Content-Based Filtering Algorithm for Mobile Recipe Application

Teh Lee Cheng, Umi Kalsom Yusof, Mohd Nor Akmal Khalid
School of Computer Sciences
Universiti Sains Malaysia
USM, 11800 Georgetown,
Pulau Pinang, Malaysia
tlcheng.ucom10@student.usm.my, umiyusof@cs.usm.my, akmal104041@gmail.com

***Abstract*—In general, people like to cook but they have no idea on what to cook and how to cook. Furthermore, they also wish to know certain recipes that suit to their preference. There are many cooking recipe websites but most of them do not provide suggestions for the user based on user's food preferences. Current recipe suggestions developed by some researchers are mainly based on the collective interests, demographic information, and ingredients content. With respect to the problem mentioned previously, a content based mobile recipe application is proposed and tested in this paper. The main objective of this proposed application is to suggest a user preferred recipe using content-based filtering algorithm. Content-based filtering algorithm (CBFA) will be applied to identify the recipes that have high possibility for user to like. This algorithm will be able to recommend recipes based on user interaction. The algorithm will consider a few attributes to identify the similarity between the recipe pages viewed by the user. The reason to choose this algorithm is because it does not have the cold-start problem and it solely depends on the active user to identify their preferences. An Android based mobile recipe application is built to test this algorithm. The results from the experiment conducted show that the proposed algorithm is effective for recommending preferred recipes to user.**

Keywords—Content-based Filtering Algorithm; Mobile Application

I. INTRODUCTION

Nowadays, people are faced with a limited selection of meals and are often constrained in searching for recipes (i.e. having full working day). Preparing meals may requires people to search, think, prepare, and even learn new recipes, but still poses a risk of frustrating the family members, spouse, or themselves [1]. This problem is caused by the new recipe which can vary from the preferred set of meals and does not fit the daily habits. Additionally, certain people would wish for different cooking recipe(s) that provide variation, but still suits their preferences [1].

In addition, the sense of taste for every person is distinctive from each other. Not every person prefers to take similar kind of meal. A person may prefer meal *x* above others, while another person may prefer meal *y* instead. Moreover, every person has their own meal preference(s) such as person *A* likes to eat spicy food while person *B* likes to eat sweet food. However, recommending a new or different recipe which does not diverge from a person's preferred taste and habit are the main issue. Therefore, to recommend the right recipe which is based on a personalized preference(s) would be very useful and commendable.

Currently, many cooking recipe websites and mobile recipe applications have been launched, such as *Cookpad* and *BigOven* that allow people to search for recipes. Various recipes can be found from these mobile applications, which sometimes include media instructions. However, they cannot provide a personalized recipe recommendation based on the user's preferences. Despite plenty of recipes being provided by the cooking application, the application user always has to spend additional time to browse through recipes that they are not particularly interested in, making the user frustrated and eventually, lose interest. They need an easy way to get their preferred recipe(s) as soon as possible and as accurate as possible, with only a single click of a button [1].

Therefore, the objective of this paper is to propose CBFA for a mobile recipe application. It focuses on recommending the application user recipe(s) based on their preferences. CBFA recommends items based on the user's preference, which depends on the ingredients of the recipe itself. In order to keep track of each user's preference, the application keeps the history that the user has viewed. An Android based mobile recipe application utilizing CBFA will be developed to evaluate CBFA's effectiveness.

The remainder of this paper is organized as follows: Section II discusses the background study of the current development of the recommendation algorithm, where comparison between the existing algorithms and the reason to choose content-based filtering algorithm is highlighted. Section III elaborates the proposed solution, which includes the application development and the content-based filtering algorithm. Section IV discusses the results obtained and the analysis done. Finally, Section V concludes the paper.

II. BACKGROUND STUDY

Recommendation system are widely used on e-commerce, social networking, and entertainment web sites such as Amazon.com, Facebook and Levis.com. Recommendation

978-1-4799-5439-1/14/$31.00 ©2014 IEEE

systems provide recommendations for products, services, and information that might have the best chance to attract people [2]. Recommendation systems applies knowledge discovery techniques to analyze the user preference based on the interaction of user , which helps the user deal with a lot of information while reducing the searching time to get the information of interest. Recommendation systems must obtain input from the user to collect the required data needed to compute for a specific recommendation. Generally, there are two types of recommender algorithms suitable to solve the underlying problem [3]: CBFA and collaborative filtering algorithm. Other recommendation algorithms include the demographic filtering algorithm which is not explicitly suitable for the considered problem but still provides useful insights.

A. Content-Based Filtering Algorithm (CBFA)

CBFA is an algorithm that performs item recommendation based on the content of the item and the user's preference. It finds the similarities between the features of an item that the user likes and then recommends another item which has the common features of the item that the user likes [4] [5]. For example, the features of a movie such as the actor, director, and genre of the movie are needed in order to make a suitable movie recommendation. Some people only like horror movies, while others may prefer to watch a movie with their favourite actors. Amazon has used this content-based profiling to provide the "favourite" feature to indicate a user's preference on categories of items. In addition, CBFA also tries to recommend items that are similar to those that the user had liked in the past [6] [7]. For example, the system would recommend movies in the same genre (horror), for someone who has watched and enjoyed many horror movies. However, in order to make a precise recommendation, the system would need sufficient information such as description on types of items that a particular user is interested in and the history of user's interaction with the system to collect and populate the user's preference(s). The system can explicitly collect relevant feedback such as ratings and comments given by the user or implicitly by observing user's interaction such as the categories that the user has viewed [7] [5]. An example of such a recommendation system is the system proposed in [8], where food is recommended based on the eating preference in the patient care facility.

B. Collaborative Filtering Algorithm

The collaborative filtering algorithm uses the preferences of other users with similar interests to predict another user's interest. Unlike content-based filtering, collaborative filtering can recommend an item by combining the opinions of other like-minded individuals without understanding what the item is about [2] [3]. For example, person X likes to watch movie A, movie B and movie C, while other users like to watch movie A, movie B, movie C and movie D. As such, the system will recommend movie D to this person X because of the interest and taste in movies that they share. Through this technique, users can discover new items that are within their interest but were previously unknown. The opinions of users can be obtained explicitly and implicitly through the rating given by users [9] [6] [7], which can be used to further improve the technique. Such systems exist for online food store recommendation as was [10], where users are aggregated based on group labels as well as their purchasing habits.

C. Demographic Filtering Algorithm

The demographic filtering algorithm uses data such as age, gender, education and income to categorize users group. For example, people of high level of education, people of below 18 years old and people of high salary can be grouped into different categories. If a customer has a high level of education, results will be filtered based on the preference of high level of education. The user's need for a product are strongly related to these variables. For example, age group such as teenagers tends to like trendy clothing, whereas age group such as high salary will tend to like branded and high quality products [11].

D. Comparative Analysis Between The Filtering Algorithms

Each of the filtering algorithms have their own strengths and drawbacks. One of the advantages of CBFA is user independence. CBFA solely depends on ratings provided by the active user to learn the user's preference [5]. However, a collaborative filtering algorithm depends on other like-minded users to rate the same item to gather the users' preferences. Moreover, CBFA does not suffer from the "cold-start" problem for new items where interaction with the user is required beforehand. The collaborative filtering algorithm will have this problem because the new item would cease be recommended only if the new item is rated by a number of users. If the user feedback is nonexistent, collaborative filtering algorithm will have difficulty finding similar users [2] [12] [3]. One of the drawbacks of CBFA is the ability to provide recommendation would negatively be affected if there is not enough data to distinguish the item that the user likes and/or dislikes [5]. On the other hand, the disadvantage of demographic filtering algorithm is mainly due to the scarcity of data from users as it requires sensitive data that can infringe user's privacy. Furthermore, recipe recommendation application is based on user preference which does not pertain to these variables, so the demographic filtering algorithm is unsuitable to handle the underlying problem.

After reviewing the advantages and drawbacks of recommendation systems, content-based filtering system is chosen to provide recipe recommendations to users. This is because content-based filtering does not have the "cold-start" problem and it solely depends on the active user in order to obtain their preference(s) [5]. Besides that, it is easy for a recipe recommendation system to gather and adapt to the user's preferences, which can be tracked through the user's search history. This is reasonable because users do not search for a recipe that they dislikes; may simply looks for recipes that they like the most. The drawbacks of the content-based filtering can be solved after some input from the user is obtained, collected, and employed into the proposed mobile recipe application.

III. Content-Based Filtering Algorithm For Suggesting User Preference Recipe

A. System Development and Framework

The mobile recipe application was developed using Java Eclipse Juno and Android SDK was used for the development environment and user interface, respectively. My SQL and SQLite database were used to store all the information related to this application. This is a client server application, Wamp server was used at the server side. PHP (Hyper-text Preprocessing) was used for database accessing. JSON was used as the data-interchange between the client and server side. Besides the suggestion function, this application also includes some other functions not covered in this paper as the main focus is on the suggestion function using the CBFA.

Basically, the proposed mobile recipe application consists of three components, which are the user interface, databases and the recommender module. The user interface translates the stored information in the database to a human-readable language for the users and receives inputs from the users. The database consists of three parts; the recipe information repository, the user profile repository, and the user's log repository. The recipe information repository and user profile repository basically stored related information for the recipes and the users, respectively. The user's log repository keeps track of every detail of every user action within the system. In addition, the user's log repository is the backbone of the CBFA where its data is used to determine the appropriate recipe to recommend. The framework of the proposed mobile recipe application is given in Fig. 1.

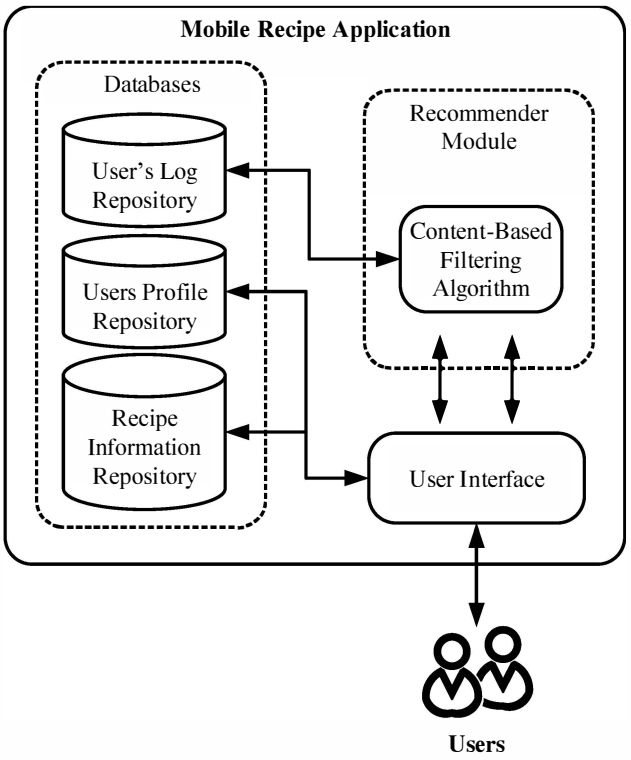

Fig. 1: The framework of the proposed mobile recipe application

B. Content-Based Filtering Algorithm

The CBFA is founded based on the content of the recipes and the user's preference. It finds the features of an item that the user likes or items that are similar to those that the user liked in the past and then recommend another item that has similar features to the item that the user had liked. So, in order to collect and identify the user's preference(s), information regarding the user's input into the application will be tracked. Every search made by a user is recorded and stored in the user's log repository. The main requirement to be considered as a candidate to be recommended is that the user must first view at least one recipe. For users who have never viewed any recipe, but if wants to request for suggestions, users can choose a suggestion based on his/her criteria instead. The user searching history which is stored in the user's log repository is used for computing and identifying his/her preference(s). This is because it is rare for a user to accidently perform a search or search for an undesirable recipe.

Essentially, there are three major steps in the recipe recommendation application using CBFA: data collection, the computation of the weight of each featured content, and the recipe recommendation. In the end, the system will recommend recipes which have the highest weight of the common feature.

1) Data Collection

At this stage, data is collected from the user's log repository and the recipe information repository. The number of views by the particular user in each category is calculated. Then, the category that has the highest number of views is selected.

2) Calculate the weight of each feature in the category that has the highest number of views

This is the most important step in order to make an accurate recommendation. If the category i has the highest number of views C_j, but it has many content's features j in specific category i, the application needs to compute the weight W_i of each featured content view C_i within that category. The equation given below is adopted from [1].

$$W_i = \frac{C_i * \sum_{j=0}^{n} C_j}{n} \qquad (1)$$

3) Recommend Recipe

After computing the weight of each featured content, only the recipes which have the highest weight of common featured content and has never been viewed by the user before will be displayed as a recommendation. The overall flowchart of the system is given in Fig. 2 below.

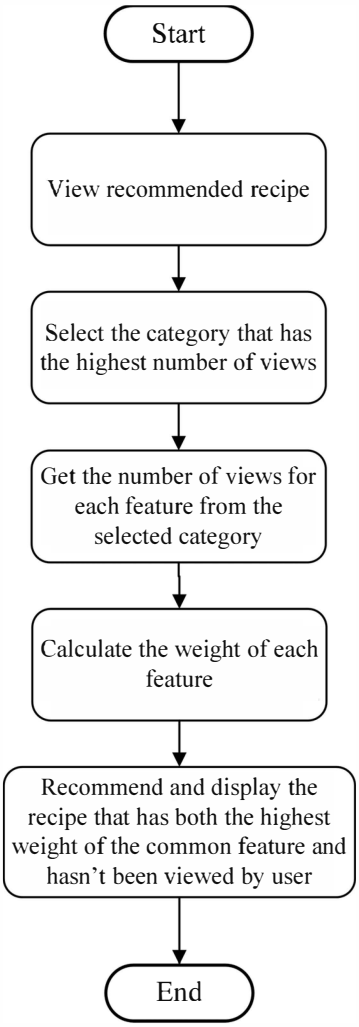

Fig. 2: Flow chart of CBFA for suggesting user preference recipe

To demonstrate the usage of the proposed CBFA computation, assume the highest view category is *dessert* with 10 views. The featured content in the *dessert* category are *cake*, *pie*, and *pudding* and the number of views by the user are 2, 5, and 3, respectively. Thus, the number of features in the highest category is 3. So the weight of the cake is (2*10)/3 = 6.67.

The application calculates the weight of all the features inside the category with the highest number of views. The next step is to compare the weight of each feature and then get the feature that has the highest weight. After that, the application retrieves the recipes that have common features and selects the feature with the highest weight that has not been viewed and recommends it to the user. Based on the example given, the highest weight of the featured content is 'pie', which implies that the user likes to eat desserts and the most he/she likes to eat is pie. Thus, the application will recommend other related recipes to the user that are ultimately based on his/her preferences.

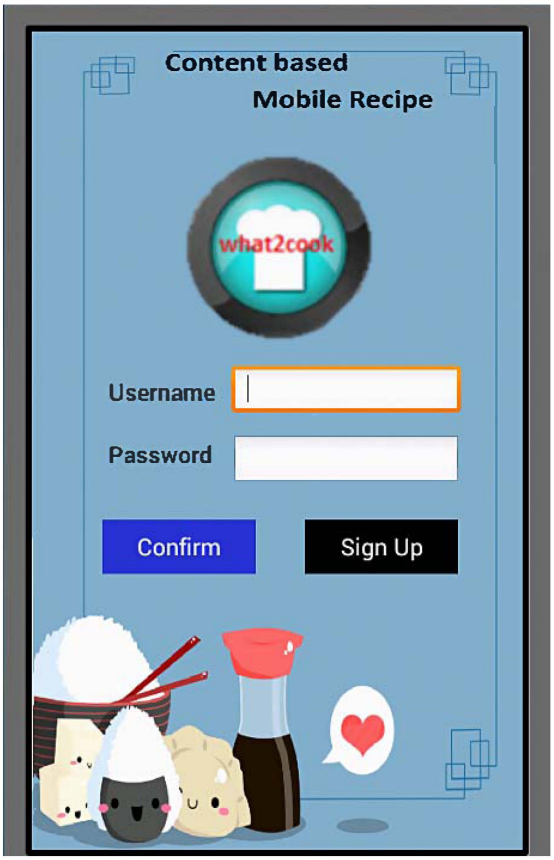

Fig. 3: The login page of the mobile recipe application

IV. Result and Discussion

To exemplify how the content-based filtering algorithm works, data regarding recipe information and user's log information was accumulated and stored in the database manually. The proposed CBFA was tested by user *A* through the mobile recipe application interface. In order to keep track of each user's preference, the application requests the user to be registered in the application. This created a personalized profile for that particular user and required user to login to the application each time the user wants to search for a recipe from the application, as shown in Fig. 3.

After the user logged into the application, the application kept track of the user's search history and stored it into the user's log repository. The user can search by category or by the recipe name. The number of searches, the recipe, and user identity are stored in the user's log repository. A sample of 5 categories with several featured content for each category with the user search history is given in Table I.

TABLE I. A SAMPLE OF CATEGORIES, FEATURED CONTENT, AND USER'S SEARCH HISTORY

Categories	*Featured*	*No. of View For User A*	*Total No. of Views*
Appetizer	Cheese	25	72
	Spicy	39	
	Vegetable	8	
Breakfast & Brunch	Bread	27	39
	Toast	12	
Chicken	Fried	24	52
	Soup	28	
Desserts	Cake	39	**88**
	Pie	38	
	Pudding	11	
Main Dish	Spagetti	13	42
	Curry	29	

Once the user clicks the suggestion button, the application compares the number of views in each category and finds the category with the highest number of views. From Table I, the category with the highest number of views by the respective user is *dessert*. Then the application calculates the weight of each featured contents within that category. The category that has the highest number of views by the respective user and the weight of each feature is presented in Table II. From the weight of features obtained, the application can distinguish that the particular user preferred *dessert* and *cake* recipes the most. As such, the application would recommend other related *cake* recipes to the respective user. The example of the recommended output for respective user *A* is given in Fig. 4.

TABLE II. EXAMPLES OF WEIGHT FOR EACH FEATURE WITH HIGHEST NUMBER OF VIEWED CATEGORY

Category	*Featured*	*No. of Views for User A*	*Weight (W_i)*
Desserts	Cake	39	**1144**
	Pie	38	1114.67
	Pudding	11	322.67

To further analyze the CBFA, the W_i and C_i of each featured content of the previous example are computed and visually illustrated in Fig. 5. The observed pattern shows that whenever a featured content (e.g. *cake*) is viewed more (increasing C_i) had significantly increased the weight of that particular featured content W_i, which implied that the featured content of that category of recipe is highly preferred by the user. In addition, appropriate distinction of the preferred featured content by the respective user can be identified and aggregated based on the category of the recipe. As seen in the Fig. 5, comparing the views of the featured content alone is insufficient since similar number of featured content views may be misinterpreted as the preferred feature of the user. However, using the CBFA computation, even with similar

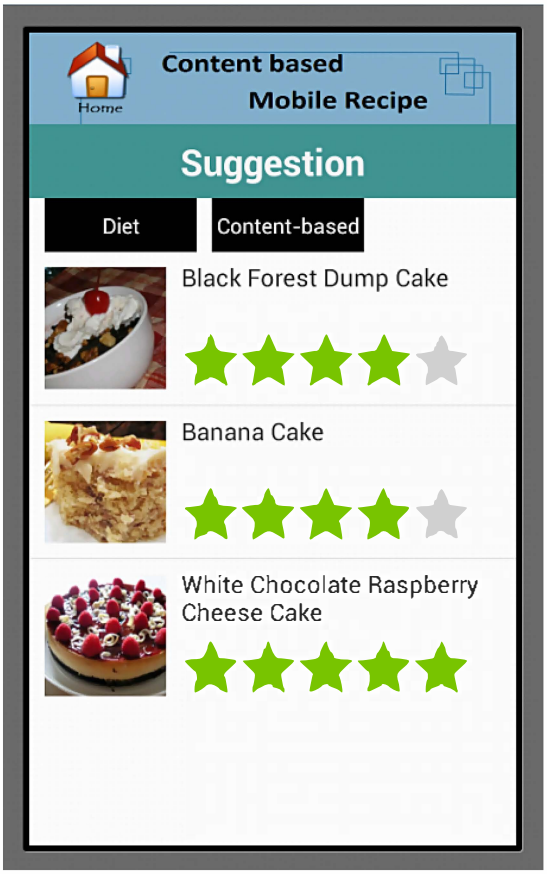

Fig. 4: The recommended recipe(s) based on content-based filtering algorithm

number of views, the featured content may be distinguished as one preferred by the respective user or otherwise. As such, this shows that the proposed CBFA is able to recommend recipe(s) that meet user preference(s).

V. CONCLUSION

In conclusion, this paper proposes a content based mobile recipe that is able to suggest a user preferred recipe using CBFA. The proposed solution helps those who have no idea what to cook with only one click of a button. With this application, the user does not need to spend a lot of time to get his/her preferred recipe.

The main objective of this paper is to suggest a preferred recipe for his/her using a CBFA. This application keeps track of user preference(s) through user interactions from the page(s) that user has viewed. In order to suggest an accurate user preferred recipe, the weight of each feature for the recipes that user has viewed is calculated. Only the recipes which have the same features as the highest weight and have not yet been viewed by the user will be displayed as recommendation for the user. Further research can be done by implementing learning or prediction algorithms (such as neural network) and more user preference(s) considered through other types of user interaction (such as user favorite's recipe) so that the accuracy of the suggestion can be enhanced.

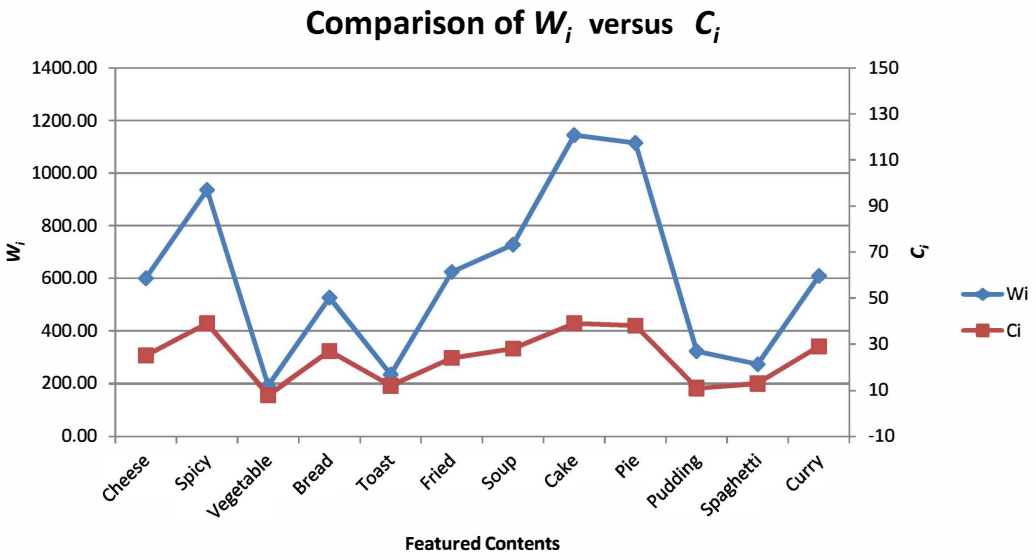

Fig. 5: The comparison of W_i against C_i

ACKNOWLEDGMENT

The author wishes to thank Universiti Sains Malaysia for the support it has extended in the completion of the present research through Short Term University Grant No: 304/PKOMP/6313026.

REFERENCES

[1] Y. van Pinxteren, G. Geleijnse, and P. Kamsteeg, "A user-centered approach to define a content-based recipe similarity measure," 2010.

[2] Q. Li and B. M. Kim, "Constructing user profiles for collaborative recommender system," in *Advanced Web Technologies and Applications*. Springer, 2004, pp. 100–110.

[3] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The adaptive web*. Springer, 2007, pp. 291–324.

[4] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*. Springer, 2007, pp. 325–341.

[5] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*. Springer, 2011, pp. 73–105.

[6] A. Rajaraman and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2011.

[7] C. Ramadoss, "Recipe recommendation."

[8] T. De Pessemier, S. Dooms, and L. Martens, "A food recommender for patients in a care facility," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 209–212.

[9] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining collaborative filtering with personal agents for better recommendations," in *AAAI/IAAI*, 1999, pp. 439–446.

[10] M. Svensson, J. Laaksolahti, K. Höök, and A. Waern, "A recipe based on-line food store," in *Proceedings of the 5th international conference on Intelligent user interfaces*. ACM, 2000, pp. 260–263.

[11] K. Taha and R. Elmasri, "Social search and personalization through demographic filtering," *Social Network Mining, Analysis, and Research Trends: Techniques and Applications*, vol. 183, 2012.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 285–295.