

An overview of recommender systems in the healthy food domain

Thi Ngoc Trang Tran¹  · Müslüm Atas¹ ·
Alexander Felfernig¹ · Martin Stettinger¹

Received: 10 October 2016 / Revised: 24 May 2017 / Accepted: 30 May 2017 /

Published online: 22 June 2017

© The Author(s) 2017. This article is an open access publication

Abstract Recently, food recommender systems have received increasing attention due to their relevance for healthy living. Most existing studies on the food domain focus on recommendations that suggest proper food items for individual users on the basis of considering their preferences or health problems. These systems also provide functionalities to keep track of nutritional consumption as well as to persuade users to change their eating behavior in positive ways. Also, group recommendation functionalities are very useful in the food domain, especially when a group of users wants to have a dinner together at home or have a birthday party in a restaurant. Such scenarios create many challenges for food recommender systems since the preferences of all group members have to be taken into account in an adequate fashion. In this paper, we present an overview of recommendation techniques for individuals and groups in the healthy food domain. In addition, we analyze the existing state-of-the-art in food recommender systems and discuss research challenges related to the development of future food recommendation technologies.

Keywords Recommender systems · Healthy food · User preferences · Nutritional needs · Eating behavior

✉ Thi Ngoc Trang Tran
ttrang@ist.tugraz.at

Müslüm Atas
muesluem.atas@ist.tugraz.at

Alexander Felfernig
alexander.felfernig@ist.tugraz.at

Martin Stettinger
martin.stettinger@ist.tugraz.at

¹ Applied Software Engineering Group, Institute for Software Technology, Graz University of Technology, Graz, Austria

1 Introduction

According to the prediction of the World Health Organization¹, the quantity of overweight adults all over the world has reached an alarming number with 2.3 billions by 2015. More significantly, overweight and obesity also cause many chronic diseases (Robertson 2004). An appropriate dietary intake is considered as an important factor to improve overall well-being. Although most people are aware of the importance of healthy eating habits, they usually tend to neglect appropriate behaviors because of busy lifestyles and/or unwillingness to spend cognitive effort on food preparation. Those problems prevent users from a healthy food consumption (Van Pinxteren et al. 2011). Hence, recommender systems are investigated as an effective solution in order to help users to change their eating behavior and to aim for healthier food choices.

However, food and diet are complex domains bringing many challenges for recommendation technologies. For making recommendations, thousands of food items/ingredients have to be collected. Besides, because foods/ingredients are usually combined with each other in a recipe instead of being consumed separately, this exponentially increases the complexity of a recommender system (Freyne and Berkovsky 2010). Furthermore, food recommender systems not only recommend food suiting users' preferences, but also suggest healthy food choices, keep track of eating behavior, understand health problems, and persuade to change user behavior.

While many existing recommender systems mainly target individuals, there is a remarkable increase of recommender systems which generate suggestions for groups. Some early systems were developed in a variety of domains, such as, *group web page* recommendation (Lieberman et al. 1999), *tour packages* for groups of tourists (Ardissono et al. 2003), *music tracks* and *playlists* for large groups of many listeners (Crossen et al. 2002), *movies* and *TV programs* for friends and family (O'Connor et al. 2001; Yu et al. 2006). Group scenarios are especially popular in the *food domain* in which a group of family members, friends or colleagues wants to make a party or simply have a meal together. However, the complexity significantly increases when food recommender systems need to take into account the preferences of all group members and strategies for achieving the consensus within group members.

In this paper, we summarize existing research related to food/recipe recommender systems which give recommendations on the basis of considering the users' preferences as well as their nutritional needs. In this context, we also discuss scenarios for applying group recommender systems in the healthy food domain. An overview of some research related to the application of recommender systems in the healthy food domain is provided in Table 1.

The contributions of this paper are the following. *First*, we provide a short overview of recommendation approaches for individuals. *Second*, we discuss *group decision making* issues which have an impact on the development of group recommendation technologies. *Third*, on the basis of categorizing food recommender systems, we analyze how well those systems can help individuals or groups to choose healthy food which best fits their preferences and health situations. *Finally*, we point out some challenges of food recommender systems with regard to *user information*, *recommendation algorithms*, *changing eating behaviors*, *explanations provision*, and *group decision making* as topics for future work.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of basic recommendation techniques for individuals and groups. In Section 3, we summarize

¹<http://www.who.int>

Table 1 A summary of state-of-the-art of recommender systems in the healthy food domain (CF: Collaborative filtering recommender systems, CB: Content-based recommender systems)

Food recommender systems	Papers	Recommendation approaches	Functionality
Type 1: Considering user preferences	(El-Dosuky et al. 2012)	<i>Knowledge-based recommendation</i>	Proposing a framework for a Personalized nutrition service with knowledge-based recommendation.
	(Freyne and Berkovsky 2010)	<i>CB, CF, Hybrid recommendation</i>	Predicting item ratings by breaking down recipes into ingredients and vice versa.
	(Freyne et al. 2011)	<i>CB, CF, Hybrid recommendation</i>	Improving the quality of recommendations by using machine learning techniques and an understanding of user reasoning.
	(Svensson et al. 2000)	CF	Developing an on-line grocery store to provide users with recipe recommendations by analyzing the social navigation in groups.
	(Elahi et al. 2015)	<i>Matrix factorization</i>	ChefPad - Generating food recommendations by eliciting users' long-term and short-term preferences.
Type 2: Considering nutritional needs of users	(Kuo et al. 2012)	<i>Graph-based recommendation</i>	Recommending sets of recipes by using user-specified ingredients.
	(Ueta et al. 2011)	<i>Goal-oriented recipe recommendation</i> which suggests the right type of nutrient to treat users' health problems	Helping users to deal with health problems (e.g., acne)
Type 3: Balancing between user preferences and nutritional needs of users	(Aberg 2006)	<i>Hybrid recommendation (CB & CF), Constraint-based recommendation</i>	Meal Planning System - Aiding the elderly to deal with malnutrition problems and change the food consumption behavior.
	(Elsweiler et al. 2015)	CB	Applying different approaches to bring healthiness aspects into recommender systems.
Type 4: Food recommender systems for groups	(Berkovsky and Freyne 2010)	<i>CF, group-based recommendation</i>	Applying different aggregation strategies and user weighting models to generate recipe recommendations to a group of users.
	(Elahi et al. 2014)	<i>Group recommendational recommendation, Critiquing-based conversational recommendation</i>	ChefPad - Generating food recommendations to groups by exploiting users' tags and ratings.

existing studies on food recommender systems for single users and categorize them according to different criteria, such as *preferences*, *nutritional needs*, *health problems*, and *eating behaviors of users*. Besides, in this section we also discuss some research related to food recommender systems in group scenarios. Research challenges for food recommender systems are discussed in Section 4. The paper is concluded with Section 5.

2 Recommender systems

Due to heavy information overloads triggered by the Internet, extracting/finding valuable information becomes increasingly difficult. In this context, recommender systems became an effective tool to extract useful information and deliver it in an efficient way. A recommender system predicts the preferences of users for unrated items and recommends new items to users. Along with the benefits of recommender systems, developing new recommendation approaches and including them in different fields rise extremely. The following subsections present an overview of recommendation techniques for individuals and groups.

2.1 Recommendation techniques for individuals

According to Burke et al. (2011) and Burke (2000), a recommender system can be defined as follows: “Any system that guides a user in a personalized way to interesting or useful objects in a large space of possible options or that produces such objects as output”. Recommender systems are intensively applied for the purpose of recommending products and services (e.g., *movies*, *books*, *digital cameras*, and *financial services*) which best meet users’ needs and preferences. Recently, in the healthy food domain, recommender systems have been discovered as a potential solution to help users to cope with the vast amount of available data related to foods/recipes. Many different techniques have been proposed for making personalized recommendations and these will be discussed in the followings.

Collaborative filtering recommender systems (CF) CF became one of the most researched techniques of recommender systems. The basic idea of CF is to use the wisdom of the crowd for making recommendations. First of all, a user rates some given items in an implicit or explicit fashion. Then, the recommender identifies the *nearest neighbors* whose tastes are similar to those of a given user and recommends items that the nearest neighbors have liked (Ekstrand et al. 2011). CF is usually implemented on the basis of the following approaches: *user-based* (Asanov 2011), *item-based* (Sarwar et al. 2001), *model-based approaches* (Koren et al. 2009), and *matrix factorization* (Bokde et al. 2015).

Content-based recommender systems (CB) These systems can make a personalized recommendation by exploiting information about available item descriptions (e.g., genre and director of movies) and user profiles describing what the users like. The main task of a CB system is to analyze the information regarding user preferences and item descriptions consumed by the user, and then recommend items based on this information. Research in this area primarily focused on recommending items with textual content, such as *web-pages* (Pazzani et al. 1996), *books* (Mooney and Roy 2000), and *documents* (Lang 1995). There are different approaches applied to make recommendations to users, such as *Information Retrieval* (Balabanović and Shoham 1997) or *Machine Learning algorithms* (Mooney and Roy 2000).

Knowledge-based recommender systems (KBS) KBS are recognized as a solution for tackling some problems generated by classical approaches (e.g., *ramp-up problems* (Burke 2000)). Moreover, these systems are especially useful in domains where the number of available item ratings is very low (e.g., apartments, financial services) or when users want to define their requirements explicitly (e.g., “the color of the car should be white”). There are two main approaches for developing knowledge-based recommender systems: *case-based recommendation* (Bridge et al. 2005) and *constraint-based recommendation* (Felfernig and Burke 2008). In addition, *critiquing-based recommendation* is considered as a variant of case-based recommendation. This approach uses users’ preferences to recommend specific items, and then elicits users’ feedback in the form of critiques for the purpose of improving the recommendation accuracy (Burke 2000). There are four basic steps in a knowledge-based recommendation setting:

- *Requirement specification*: Users can interact with a recommender system for specifying their requirements.
- *Repair of inconsistent requirements*: If the recommender can not find a solution, it suggests a set of repair actions, i.e., it proposes alternatives to user requirements ensuring the identification of a recommendation (Felfernig et al. 2011).
- *Presentation of results*: A set of alternatives is delivered to the user. These are usually presented as a ranked list according to the item utility for the user (Felfernig et al. 2006).
- *Explanation*: For each presented alternative, the user can activate a corresponding explanation to understand why a specific item has been recommended (Felfernig et al. 2006).

Hybrid recommender systems (HRS) HRS are based on the combination of the above mentioned techniques. According to Ricci et al. (2010): “A hybrid system combining techniques A and B tries to use the advantages of A to fix the disadvantages of B”. For instance, CF methods have to face the *new-item* problem. Whereas, CB approaches can tackle this problem because the prediction for new items is usually based on available descriptions of these items. Burke (2002) presents some hybrid approaches which combine both CF and CB, including *weighted, switching, mixed, feature combination, cascade, feature augmentation, and meta-level*.

2.2 Recommendation techniques for groups

Research on recommender systems as discussed in Section 2.1 only focuses on recommending items to individual users. However, in reality, there is a high probability of situations where recommender systems should support a group of users. For instance, a tourist package for a group of friends or a Christmas party destination for all colleagues in a company. In such situations, *Group Recommender Systems* (Masthoff 2011) are considered as an optimal solution. In this subsection, we present an overview of some basic aspects of group-based recommendation.

Aggregation strategies The main problem that group recommender systems need to solve is how to aggregate preferences based on information about the interests of each individual. Masthoff (2011) presented many different strategies for merging individual user profiles into a group profile. These strategies can be also used for combining individual recommendations into group recommendations. Mostly used aggregation strategies for group

recommendations are *least misery* (O'Connor et al. 2001), *average* (Ardissono et al. 2003), and *multiplicative* (Masthoff 2004).

Group formation In group recommendation scenarios, group creation and group maintenance are important steps that should be addressed. Groups can be built intentionally by an explicit definition from the users (Smith et al. 1998) or unintentionally by an automatic identification from the system (McCarthy and Anagnost 1998). Within a group, roles of group members can be conferred differently according to their importance level within the group (Cantador and Castells 2012; Berkovsky and Freyne 2010). For instance, in a holiday planning scenario of a family, parents have more influence on choosing a tourism destination than children.

Group recommendation approaches Group recommendations are mostly determined by using an *aggregated model* or an *aggregated prediction* (Jameson and Smyth 2007).

- *Aggregated model* generates predictions for a group on the basis of aggregating individual user preferences into a *group profile*. The group recommendation process can be executed in three steps: *First*, users with similar preferences will be classified in subgroups. *Next*, the available items will be ranked based on each subgroup preference. *Finally*, related items in subgroups are merged to get the ranking for the whole group. This approach was applied in some well-known systems, e.g., MUSICFX (McCarthy and Anagnost 1998) and INTRIGUE (Ardissono et al. 2003) for the purpose of supporting a group of users to choose suitable alternatives.
- *Aggregated prediction* firstly computes the recommendation for each group member and then computes the intersection of individual recommendations to get the common recommendations for whole the group. For instance, POLYLENS (O'Connor et al. 2001) generates a ranked list of movies for each group member by using a classic CF approach. After that, the individual ranked lists are merged according to the *least misery* strategy, i.e., group's happiness is the minimum of the individual members' happiness scores.

Group decision making After forming groups, discovering some constraints within a group is an important phase which facilitates a recommender to make group recommendations. For instance, in the scenario of recommending recipes to a group of family members, because of the *seafood allergy* of one family member, recipes including *shrimp* or *sea-crab* might not be recommended to the whole group. In addition, in group recommender systems, sometimes, knowing the preferences of other group members will have an impact on the decisions of other users. TRAVEL DECISION FORUM (Jameson 2004) provides an interaction environment which allows members to optionally view (or copy) the preferences already specified by other group members. The preference visibility helps users to save time and minimize conflicts generated in the decision making process (Jameson 2004). However, in some decision scenarios, the insight to individual preferences of all group members can deteriorate the quality of the decision outcome (Stettinger et al. 2015). This issue was known as an *anchoring effect* (Adomavicius et al. 2011; Felfernig 2014) which is responsible for decisions biased by a shown reference value. In the context of group decision scenarios, the anchoring effect can be controlled by not completely disclosing the preferences of other group members in early stages of the decision process (Felfernig et al. 2012). In CHOICLA (Stettinger et al. 2015), a user can solely see the summary of all given ratings of other group

members for a specific alternative after giving his/her rating. Seeing the summarized rating prevents all users from statistical inferences, which can influence on the quality of decision processes.

Until now, group recommendations are still a novel area compared to research on individual recommendations (Masthoff 2011). There are still open issues on group decision making which need to be resolved in the future research, such as *bundle recommendations*, *intelligent user interface design*, *group aggregation strategies for cold-start problems* (Masthoff 2011), *consensus achievement within group members*, and *counteracting decision biases* in group decision processes (Felfernig et al. 2014a).

3 Food recommender systems

“Where should we go for lunch?” or “What should we eat for dinner?” are usual questions we have to answer very frequently. While many recommender systems only tried to match users’ preferences to music, movie, or book domains, recently they also have been applied in the food domain in order to give reliable answers to the above questions. For instance, *RecipeKey*² is a food recommender system that filters recipes on the basis of considering favorite ingredients, existing food allergies, and item descriptions (e.g., meal type, cuisine, preparation time, etc.) chosen by users.

In relation to the food consumption these days, it is noticeable that there has been an increase of lifestyle-related illnesses, such as diabetes and obesity, which are the cause of many chronic diseases (Robertson 2004). This problem can be improved by applying appropriate dietary (Knowler et al. 2002). In this context, food recommender systems are also investigated as a potential means to aid people nourish themselves more healthily (Elsweiler et al. 2015). It makes sense to utilize food recommender systems as a part of a strategy for changing eating behaviour of users. In this case, food recommender systems not only learn users’ preferences for ingredients and food styles, but also select healthy food by taking into account health problems, nutritional needs, and previous eating behaviors.

As mentioned in Mika (2011), there are two types of food recommender systems. The first type (type 1) recommends *healthier recipes* or *food items* which are most similar to the ones the user liked in the past. The second type of recommender system (type 2) only recommends to users those items which *have been identified beforehand by health care providers*. In addition, in this section, we also discuss two other types of food recommender systems (type 3 and type 4) which consider other scenarios when making recommendations. Type 3 generates recommendations on the basis of *considering both above criteria* for the purpose of balancing between the food users like and the food users should consume. All these types of recommender systems are primarily designed for individual users. Type 4 represents *group recommendations* in which food items are consumed by groups of users rather than by individuals. These four types of food recommender systems will be made more explicit and will be discussed in more detail in the following subsections.

3.1 Type 1: Considering user preferences

In the healthy food domain, learning user tastes is recognized as a crucial pre-requisite step in order to suggest dishes that users will like. All research discussed in this subsection aims

²<http://www.recipekey.com>

for recommending food items or menus to individual users on the basis of exploring user tastes. Most of them use popular recommendation techniques (Freyne and Berkovsky 2010; Svensson et al. 2000; El-Dosuky et al. 2012), and/or combine with different techniques in order to improve the quality of recommendation (Elahi et al. 2015; Kuo et al. 2012) (see Table 1).

First of all, we present a food recommender system (El-Dosuky et al. 2012) with a simple scenario which only recommends individual food items to users. The authors use TF-IDF (Term Frequency-Inverse Document Frequency) term extraction method for creating the user profile and apply some computations for identifying the similarity between a recipe and the user profile. In addition, healthy and standard food databases, which have been extracted from the United States Department of Agriculture³ (USDA), are incorporated into the knowledge base. The knowledge base is a domain ontology consisting of classes, relationships, and instances of classes. For getting a recommendation, each user manually rates the food items of a specific category (e.g., fruits, vegetables, meat, etc.) as relevant or non-relevant for his/her interest. After that, the recommender will compute the similarity between the food items and the previously computed user profile. If the similarity value is higher than a predefined threshold, the food item is recommended, otherwise it gets ignored.

In another research, Freyne et al. (Freyne and Berkovsky 2010) use a CB algorithm to predict the rating value for a target recipe on the basis of exploiting the information of corresponding ingredients included in this recipe. The prediction process includes the following steps:

- Break down an unrated target recipe r_t into ingredients $ingr_1, ingr_2, \dots, ingr_n$.
- Assign the rating value for each ingredient in the target recipe r_t according to (1) as shown below. Particularly, the rating value of the user u_a for a specific ingredient $ingr_i$ in the target recipe r_t (i.e., $rat(u_a, ingr_i)$) is calculated by using rating values of the user u_a for all other recipes r_l which contain the ingredient $ingr_i$ (i.e., $rat(u_a, r_l)$). The value l mentioned in (1) is the number of recipes containing $ingr_i$.

$$rat(u_a, ingr_i) = \frac{\sum_{l \text{ s.t. } ingr_i \in r_l} rat(u_a, r_l)}{l} \quad (1)$$

- Predict the rating value of the user u_a for the target recipe r_t (i.e., $pred(u_a, r_t)$) based on the average of all the rating values of all ingredients $ingr_1, \dots, ingr_j$ included in this recipe (see (2)).

$$pred(u_a, r_t) = \frac{\sum_{j \in r_t} rat(u_a, ingr_j)}{j} \quad (2)$$

Recipes with a high predicted rating value will be recommended to user u_a . An illustration for predicting a rating value for a target recipe is presented in the following example:

Let us assume that $recipe_1$ is a recipe which has not been rated by user u_a . It includes 3 ingredients, i.e., $ingr_1, ingr_2$, and $ingr_3$. $ingr_1$ is included in $recipe_4$ and $recipe_2$, $ingr_2$ is included in $recipe_3$, and $ingr_3$ is included in $recipe_2$ and $recipe_3$. Rating values of user u_a for $recipe_2, recipe_3$, and $recipe_4$ are respectively 4, 2, and 5 (see Fig. 1).

³<https://ndb.nal.usda.gov/>

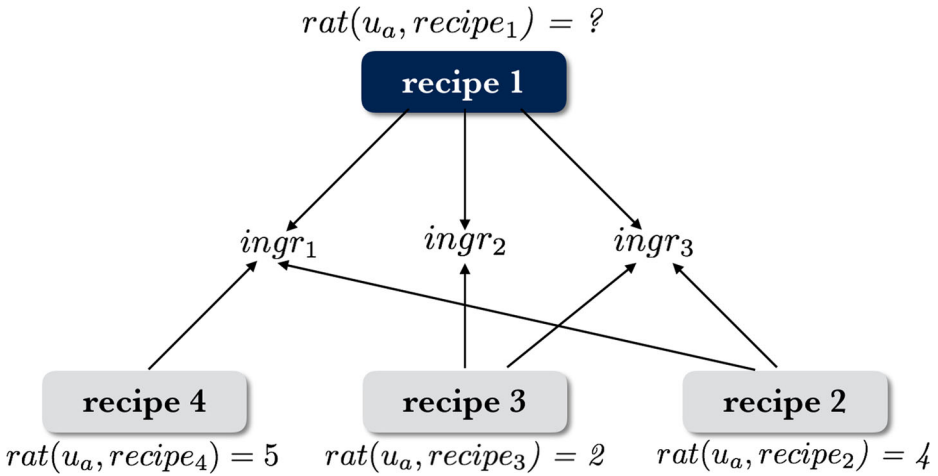


Fig. 1 Predicting the rating value for a target recipe by using a CB algorithm proposed by Freyne and Berkovsky (2010)

According to (1), rating values for ingredients of $recipe_1$ will be evaluated as follows:

$$\begin{aligned}
 rat(u_a, ingr_1) &= \frac{rat(u_a, recipe_4) + rat(u_a, recipe_2)}{2} = \frac{5 + 4}{2} = 4.5 \\
 rat(u_a, ingr_2) &= rat(u_a, recipe_3) = 2 \\
 rat(u_a, ingr_3) &= \frac{rat(u_a, recipe_2) + rat(u_a, recipe_3)}{2} = \frac{4 + 2}{2} = 3
 \end{aligned}$$

Prediction value of $recipe_1$ for user u_a is calculated by applying (2) as follows:

$$\begin{aligned}
 pred(u_a, recipe_1) &= \frac{rat(u_a, ingr_1) + rat(u_a, ingr_2) + rat(u_a, ingr_3)}{3} = \frac{4.5 + 2 + 3}{3} \\
 &= 3.166
 \end{aligned}$$

Recently, some new approaches have been included to food recommender systems, such as *using labels for different clusters of users* (Svensson et al. 2000), *active learning algorithms*, and *matrix factorization* (Elahi et al. 2015). Particularly, in Svensson et al. (2000), the authors design an on-line food shop for the purpose of suggesting kinds of food that should be purchased by users. Based on recipes which users have chosen before, user groups are labeled and named according to their content, for instance, “*Meat lovers*”, “*Vegetarians*”, “*Spice lovers*”, etc. The recommended recipes are determined on the basis of three different characteristics chosen by users: *user groups*, *food categories* (e.g., fish, oriental, Italian, red meat, chicken), and *ingredients* (e.g., rice, spaghetti, curry, tomatoes). Users select recipes from the recommendation list and put them into a shopping basket. Then, all ingredients of chosen recipes are automatically added to the list of items which is delivered to a user’s doorstep. In addition, in order to enhance the social interaction for recipes, some additional features (e.g., the average rating value or comments from other users) are added into each recommended recipe.

Elahi et al. (2015) propose a food recommender system by using an *active learning algorithm* and *matrix factorization*. This research provides users with a complete human-computer interaction for the purpose of collecting *long-term user preferences* in terms of

recipe ratings and tags. In addition, when requesting recommendations, users are required to provide *short-term preferences* referring to ingredients which they want to cook or to include in the meal. Then, the system utilizes both types of user preferences to make recommendations. The long-term preferences are exploited by a *Matrix Factorization rating prediction model* designed to consider both user tags and ratings. Each user and each recipe are modeled by vectors that represent their latent features. The rating value of a user for a specific item is estimated by computing the inner product of the user and item vectors. With short-term preferences, the system filters recipes according to the current user preferences. The recipes with highest rating values are recommended to the user.

While most of existing research in the food domain only focuses on making recommendations on food items or recipes, there is a need for users to plan menus with the combination of many recipes into complete meals. With this idea, Kuo et al. (2012) propose an intelligent menu planning mechanism which suggests a set of recipes by using a *graph-based algorithm*. First, an undirected recipe graph is constructed, where *each node* is a recipe possessing a set of ingredients, *each edge* represents the relationship between two recipes, and *edge weight* represents the distance between two recipes (see Fig. 2). The weight of each edge connecting two different recipes describes *the cost* of a menu which includes these two recipes. The lower the weight the higher the probability two recipes co-occur in a menu. For instance, in Fig. 3, the recipe “*Italian Bread*” has co-occurrence relationship with five recipes, i.e., “*Tiramisu*”, “*Lasagne*”, “*Mozzarella, Tomato, and Basil Salad*”, “*Caesar Salad*”, and “*Stuffed Shells*”. Among these five recipes, “*Tiramisu*” has the highest co-occurrence relationship with “*Italian Bread*” since the weight of their edge is lowest (i.e., 0.11). Whereas, “*Stuffed Shells*” has the lowest relationship with “*Italian Bread*” because the weight of their edge is highest (i.e., 0.5).

In addition, the cost of a menu is also defined as a weighted sum of edges of the minimum spanning tree on the induced sub-graph. From that, a *menu plan* is created by choosing a set of recipes which contains all query ingredients (i.e., ingredients requested by users) and the menu cost is minimal. For instance, in Fig. 2, with query ingredients {*tomato, flour, basil*}, we can find many different sets of recipes, for instance, {“*Mozzarella, Tomato and*

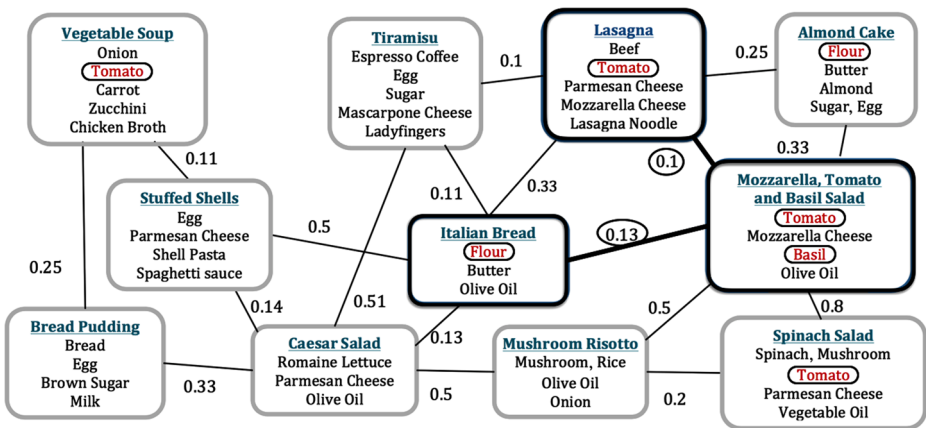


Fig. 2 An example of recipe graph G for menu planning (Kuo et al. 2012). “*Tomato, flour, basil*” (ingredients shown with black borders) are query ingredients. The recommended menu plan is a set of recipes {“*Mozzarella, Tomato and Basil Salad*”, “*Lasagne*”, “*Italian Bread*”} (nodes shown with black frames) which the total menu cost is minimal (i.e., 0.23)

Adults (years old)	Reference values for energy intake in kcal/day					
	PAL-value (1.4)		PAL-value (1.6)		PAL-value (1.8)	
	male	female	male	female	male	female
15 to 19	2600	2000	3000	2300	3400	2600
19 to 25	2400	1900	2800	2200	3100	2500
25 to 51	2300	1800	2700	2100	3000	2400
51 to 65	2200	1700	2500	2000	2800	2200
> 65	2100	1700	2500	1900	2800	2100

Fig. 3 Reference values for nutritional intake. Bonn. 2. Edition, 1. Volume (2015) published by German Nutrition Association, Austrian Nutrition Association, and Swiss Nutrition Association

Basil Salad”, *“Lasagna*”, *“Italian Bread*”}, {*“Mozzarella, Tomato and Basil*”, *“Lasagne*”, *“Almond cake*”}, {*“Mozzarella, Tomato and Basil*”, *“Italian Bread*”, *“Spinach Salad*”}, etc. However, the first set {*“Mozzarella, Tomato and Basil Salad*”, *“Lasagna*”, *“Italian Bread*”} will be recommended to users because its total menu cost is minimal (i.e., 0.23).

3.2 Type 2: Considering nutritional needs of users

Nowadays, unhealthy eating habits and imbalanced nutrition increase possibilities of people having obesity and other dietary-related conditions such as diabetes, hypertension, etc. As a treatment or preventive measure, nutritionists or dietitians usually recommend regular exercises and design individualized meal plans for their patients. Unfortunately, these nutrition experts are overloaded with too many patients to manually tailor an individualized meal plan for each user. That is where food recommender systems can be used as an intelligent nutrition consultation system. In this subsection, we provide a discussion of recommender systems that takes into account nutritional needs (see Table 1).

First, we discuss a simple recommendation scenario showing how menu items can be recommended to users on the basis of considering their *nutritional needs* as well as *health problems*. In this context, a user enters some personal information (e.g., *age, gender, occupation, physical activities, health problem*, etc.). This information is the basis for selecting food items which best fit the user’s nutritional needs. The following example will be an illustration of this scenario.

In a menu recommender system, we assume that there are 5 menus with corresponding information, e.g. *ingredients, calories, fat* (see Table 2). A user u_a enters the following information: *Age: 52, Gender: male, Occupation: office worker, Physical activity: walking (10 minutes/day), Health problem: cardiovascular*. For recommending appropriate menus to user u_a , the following steps should be performed:

- Step 1: An energy table from DACH⁴ (see Fig. 3) is used to estimate the amount of calories (in *kcal*) which the user u_a should get per day. The amount of calories intake per day for each person is estimated according to *age, gender* and *PAL* (Physical Activity

⁴<http://www.sge-ssn.ch>.

Table 2 A list of available menus with corresponding information

Menus	Main ingredients	Calories (kcal)	Fat(%)
<i>menu</i> ₁	Butter, chicken, potato, cucumber, cream, garlic, salt, pepper	2010	27
<i>menu</i> ₂	Pork, mushroom, broccoli, paprika, green onion, oil, salt, pepper	2200	30
<i>menu</i> ₃	Chicken, mushroom, salad, onion, olive oil, tomato, salt, pepper	1500	21
<i>menu</i> ₄	Beef, shrimp, tomato, garlic, egg, salt, pepper	2400	31
<i>menu</i> ₅	Pork, bean, tomato, pumpkin oil, salad, egg, salt, pepper	1700	25

Level) value. PAL value is categorized into 3 types:

+ PAL = 1.4: Is used for people who have exclusively sedentary lifestyles (such as *office workers, precision mechanics*) with very little or no strenuous leisure activity.

+ PAL = 1.6: Is used for people who have sedentary lifestyles, but an additional energy is required for long-time walking and standing activities, such as *laboratory assistants, students, production line workers*.

+ PAL = 1.8: Is used for people who have extensive lifestyles, for instance, *sellors, waiters, mechanics, artisans*.

In this example, user u_a is an office worker with very little physical activity (only 10 minutes/day for walking), that means his PAL value belongs to the first type. By looking up information regarding *age, gender and physical activity* from Fig. 3, we can find the daily calories intake for u_a is 2200 kcal.

- Step 2: Filtering menu with the amount of calories smaller or equal 2200 kcal/day.
- Step 3: Ranking filtered menus in the ascending order of fat (since u_a has heart disease, less fatty menu will be shown to him first).

In Table 2, we can see that *menu*₄ will not be added to the recommendation list because its calories is more than 2200 kcal. The list of recommended menu is ranked in the ascending order of fat (see Table 3).

For the purpose of improving health conditions of users, Ueta et al. (2011) propose a *goal-oriented recipe recommendation* in order to provide a list of dishes that contains the right type of nutrient to treat users' health problems. To do that, first of all, a user enters her health problem in natural language, for instance, "*I want to cure my acne*". Next, the system analyzes the user's request and identifies the keywords describing the health problem (e.g., *acne*). The noun is pushed into the co-occurrence database to search the nutrient co-occurring mostly with it. For instance, by searching the noun *acne* in the co-occurrence database, *pantothenic acid* is found as a nutrient component which can be used for curing *acne* because it co-occurs with "*acne*" more often than any other nutrients. Finally, the nutrients identified in the previous step are used to find dishes which are closest to those nutrients in a

Table 3 A list of recommended menus to user u_a

Menus	Main ingredients	Calories (kcal)	Fat(%)
<i>menu</i> ₃	Chicken, mushroom, salad, onion, olive oil, tomato, salt, pepper	1500	21
<i>menu</i> ₅	Pork, bean, tomato, pumpkin oil, salad, egg, olive oil, salt, pepper	1700	25
<i>menu</i> ₁	Butter, chicken, potato, cucumber, cream, garlic, salt, pepper	2010	27
<i>menu</i> ₂	Pork, mushroom, broccoli, paprika, green onion, oil, salt, pepper	2200	30

food database. This food database includes two sub-databases, ingredient nutrient database and nutritional information database for recipes. The *ingredient nutrient database* contains information about the nutritional value of each ingredient. The *nutritional information database* includes recipe types and the amount of nutrients contained in each recipe. The ingredients in each recipe are identified and then their nutritional elements are calculated using the ingredient nutrient database. When recommending recipes for users, the system also considers the daily nutrient intake of users. These requirements vary according to age and gender of users.

In a research related to dealing with malnutrition for the elderly, Aberg (2006) proposes a menu-planning tool which is required to take into account the following user-related information:

- Dietary restrictions, such as allergic ingredients.
- Nutritional values, such as the amount of fat or protein contained in a recipe.
- Preparation time of a meal.
- Preparation difficulty of a meal.
- Cost of necessary ingredients for a meal.
- The availability of ingredients for a meal.
- The variety of meals in terms of used ingredients and meal category.
- User food preferences, i.e., rating of a user for a certain recipe.

To be able to consider all these requirements, the author apply a hybrid design on the basis of combining *CF*, *CB*, and *constraint-based recommendation*. *CF* recommendation uses the ratings to predict the user’s feedback on unrated recipes. With the *CB approach*, the author uses *XML-based mark-up* language to represent the needed information for the recipes in the database. A *constraint-based recommendation approach* represented as a constraint satisfaction problem is used to construct optimal meal plans. A constraint satisfaction problem is modeled with two different approaches: *parameter-based* approach and *recipe-based* approach. However, the author did not mention in detail the *recipe-based approach*. Therefore, in this paper, we solely discuss the *parameter-based approach* and the details of this approach is presented in Table 4. A prototype is developed to offer a meal-plan recommendation to users in a certain time period. Users can switch between the top-5 meal plans and give ratings on recommended recipes or create special settings for a meal.

For demonstration purposes, we propose an example of a constraint-satisfaction problem, which is similar to a parameter-based approach (Aberg 2006) to suggest a recipe on the basis of taking into account user’s preferences. In this example, we assume that variables are used for representing the parameters of a recipe, such as *time*, *cost*, *energy*, *protein*, *allergies*, *disease*, where: *time* (in *minutes*) is the preparation time of a recipe, *cost* (in *euro*) is the cost of a recipe, *energy* (in *kcal*) is the nutritional value of a recipe, *protein* (in %) is the percentage of protein contained in a recipe, *allergies* represents a set of allergic ingredients of users, and *disease* represents health problems of users. Each variable has corresponding domain definition, for instance, $\text{dom}(\text{time}) = [1..60]$. In addition, a knowledge base *CKB* (Constraint Knowledge Base) includes constraints used for describing the knowledge base. For instance, $\text{time} < 60$ denotes the fact that preparation time of a recipe should be lower than 60 minutes. *PREF* is the set of user preferences, which should be consistent with *CKB* such that a corresponding solution can be identified.

- $V = \{\text{time}, \text{cost}, \text{ingredients}, \text{energy}, \text{protein}, \text{allergies}, \text{diseases}\}$
- $D = \{\text{dom}(\text{time}) = [1..60], \text{dom}(\text{cost}) = [1..100], \text{dom}(\text{energy}) = [1..3000], \text{dom}(\text{protein}) = [1..100], \text{dom}(\text{allergies}) = [\text{milk}, \text{egg}, \text{peanut}, \text{seafood}, \text{wheat}],$

Table 4 The constraint satisfaction problem modeled with a *parameter-based approach* (Aberg 2006)

Constraint satisfaction problem	Parameter-based approach
Variables	Time, cost, energy, protein
Variable domains	Variable domains are defined on the basis of existing values in the recipe database (e.g., $cost = [1..100]$ represents the cost of a recipe can be from 0 to 100 euros).
Knowledge base	<p><i>Hard constraints:</i></p> <ul style="list-style-type: none"> – The constraint: $\{allergies = seafood \Rightarrow ingredients \neq sea - crab\}$ represents the knowledge that if a user is allergic to seafood then sea-crab should not be included into recommended recipes. <p><i>Soft constraints:</i></p> <ul style="list-style-type: none"> – For the variety of recipes, recipes having many similar ingredients to the previous meals will not be chosen (e.g., <i>beef</i> and <i>potato</i> will not be chosen for dinner today because they were already consumed on lunch); – Recipes with high predicted rating will have higher probability to be recommended to users.

$dom(diseases) = [diabetes, cardiovascular, parkinson, digestion, alzheimer, osteoarthritis, osteoporosis],$

$dom(ingredients) = [vegetables, shrimp, sea - crab, fish, pork, beef, chicken, spices, butter, cheese, fruits] \}$

- $CKB = \{c_1 : time < 60, c_2 : cost < 100, c_3 : energy < 3000, c_4 : protein < 35\%, c_5 : disease = cardiovascular \Rightarrow protein < 30, c_6 : allergies = seafood \Rightarrow ingredients \neq sea - crab \}$
- $PREF = \{pref_1 : time < 30, pref_2 : cost < 50, pref_3 : energy = 2200, pref_4 : protein = 25\%, pref_5 : allergies = seafood, pref_6 : disease = cardiovascular \}$

On the basis of the constraint satisfaction problem as specified above, one solution can be determined for users: $\{time = 25, cost = 40, ingredients = \{vegetables, chicken, spices, fruits\}, energy = 2200, protein = 25\%\}$.

3.3 Type 3: Balancing between user preferences and nutritional needs of users

Considering either user preferences or nutritional needs in an isolated fashion sometimes leads to sub-optimal recommendations of food items. For instance, if recommenders only take into account user preferences then bad eating habits would also be encouraged. On the contrary, if only nutritional needs are considered then proposed food items sometimes will not be attractive to users. Therefore, considering both, user preferences and nutritional needs seems to provide the best solution since users receive more relevant recommendations, they become more interested and increasingly engaged in using them.

We now discuss a simple recommendation scenario showing how a food recommender system can suggest menu items on the basis of considering both user preferences and nutritional needs. In this example, we assume the existence of a menu table as shown in Table 2. A user u_a provides personal information as follows: *Age*: 52, *Gender*: male, *Occupation*: office worker, *Physical activity*: walking (10 minutes/day), *Health problem*: cardiovascular, *Favorite ingredients*: tomato. In this scenario, the recommender system considers both

ingredients preferred by the user u_a and further user-related information (e.g., *age, gender, occupation, physical activity, and health problem*). The list of recommended menus is created by performing the following steps:

- *Step 1:* Estimating the daily amount of calories for user u_a by looking up the energy table shown in Fig. 3. The user u_a is an office worker and has very little physical activity per day (only 10 minutes/day for walking), hence the nutrient intake of the user u_a is 2200 kcal.
- *Step 2:* Filtering menus from Table 2 which contain lower or equal 2200 kcal of calories, and include favourite ingredient “*tomato*”.
- *Step 3:* Ranking filtered menus in the ascending order of fat (because u_a has vascular disease, less fatty menus will be shown to him first).

After accomplishing these steps, there are two menus, i.e., $menu_3$ and $menu_5$ will be recommended to user u_a (see Table 5).

Also for the purpose of balancing users’ preferences and nutritional needs, Elswailer et al. (2015) propose two approaches to integrate nutritional aspects into recommendations.

- *The first approach* figures out trade-offs between giving the user some foods she really likes and some foods which are really healthy to her. This approach is implemented by using the following steps. First, a prediction algorithm estimates the top recipes for the user, i.e., a set of recipes with predicted probability above a certain threshold. Next, the amount of calories and fat per gram for each recipe in the chosen set is calculated. Finally, meals with less fat or calories per gram will be chosen for the final recommendation.
- *In the second approach*, instead of recommending individual meals, this approach proposes complete meal plans, which are generated not only based on the users’ food preferences but also conform to daily nutritional guidelines (Harvey and Elswailer 2015). For making recommendations, the user provides information regarding his/her preferences by rating a number of recipes in the system on a 5-star rating scale. In addition, the “Recommender” also takes into account additional users’ personal information, such as *height, weight, age, daily activity level, and goal (lose, gain, or maintain weight)* in order to calculate the nutritional needs. The nutritional requirements of users are calculated by using an updated version of the Harris Benedict equation (Roza and Shizgal 1984). After that, the “Recommender” predicts ratings for unrated recipes and sends a ranked list of recipes with high ratings (e.g., 4 or 5 stars) to the “Planner”. The “Planner” takes *top-n recipes* from the ranked list of recipes and splits them into two separated sets: one for breakfasts and one for main meals. A full search is performed to find all combinations of these recipes in the sequence {Breakfast, Main meal, Main meal} which meets the target nutritional needs. For instance, {*Muesli Breakfast Muffins, Catalan Chickpeas, Chicken Cacciatore*} (Harvey and Elswailer 2015) represents a complete menu recommended to users, where *Muesli Breakfast Muffins* is for

Table 5 A list of menus recommended to user u_a on the basis of considering his favorite ingredient (i.e., *tomato*) and his nutritional needs

Menus	Main ingredients	Calories (kcal)	Fat(%)
$menu_3$	Chicken, mushroom, salad, onion, olive oil, tomato, salt, pepper	1500	21
$menu_5$	Pork, bean, tomato, pumpkin oil, salad, egg, salt, pepper	1700	25

breakfast, *Catalan Chickpeas* for lunch, and *Chicken Cacciatore* for dinner. Combinations with the same recipes can not be repeated, for instance, $\{r_1, r_2, r_3\}$ and $\{r_1, r_3, r_2\}$ are considered as only one menu plan.

Although two of the above proposed approaches are helpful for supporting the trade-off between users' preferences and healthy foods, the suitability of combining separate ingredients into a complete meal should be considered in more detail to make an appealing meal plan (Elsweiler et al. 2015).

3.4 Type 4: Food recommender systems for groups

As mentioned above, in many real-world scenarios, recipe and food consumption are good examples of a group activity, for instance, a birthday party with friends, daily meals with family members (Elahi et al. 2014). In these scenarios, recommendations should be tailored to the entire group in order to assure the maximum satisfaction of each member and the group as a whole.

CF is one of the most widely used recommendation techniques and also applied in many group recommender systems (McCarthy et al. 2006; O'Connor et al. 2001). In the food domain, Berkovsky and Freyne (2010) investigate the applicability of two CF recommendation strategies for the purpose of discovering which strategy is most relevant when making CF recommendations for a group. The authors discuss two group-based recommendation strategies as the following:

- **Aggregated models strategy.** First, this strategy computes a rating $rat(f_a, r_i)$ for a family f_a and recipe r_i by aggregating the individual ratings $rat(u_x, r_i)$ of family members $u_x \in f_a$ who rated recipe r_i according to their relative weight $\omega(u_x, f_a)$ (see (3)). The authors add weights into the rating calculation process for the purpose of allowing some users in a family to have more influence on the group decision than others. For instance, parents have more influence on the group decision than children, therefore weights assigned for parents are higher than the children's ones. The details of weighting models will be presented in the next paragraph.

$$rat(f_a, r_i) = \frac{\sum_{x \in f_a} \omega(u_x, f_a) rat(u_x, r_i)}{\sum_{x \in f_a} \omega(u_x, f_a)} \quad (3)$$

After that, CF is applied to the family model. Particularly, a prediction $pred(f_a, r_i)$ for the whole family f_a and unrated recipe r_i is generated by computing similarity degree $sim(f_a, f_b)$ between the family f_a and all other families $f_b \in F$, and then aggregating all family's ratings $rat(f_b, r_i)$ for recipe r_i according to the similarity degree $sim(f_a, f_b)$ (see (4)).

$$pred(f_a, r_i) = \frac{\sum_{f_b \in F} sim(f_a, f_b) rat(f_b, r_i)}{\sum_{f_b \in F} sim(f_a, f_b)} \quad (4)$$

- **Aggregated predictions strategy.** First, this strategy generates individual predictions $pred(u_x, r_i)$ for user u_x and unrated recipe r_i by using the standard CF algorithm (see (5)). In this prediction, the degree of similarity $sim(u_x, u_y)$ between the target user u_x and all other users $u_y \in U$ is calculated according to (6) Freyne et al. (2011). Then, individual ratings $rat(u_y, r_i)$ of users who rated r_i are aggregated according to the

similarity degree $sim(u_x, u_y)$.

$$pred(u_x, r_i) = \frac{\sum_{y \in U} sim(u_x, u_y) rat(u_y, r_i)}{\sum_{i \in U} sim(u_x, u_y)} \tag{5}$$

where:

$$sim(u_x, u_y) = \frac{\sum_{i=1}^k (u_{x_i} - \bar{u}_x)(u_{y_i} - \bar{u}_y)}{\sqrt{\sum_{i=1}^k (u_{x_i} - \bar{u}_x)^2} \sqrt{\sum_{i=1}^k (u_{y_i} - \bar{u}_y)^2}} \tag{6}$$

where k is the number of items already rated by the user u_x and the user u_y .

After that, to generate the prediction $pred(f_a, r_i)$ for the whole family f_a and recipe r_i , individual predictions $pred(u_x, r_i)$ of family members $u_x \in f_a$ are aggregated according to their relative weight $\omega(u_x, f_a)$ (see (7)).

$$pred(f_a, r_i) = \frac{\sum_{x \in f_a} \omega(u_x, f_a) pred(u_x, r_i)}{\sum_{x \in f_a} \omega(u_x, f_a)} \tag{7}$$

Both *aggregated models strategy* and *aggregated predictions strategy* recommend a list of recipes to the whole family by considering the task of recommending *top-k* recipes, i.e., k recipes having the highest predicted ratings.

The evaluation results on MAE (Mean Absolute Error) show that *aggregated models strategy* are usually predominant to *aggregated predictions strategy* (Berkovsky and Freyne 2010). This means that individual models of users should be aggregated into a group model first and then using this model in the recommendation process.

- **Weighting models.** Inspired by allowing for some users to have more influence than others, the authors proposed four different weighting models when aggregating the data of individual users. Two first models (called *uniform model* and *role-based model*) assign pre-defined weights for users. Particularly, the *uniform model* uses the same weight for all group members. The *role-based model* weights users according to their role. For instance, there are two roles specified in a family party: *organizer* and *family member*. The weight for the *organizer* will be 2 because she is responsible for organizing the party as well as preparing food. Whereas, the weights for *family-members* are 1 because they are likely less important people. Two other models (called *role-based model* and *family-log model*) weight users according to their interactions with the content. The *role-based model* weights users according to their activities across the entire community. The activity of a certain user is predicted based on the number of ratings which (s)he rated for items. The *family-log model* weights users according to their activities in relation to other family members.

With the idea of combining individual user preferences into a group profile by using aggregation heuristics (Masthoff 2011) (e.g., *Least Misery*, *Average*, *Most Pleasure*, *Group Distance*, *Ensemble*, etc.), we discuss in this subsection a simple group recommendation scenario in the food domain to show how a group recommendation can be created.

Supposing that in a recipe recommender system we have a group with four users (e.g., *user1..4*) who rated 5 recipes (e.g., *recipe1..5*) by using a 5-star rating scale. We use *Least Misery* strategy (Masthoff 2011) to aggregate individual user preferences into a group profile as a whole. *Least Misery strategy* makes sense in recipe decision scenarios since it helps to minimize the misery within a group, i.e., recipes which are not liked or can not be consumed by at least one group member will not be recommended to the whole group. In our example, the *group rating value* for each recipe is the minimum of all ratings given by all group members (see Table 6). After that, the recipe having the highest group rating value

Table 6 An example of using *Least Misery* strategy to aggregate individual user preferences into a group profile. *recipe₁* is recommended to the group because its group rating value is highest

Recipes					
User	<i>recipe₁</i>	<i>recipe₂</i>	<i>recipe₃</i>	<i>recipe₄</i>	<i>recipe₅</i>
<i>user₁</i>	4	4	4	2	5
<i>user₂</i>	4	4	5	4	5
<i>user₃</i>	5	2	5	4	3
<i>user₄</i>	4	5	3	3	4
Group (Least misery strategy)	4	2	3	2	3

will be recommended to the group (Cantador and Castells 2012). In this example, *recipe₁* is recommended to the group because its group rating value is highest (i.e., 4).

Also for the purpose of supporting a group decision making process in a family, Elahi et al. (2014) proposed a novel interactive environment for groups in planning their meals through a conversational process based on critiquing (Chen and Pu 2012). The system consists of two components. The first one is a tagging and critiquing-based user interface. The second one is a utility function that takes into account diet compliance and healthiness of the users. The utility for each meal is calculated on the basis of considering *meal time*, *user rating*, *diet plan*, and *health situation* of each group member. After that, the utility of each meal for the whole group is quantified by aggregating the individual utility scores of all group members. Based on the utility of each meal for the whole group, the system delivers a meal recommendation list for the group. Each group has a *group leader* (also called *the cook*), and *participants* who will attend the group meal. Sometimes, the cook must not select the recipe with the highest utility score. (S)he can accept or refuse recipe(s) for some reasons (e.g., the unavailability of ingredients or insufficient cooking-skills). The participants are allowed to criticize the meal which was chosen by the cook. This critiquing process will be repeated until all members are satisfied.

Until now, to the best of our knowledge, there has been only few research on food recommender systems for groups. In the mentioned study (Elahi et al. 2014), although proposing a new interactive mechanism for group in the food domain, it exposes a lot of issues to be tackled in terms of group decision making, such as *bundle recommendation*, *fast consensus in a group*, *time of preference visibility*, etc. Figure 4 illustrates the user interfaces of the CHOICLA⁵ group decision support environment (Stettinger 2014), which can be applied as a potential solution for supporting group decision making process in the food domain. CHOICLA can support a group of friends to choose a menu for a Christmas party in an asynchronous fashion. That means all group members can join the decision making process without being on-line together at the same time. In this scenario, one member creates a decision (e.g., *Christmas party*) and enters some menus into the decision. Each menu is described by *name*, *photo*, and *description*. While joining in a decision, each group member is able to invite other members to participate in this decision. Invited members give their preferences by rating proposed menus (e.g., using thumbs up and thumbs down) and can discuss with each other by using the “*comment*” functionality. Rating values from group members will be aggregated into group preferences using some group decision heuristics (e.g., *average*, *least misery*, *most pleasure*, etc.) Masthoff (2011) to propose a menu for the

⁵www.choicla.com

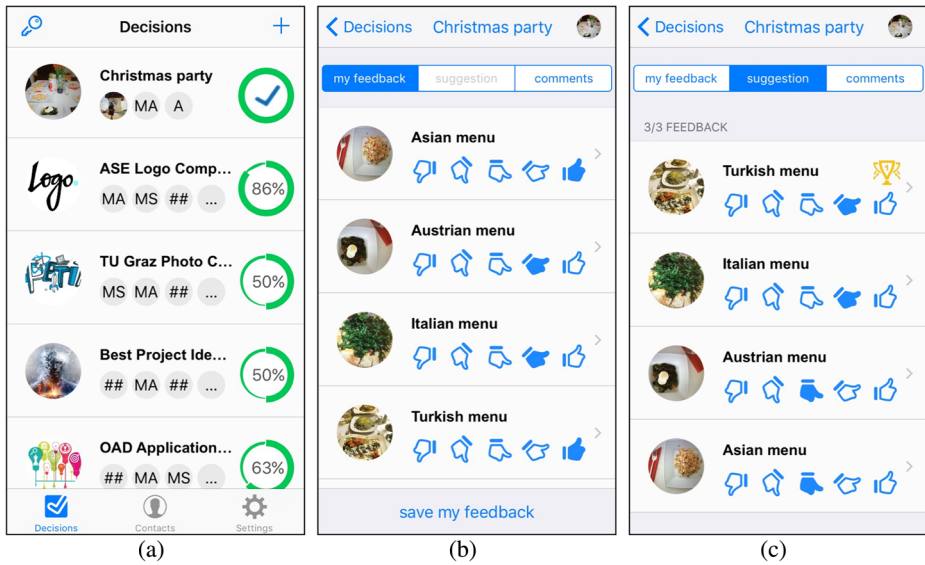


Fig. 4 Screenshots of the CHOICLA group decision support environment (iOS version). Figure 4a shows a list of different group decisions created by users. Users can rate alternatives by using the user interface shown in Fig. 4b. The suggestion for the whole group is shown in the “Suggestion” tab (Fig. 4c). The alternative enclosed with the *medal icon* is the suggested alternative for the whole group. For instance, “Turkish menu” is the menu chosen by CHOICLA for recommending to the whole group

whole group. To avoid the anchoring effects bias (Felfernig 2014), the group suggestion is solely shown to a group member after he/she saved the ratings for menus. Having said that, CHOICLA is the potential application for group decision processes in the food domain. However, the future version of CHOICLA should integrate a complete group decision process for the food domain, which takes into account additional information of all group members (such as *health situations, allergies, nutritional consumption, cooking skills, the availability of ingredients, etc.*), in order to recommend healthy food to the whole group.

4 Research challenges

Existing research on food recommender systems plays a crucial role in supporting users to choose a diet which suits interests and health conditions. These studies exploit the information regarding user profiles and recipes in order to generate food recommendations. It has been recognized that the quality of recommendations is strongly influenced by the adequacy and accuracy of user information as well as nutritional information of food. However, recent studies have not provided detailed discussions on this issue. In addition, although some papers (e.g., Ueta et al. 2011, Aberg 2006) propose food recommendations to tackle health problems, suggestions regarding changing eating behaviors, which are the premise to maintain a healthy lifestyle, are still missing. Explanations could help users more trust in recommendations and encourage them to follow good eating habits, however the inclusion of explanations into food recommender systems has not received the interests from researchers. Besides, research on food recommender systems mainly focuses on single-user scenarios rather than on group scenarios. Until now, research on group recommender

systems in the healthy food domain is very limited. Berkovsky and Freyne (2010) is one of the studies which proposes some aggregation strategies to generate food recommendations for groups of users. However, there still exist some open issues which should be taken into account within the scope of future work, such as achieving fast consensus within the group or fostering the fairness among group members. In this section, we will discuss research challenges in food recommender systems and propose some potential solutions. A summary of open issues is presented in Table 7.

4.1 Challenges regarding user information

- *The uncertainty of nutritional information from users:* In order to make recommendations, the system needs to collect *nutritional needs, ratings for food items/recipes and information of previous meals* from users (Mika 2011). Most of the information is only provided through continuous interactions with users. However, in reality, recording nutritional intake from users can not avoid faults because users usually forget or give wrong information about the foods they have consumed (Mika 2011). Although some systems were proposed to tackle with these problems, for instance, FOODLOG (Aizawa et al. 2010), they are not able to give the accurate information about the consumed meals, even though they can estimate the nutritional balance among different kinds of food in a meal.

Table 7 A summary of research challenges in food recommender systems and proposed solutions

Research challenges	Proposed solutions
Collecting user information	<ul style="list-style-type: none"> – Taking advantage of information about users' previous meals (Van Pinxteren et al. 2011). – Implicitly collecting user information so that they don't have to invest too much time and effort (Freyne and Berkovsky 2010).
Gathering nutritional information of recipes	<ul style="list-style-type: none"> – The quantity of gathered recipes should be representative enough to vary the recommendations.
Recommendation algorithms	<ul style="list-style-type: none"> – Improving the quality of recommendations by integrating constraints (e.g., health situations, nutrition needs, the availability of ingredients) into the recommendation process.
Explaining recommendations	<ul style="list-style-type: none"> – Providing explanations which increase the trustworthiness of decision outcomes and persuade users to accept food recommendations (Elahi et al. 2014).
Changing eating behaviors	<ul style="list-style-type: none"> – Employing health psychology theory in food recommender systems to encourage users to comply healthy eating behaviors (Snooks 2009). – Proposing potential dietary changes on the basis of exploiting the ideal nutrients from reliable resources (e.g., USDA, DACH).
Generating bundle recommendations	<ul style="list-style-type: none"> – Expressing acceptable trade-offs among group members by employing negotiation and argumentation mechanisms (Felfernig et al. 2014b).
Achieving fast consensus in group decision making	<ul style="list-style-type: none"> – Enriching user interfaces supporting basic negotiation mechanisms among group members (Thuy Ngoc Nguyen 2017).

- *Collecting user rating data*: Food recommender systems need information about users' preferences to recommend similar food items ((Van Pinxteren et al. 2011; Mika 2011)). This information can be gathered by asking users to rate foods/recipes. However, it is not convenient if the system asks users to rate too many items. Hence, a challenge entailed is “*How to collect enough users' ratings while saving their effort?*” Freyne and Berkovsky (2010). Besides, similar to keeping reporting food consumption (as mentioned above), persuading users to keep rating dishes also becomes another challenge for food recommender systems (Mika 2011).

4.2 Challenges regarding recommendation algorithms

As mentioned in Mika (2011), in order to calculate nutritional recommendations for users, any algorithm needs the following information:

- *User information* (e.g., likes, dislikes, food consumption, or nutritional needs): Similar to recommender systems in other domains, food recommender systems also face with the cold-start problem when the system is used in the first time (Mika 2011). This problem can be surmounted by using information about users' previous meals to calculate similarity and then recommend new recipes to users (Van Pinxteren et al. 2011). However, this solution requires many user efforts and abates the desire of system usage.
- *Recipe databases*: Mika (2011) discussed two challenges that need to be solved:

How many recipes the system should have? The quantity of gathered recipes should be large enough to accommodate the preferences of many users and vary the recommended recipes while still minimize the time for making recommendations. This is a tricky problem when the system tries to balance between the variety of recommendations and system response time. Hoxmeier et al. (Hoxmeier and Manager 2000) point out that long response times triggers user dissatisfaction which further decreases continuous use of the system.

How to gather accurate nutritional information of recipes? It is observed that with the same food item, if we use different ways to cook it then we will get different nutritional values from it Mika (2011). Therefore, it is very difficult to ensure that whether gathered nutritional tables for food items are accurate because when comparing different nutritional value table of foods, sometimes it returns varying values for the same food items (Mika 2011). For instance, the nutritional value of *celery* in “a salad recipe” is different from the nutritional value of itself “in a fried recipe”, since cooking with high temperature make celery lose a big amount of *essential oil*. It means that the amount of essential oil of celery in the “fried recipe” could be lower than in the “salad recipe”.

- *A set of constraints or rules*: Considering more constraints and rules in the recommendation process will improve the quality of recommendations (Mika 2011). For instance, with a user who has heart disease, the system should recommend menus with less fat and salt. Moreover, it is very necessary to detect the conflicts among the constraints or rules which prevent the recommendation algorithms from finding a solution. However, with the large database (e.g., thousands of foods/recipes), checking constraints/rules in the database brings negative effects for system performance (Mika 2011). In addition, food recommender systems should take into account constraints with regard to the availability of ingredients in the households for the purpose of helping users to save money and prevent the food waste behavior. The challenge here is how to propose food which meets health situations and nutritional needs of users, as well as taking

advantages of the ingredients that are currently in the fridge. In this scenario, recommender systems seems to require many efforts from users because users have to report the consumption of all ingredients regularly and this can prevent users from using the system permanently.

4.3 Challenges regarding changing eating behavior of users

Nowadays, many people are suffering health problems because of inappropriate eating habits (Snooks 2009). For instance, some people eat too much food compared to their physical activity level and gradually become obese. Whereas, others (e.g., the elderly, the dieters) restrict extremely nutrition intake and this leads to malnutrition. Therefore, one of the main functions of food recommender systems is understanding users' eating behaviors and persuading them to change eating behaviors in positive ways. However, this is a big challenge for food recommender systems because eating is a lifelong behavior which is influenced by many factors, especially psychological factors. Hence, food recommender systems should integrate health psychology theory in order to stimulate users to comply healthy eating behaviors. The first approach can be used by applying one simple change at a specific time until the user behavior becomes habitual (Snooks 2009). Another approach can be applied for food recommender systems by comparing to the ideal nutrient. Users can find the structure of ideal diet according to the age and physical activity level from reliable resources (e.g., USDA, DACH) and then compare what food they ate to what is recommended (Snooks 2009). The comparison approach is also proposed in Mankoff et al. (2002) in order to provide users potential dietary changes.

4.4 Challenges regarding explanations

Explanations play an important role in recommender systems since they increase the trust of users in decision outcomes (Tintarev and Masthoff 2007). In the healthy food domain, explanations are even more necessary since they not only increase the trust in recommendations but also stimulate users to consume healthy foods or change their eating behaviors. For this purpose, it makes sense that explanations of food recommender systems clarify how a decision outcome is created (Elahi et al. 2014). Besides, a detailed description of food items (e.g., nutritional value table for a recipe) needs to be included in a way that emphasizes the healthiness of a specific food for users.

4.5 Challenges regarding group decision making

As mentioned in previous sections of the paper, recommending recipes/food items usually involves groups rather than individual users. However, there is a low amount of research on food recommender systems for groups. Therefore, it is still an open topic which needs to be analyzed in future research.

- *Bundle recommendations*: Group recommender systems usually attach the requirements/preferences of different users into group recommendation. This is the crucial idea discussed in many related studies (Masthoff 2011; O'Connor et al. 2001; Berkovsky and Freyne 2010). In the food domain, the aggregation process raises more challenges when users want to get recommendations for a complete meal with the combination of many recipes/food or a food schedule for more than one day (e.g., foods for next week). This issue is known as *bundle recommendation* which is a new research branch

of recommender systems. The idea here is recommending a sequence of items instead of separated ones. In the healthy food domain, recommending a complete meal is even more complicated because the system has to consider not only preferences of group members but also other aspects, e.g., *the variety of meals, weather and season* (Van Pinxteren et al. 2011), *the healthiness of recipes, health problems or daily nutrition needs of group members*, etc. On the other hand, the recommendation of bundles has to assure the fairness among users within the group. It means that negotiation and argumentation mechanisms have to be developed in order to support group members to express acceptable trade-offs (Felfernig et al. 2014b). For instance, in a meal plan for a week, the preferences of users who were discriminated in previous meals should have a higher emphasis in the upcoming meals.

- *Achieving fast consensus in groups*: In group recommender systems, although different aggregation approaches have been applied to generate group recommendations, such processes do not ensure that the recommended items reflect a high agreement level among group members (Castro et al. 2015). In this context, achieving consensus helps to bring individual preferences closer to each other before delivering group recommendations. However, further issues need to be considered in order to accelerate the achievement of consensus in groups. One of the promising solutions is enriching user interfaces which support basic negotiation mechanisms among users. User interfaces are designed such that all members can share their preferences within the group (Thuy Ngoc Nguyen 2017). Knowing the preferences of each other helps the group to reach a consensus quickly. An example thereof is the following: user *A* prefers cheese, whereas user *B* is interested in beef. There is a probability to achieve a consensus between user *A* and user *B* is that user *A* will eat recipes with beef as long as they include cheese. How to represent the current decision situation is considered as an issue of future work.

5 Conclusion

In this paper, we provide an overview of recommender systems in the healthy food domain on the basis of discussing four different types of food recommender systems. The first three types present some existing studies in the healthy food domain, which mainly focus on tailoring recommendations to individuals, by considering the preferences and/or nutritional needs of users. Meanwhile, recent studies presented in the fourth type target at consulting healthy food items in group scenarios. Popular recommendation approaches (e.g., Collaborative Filtering recommendation, Content-based recommendation, Constraint-based recommendation) are used in many food recommender systems. Besides, hybrid approaches are also employed in order to improve the recommender's performance. Although being considered in different contexts, in general all food recommender systems play a vital role in providing food items meeting preferences and adequate nutritional needs of users as well as persuading them to comply positive eating behaviors. Some challenges regarding user information, recommendation algorithms, changing eating behaviors, explanations provision, and group decision making are discussed as issues for further work.

Acknowledgements Open access funding provided by Graz University of Technology. This work presented in this paper has been developed within the scope of the WE-WANT project (Enabling Technologies for Group-based Configuration) which is funded by the Austrian Research Promotion Agency (850702).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Aberg, J. (2006). Dealing With malnutrition: A meal planning system for elderly.
- Adomavicius, G., Bockstedt, J., Curley, S., & Zhang, J. (2011). *Recommender systems, consumer preferences, and anchoring effects*, 811, 35–42.
- Aizawa, K., de Silva, G.C., Ogawa, M., & Sato, Y. (2010). Food log by snapping and processing images. In *2010 16th international conference on virtual systems and multimedia, IEEE* (pp. 71–74).
- Ardissono, L., Goy, A., Petrone, G., Segnan, M., & Torasso, P. (2003). Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices. *Applied Artificial Intelligence*, 17(8), 687–714.
- Asanov, D. (2011). *Algorithms and methods in recommender systems Berlin Institute of Technology*. Germany: Berlin.
- Balabanović, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66–72.
- Berkovsky, S., & Freyne, J. (2010). Group-based recipe recommendations: Analysis of data aggregation strategies. In *Proceedings of the fourth ACM conference on recommender systems, RecSys '10* (pp. 111–118).
- Bokde, D.K., Girase, S., & Mukhopadhyay, D. (2015). Role of matrix factorization model in collaborative filtering algorithm: A survey. CoRR abs/1503.07475.
- Bridge, D., Göker, M.H., McGinty, L., & Smyth, B. (2005). Case-based recommender systems. *Knowl Eng Review*, 20(3), 315–320.
- Burke, R. (2000). Knowledge-based recommender systems. In *Encyclopedia of library and information systems, vol 69, Marcel Dekker* (pp. 180–200).
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 331–370.
- Burke, R., Felfernig, A., & Göker, M.H. (2011). Recommender systems: an overview. *AI Magazine*, 32, 13–18.
- Cantador, I., & Castells, P. (2012). Group recommender systems: New perspectives in the social web. In *Recommender systems for the social Web, intelligent systems reference library, vol 32, Springer Berlin Heidelberg* (pp. 139–157).
- Castro, J., Quesada, F.J., Palomares, I., & Martínez-López, L. (2015). A consensus-driven group recommender system. *International Journal of Intelligent Systems*, 30(8), 887–906.
- Chen, L., & Pu, P. (2012). Critiquing-based recommenders: survey and emerging trends. *User Model User-Adapt Interact*, 22(1-2), 125–150.
- Crossen, A., Budzik, J., & Hammond, K.J. (2002). Flytrap: Intelligent group music recommendation. In *Proceedings of the 7th international conference on intelligent user interfaces, ACM, New York, NY, USA, IUI '02* (pp. 184–185).
- Ekstrand, M.D., Riedl, J.T., & Konstan, J.A. (2011). Collaborative filtering recommender systems. *Found Trends Hum-Comput Interact*, 4(2), 81–173.
- El-Dosuky, M.A., Rashad, M.Z., Hamza, T.T., & El-Bassiouny, A.H. (2012). Food recommendation using ontology and heuristics. In *AMLTA, Springer, communications in computer and information science*, (Vol. 322 pp. 423–429).
- Elahi, M., Ge, M., Ricci, F., Massimo, D., & Berkovsky, S. (2014). Interactive food recommendation for groups. In *RECSYS*, Vol. 1247.
- Elahi, M., Ge, M., Ricci, F., Fernández-Tobías, I., Berkovsky, S., & Massimo, D. (2015). Interaction design in a mobile food recommender system. In *IntRS@recsys, CEUR-WS.org, CEUR workshop proceedings*, (Vol. 1438 pp. 49-52).
- Elsweiler, D., Harvey, M., Ludwig, B., & Said, A. (2015). Bringing the healthy into food recommenders. In Ge, M., & Ricci, F. (Eds.) *DMRS, CEUR-WS.org, CEUR workshop proceedings*, (Vol. 1533 pp. 33-36).
- Felfernig, A. (2014). Biases in decision making. In *Proceedings of the first international workshop on decision making and recommender systems (DMRS2014), Bolzano, Italy, September 18-19, 2014., vol 1278, CEUR Proceedings* (pp. 32–37).

- Felfernig, A., & Burke, R. (2008). Constraint-based recommender systems: Technologies and research issues. In *Proceedings of the 10th international conference on electronic commerce, ACM, New York, NY, USA, ICEC '08* (pp. 3:1–3:10).
- Felfernig, A., Teppan, E., & Gula, B. (2006). Knowledge-based recommender technologies for marketing and sales. *Pattern Recognition and Artificial Intelligence*, 21(2), 1–22.
- Felfernig, A., Friedrich, G., Jannach, D., & Zanker, M. (2011). Recommender systems handbook, Springer US, chapter: Developing Constraint-based Recommenders, 187–215.
- Felfernig, A., Zehentner, C., Ninaus, G., Grabner, H., Maalej, W., Pagano, D., Weninger, L., & Reinfrank, F. (2012). Advances in user modeling, Springer Berlin Heidelberg, chapter: Group Decision Support for Requirements Negotiation, 105–116.
- Felfernig, A., Hotz, L., Bagley, C., & Tiihonen, J. (2014a). *Knowledge-based Configuration: From Research to Business Cases*, 1st edn. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Felfernig, A., Stettinger, M., Ninaus, G., Jeran, M., Reiterer, S., Falkner, A.A., Leitner, G., & Tiihonen, J. (2014b). Towards open configuration. In *Proceedings of the 16th international configuration workshop, Novi Sad, Serbia, September 25-26, 2014* (pp. 89–94).
- Freyne, J., & Berkovsky, S. (2010). Intelligent food planning: personalized recipe recommendation. In *Proceedings of the 15th international conference on Intelligent user interfaces, ACM, New York, NY, USA, IUI '10* (pp. 321–324).
- Freyne, J., Berkovsky, S., & Smith, G. (2011). Recipe recommendation: Accuracy and reasoning. In *19th international conference, UMAP 2011, Girona, Spain, July 11-15, 2011., Springer Berlin Heidelberg* (pp. 99–110).
- Harvey, M., & Elswiler, D. (2015). Automated recommendation of healthy, personalised meal plans. In *Proceedings of the 9th ACM conference on recommender systems, ACM, New York, NY, USA, RecSys '15* (pp. 327–328).
- Hoxmeier, J.A.D.P., & Manager, C.D. (2000). System response time and user satisfaction: An experimental study of browser-based applications. In *Proceedings of the association of information systems americas conference* (pp. 10–13).
- Jameson, A. (2004). More than the sum of its members: Challenges for group recommender systems. In *Proceedings of the working conference on advanced visual interfaces, ACM, New York, NY, USA, AVI '04* (pp. 48–54).
- Jameson, A., & Smyth, B. (2007). The adaptive web. chap Recommendation to Groups, 596–627.
- Knowler, W., Barrett-Connor, E., Fowler, S., Hamman, R., Lachin, J., Walker, E., & Nathan, D. (2002). Reduction in the incidence of type 2 diabetes with lifestyle intervention or metformin. *New England Journal of Medicine*, 346(6), 393–403.
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37.
- Kuo, F.F., Li, C.T., Shan, M.K., & Lee, S.Y. (2012). Intelligent menu planning: Recommending set of recipes by ingredients. In *Proceedings of the ACM multimedia 2012 workshop on multimedia for cooking and eating activities, ACM, New York, NY, USA, CEA '12* (pp. 1–6).
- Lang, K. (1995). Newsweeper: Learning to filter netnews. In Prieditis, A., & Russell, S.J. (Eds.) *Machine learning, proceedings of the twelfth international conference on machine learning, Tahoe City, California, USA, July 9-12, 1995, Morgan Kaufmann* (pp. 331–339).
- Lieberman, H., Van Dyke, N.W., & Vivacqua, A.S. (1999). Let's browse: A collaborative web browsing agent. In *Proceedings of the 4th international conference on intelligent user interfaces, ACM, New York, NY, USA, IUI '99* (pp. 65–68).
- Mankoff, J., Hsieh, G., Hung, H.C., Lee, S., & Nitao, E. (2002). Using low-cost sensing to support nutritional awareness. In *Ubicomp, Springer, lecture notes in computer science*, (Vol. 2498 pp. 371–376).
- Masthoff, J. (2004). Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1), 37–85.
- Masthoff, J. (2011). Group recommender systems: Combining individual models. In *Recommender systems handbook, Springer* (pp. 677–702).
- McCarthy, J.F., & Anagnost, T.D. (1998). Musicfx: An arbiter of group preferences for computer supported collaborative workouts. In *Proceedings of the 1998 ACM conference on computer supported cooperative work, ACM, New York, NY, USA, CSCW '98* (pp. 363–372).
- McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., & Nixon, P. (2006). Group recommender systems: A critiquing based approach. In *Proceedings of the 11th international conference on intelligent user interfaces, ACM, New York, NY, USA, IUI '06* (pp. 267–269).
- Mika, S. (2011). Challenges for nutrition recommender systems. CEUR-WS.org, Workshop Proceedings on Context Aware Intelligent Assistance, 25–33.
- Mooney, R.J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. 195–204.

- O'Connor, M., Cosley, D., Konstan, J.A., & Riedl, J. (2001). Polylens: A recommender system for groups of users. In *Proceedings of the seventh conference on European conference on computer supported cooperative work, Kluwer Academic Publishers, Norwell, MA, USA, ECSCW'01* (pp. 199–218).
- Pazzani, M.J., Muramatsu, J., & Billsus, D. (1996). Syskill and webert: Identifying interesting web sites. In Clancey, W.J., & Weld, D.S. (Eds.) *AAAI/IAAI, Vol. 1, AAAI Press / The MIT Press* (pp. 54–61).
- Ricci, F., Rokach, L., Shapira, B., & Kantor, P.B. (2010). *Recommender Systems Handbook*, 1st edn. New York, NY, USA: Springer-Verlag New York, Inc.
- Robertson, A. (2004). Food and health in europe: a new basis for action. Academic Search Complete, WHO Regional Office for Europe.
- Roza, A.M., & Shizgal, H.M. (1984). The harris benedict equation reevaluated: resting energy requirements and the body cell mass. *The American Journal of Clinical nutrition*, 40(1), 168–182.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web, ACM, New York, NY, USA, WWW '01* (pp. 285–295).
- Smith, R.B., Hixon, R., & Horan, B. (1998). Supporting flexible roles in a shared space. In *Proceedings of the 1998 ACM conference on computer supported cooperative work, ACM, New York, NY, USA, CSCW '98* (pp. 197–206).
- Snooks, M. (2009). Health Psychology: Biological, Psychological, and Sociocultural Perspectives, Jones & Bartlett Learning chapter 5: Applications of Health Psychology to Eating Behaviors: Improving health through nutritional changes.
- Stettinger, M. (2014). Choicla: Towards domain-independent decision support for groups of users. In *Proceedings of the 8th ACM conference on recommender systems, ACM, New York, NY, USA, RecSys '14* (pp. 425–428).
- Stettinger, M., Felfernig, A., Leitner, G., Reiterer, S., & Jeran, M. (2015). Counteracting serial position effects in the choicla group decision support environment. In *Proceedings of the 20th international conference on intelligent user interfaces. IUI 2015 (ACM - San Francisco)* (pp. 148–157).
- Svensson, M., Laaksojahti, J., Höök, K., & Waern, A. (2000). A recipe based on-line food store. In *Proceedings of the 5th international conference on intelligent user interfaces, ACM, New York, NY, USA, IUI '00* (pp. 260–263).
- Thuy Ngoc Nguyen, F.R. (2017). A chat-based group recommender system for tourism. *Information and Communication Technologies*, 17–30.
- Tintarev, N., & Masthoff, J. (2007). A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd international conference on data engineering workshop, IEEE Computer Society, Washington, DC, USA, ICDEW '07* (pp. 801–810).
- Ueta, T., Iwakami, M., & Ito, T. (2011). A recipe recommendation system based on automatic nutrition information extraction. In *Proceedings of the 5th international conference on knowledge science, engineering and management, Springer-Verlag, Berlin, Heidelberg, KSEM'11* (pp. 79–90).
- Van Pinxteren, Y., Geleijnse, G., & Kamsteeg, P. (2011). Deriving a recipe similarity measure for recommending healthful meals. In *Proceedings of the 16th international conference on intelligent user interfaces, ACM, New York, NY, USA, IUI '11* (pp. 105–114).
- Yu, Z., Zhou, X., Hao, Y., & Gu, J. (2006). Tv program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction*, 16(1), 63–82.