-72, 1994.

[15] Q. Liu, *Research on User Interface Generation Technology Based on Interface Pattern*, Northwest University, 2014.

[16] J. Harland, J. O. Blech and I. Peake, Formal behavioural models to facilitate distributed development and commissioning in industrial automation, *Special Session on Collaborative Aspects of Formal Methods*, pp.363-369, 2016.

[17] S. Wendler, D. Ammon and T. Kikova, Theoretical and practical implications of user interface patterns applied for the development of graphical user interfaces, *International Journal on Advances in Software*, no.6, pp.25-44, 2013.

[18] J. Engel, C. Märtin and C. Herdin, Formal pattern specifications to facilitate semi-automated user interface generation, *International Conference on Human-Computer Interaction: Human-Centred Design Approaches, Methods, Tools, and Environments*, pp.300-309, 2013.

[19] J. Engel and C. Märtin, PaMGIS: A framework for pattern-based modeling and generation of interactive systems, *International Conference on Human-Computer Interaction New Trends*, San Diego, CA, USA, no.7, pp.826-835, 2009.

[20] Q. Gon and J. Cheng, Shortest path algorithm for hypergraph, *Journal of Chongqing University: Natural Science Edition*, vol.28, no.11, pp.106-109, 2005.

[21] I. Bloch and A. Bretto, Mathematical morphology on hypergraphs: Preliminary definitions and results, *IAPR International Conference on Discrete Geometry for Computer Imagery*, Springer-Verlag, pp.429-440, 2011.

[22] P. Kapec, Visual programming environment based on hypergraph representations, *International Conference on Computer Vision and Graphics*, Springer-Verlag, pp.9-16, 2010.

[23] R. Zass and A. Shashua, Probabilistic graph and hypergraph matching, *Computer Vision and Pattern Recognition*, pp.1-8, 2008.

[24] A. Dearden and J. Finlay, Pattern languages in HCI: A critical review, *Human-Computer Interaction*, no.1, pp.49-102, 2006.

[25] C. Phanouriou, UIML: A device-independent user interface markup language, *Dissertation Abstracts International*, 2000.

[26] Y. Zhang, L. Wang and H. Cao, A recommendation algorithm based on user-interest-project tripartite graph, *Pattern Recognition and Artificial Intelligence*, vol.28, no.10, pp.913-921, 2015.

[27] B. Yang and P. Zhao, A survey of recommendation algorithms, *Journal of Shanxi University*, vol.34, no.3, pp.337-350, 2011.

[28] H. Rong, S. Huo and C. Hu, Collaborative filtering recommendation algorithm based on user similarity, *Journal of Communications*, no.2, pp.16-24, 2014.

[29] S. Liu, *Research and Application of Key Issues in Recommendation-Oriented Systems*, University of Science and Technology of China, 2014.

[30] Y. Li, B. Wang and J. Li, Summarization of search engine recommendation technology, *Chinese Journal of Information*, vol.24, no.6, pp.75-84, 2010.

[31] W. Cao, *Research and Implementation of Chinese Information Retrieval System Based on Structured Vector Space Model*, Beijing Industry University, 2008.

[32] S. Henninger and P. Ashokkumar, Disseminating usability design knowledge through ontology-based pattern languages, *The 3rd International Semantic Web User Interaction Workshop (SWUI)*, University of Nebraska-Lincoln, 2006.

[33] W. Jia and Q. Hua, User interface pattern language based on category theory, *Journal of Computer-Aided Design & Computer Graphics*, vol.29, no.6, 2017.

[34] L. Zhou, X. Ping and S. Xu, Clustering integration algorithm based on spectral clustering, *Acta Automatica Sinica*, vol.38, no.8, pp.1335-1342, 2012.

[35] H. Wang and J. Chen, Research on spectral clustering method based on graph partition, *Computer Engineering and Design*, vol.32, no.1, pp.289-292, 2011.

[36] C. Wu and C. Zhu, A method of social relationship data processing based on spectral clustering, *Electronic Design Engineering*, vol.24, no.20, pp.20-23, 2016.

[37] M. van Welie, *http://www.welie.com/patterns/index.php.*

[38] Z. Huang, J. Zhang and B. Zhang, A survey of semantic recommendation algorithm, *Journal of Electronics*, vol.44, no.9, pp.2262-2275, 2016.

[39] H. Fang and H. Fang, Fuzzy evaluation model and application of educational software usability for teachers, *Computer Engineering and Applications*, vol.44, no.29, pp.245-248, 2008.
[40] D. Xiong and X. Xian, Improvement of fuzzy comprehensive evaluation method, *Journal of Chongqing University (Natural Science Edition)*, vol.26, no.6, pp.93-95, 2003.
[41] W. Gu, S. Dong and Z. Zeng, HCKP: An effective hybrid classfication for Chinese news, *Journal of Information and Computational Science*, vol.11, no.9, pp.3117-3123, 2014.
[42] W. Gu, S. Dong, Z. Zeng and J. He, An effective news recommendation method for microblog user, *The Scientific World Journal*, vol.2014, no.5, 2014.
[43] Z. Huang, J. Zhang and B. Zhang, Survey of semantics-based recommendation algorithms, *The 4th International Joint Conference on Computational Sciences and Optimization*, 2016.
[44] E. Nilsson, Design patterns for user interface for mobile applications, *Advances in Engineering Software*, vol.40, no.12, pp.307-312, 2008.