Yum-Me: A Personalized Nutrient-Based Meal Recommender System

LONGQI YANG, Cornell Tech, Cornell University
CHENG-KANG HSIEH, University of California, Los Angeles
HONGJIAN YANG and JOHN P. POLLAK, Cornell Tech, Cornell University
NICOLA DELL, Cornell Tech, The Jacobs Institute, Cornell University
SERGE BELONGIE, Cornell Tech, Cornell University
CURTIS COLE, Weill Cornell Medical College, Cornell University
DEBORAH ESTRIN, Cornell Tech, Cornell University

Nutrient-based meal recommendations have the potential to help individuals prevent or manage conditions such as diabetes and obesity. However, learning people's food preferences and making recommendations that simultaneously appeal to their palate and satisfy nutritional expectations are challenging. Existing approaches either only learn high-level preferences or require a prolonged learning period. We propose *Yum-me*, a personalized nutrient-based meal recommender system designed to meet individuals' nutritional expectations, dietary restrictions, and fine-grained food preferences. Yum-me enables a simple and accurate food preference profiling procedure via a visual quiz-based user interface and projects the learned profile into the domain of nutritionally appropriate food options to find ones that will appeal to the user. We present the design and implementation of Yum-me and further describe and evaluate two innovative contributions. The first contriution is an open source state-of-the-art food image analysis model, named *FoodDist*. We demonstrate FoodDist's superior performance through careful benchmarking and discuss its applicability across a wide array of dietary applications. The second contribution is a novel online learning framework that learns food preference from itemwise and pairwise image comparisons. We evaluate the framework in a field study of 227 anonymous users and demonstrate that it outperforms other baselines by a significant margin. We further conducted an end-to-end validation of the feasibility and effectiveness of Yum-me through a 60-person user study, in which Yum-me improves the recommendation acceptance rate by 42.63%.

CCS Concepts: • **Information systems → Information retrieval**; **Users and interactive retrieval**; **Personalization;**

Additional Key Words and Phrases: Nutrient-based meal recommendation, personalization, visual interface, food preferences, online learning

This work is funded through Awards from NSF (#1344587, #1343058) and NIH (#1U54EB020404); as well as gift funding from AOL, RWJF, UnitedHealth Group, Google, and Adobe.

Authors' addresses: L. Yang, 2 West Loop Road, NY, NY 10044, Department of Computer Science, Cornell Tech, Cornell University; email: ylongqi@cs.cornell.edu; C.-K. Hsieh, 4732 Boelter Hall, Los Angeles, CA 90095, Department of Computer Science, UCLA; email: changun@cs.ucla.edu; H. Yang, 2 West Loop Road, NY, NY 10044, Cornell Tech, Cornell University; email: hy457@cornell.edu; J. P. Pollak, 2 West Loop Road, NY, NY 10044, Cornell Tech, Cornell University; email: jpp9@cornell.edu; N. Dell, 2 West Loop Road, NY, NY 10044, Department of Information Science, Cornell Tech, The Jacobs Institute, Cornell University; email: nixdell@cornell.edu; S. Belongie, 2 West Loop Road, NY, NY 10044, Department of Computer Science, Cornell Tech, Cornell University; email: sjb344@cornell.edu; C. Cole, 505 East 70th Street, Helmsley Tower, 4th Floor, NY, NY 10021, Weill Cornell Medical College, Cornell University; email: ccole@med.cornell.edu; D. Estrin, 2 West Loop Road, NY, NY 10044, Department of Computer Science, Cornell Tech, Cornell University; email: destrin@cornell.edu.

This work is licensed under a Creative Commons Attribution International 4.0 License.

2017 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 1046-8188/2017/07-ART7 $15.00
DOI: http://dx.doi.org/10.1145/3072614

ACM Reference Format:
Longqi Yang, Cheng-Kang Hsieh, Hongjian Yang, John P. Pollak, Nicola Dell, Serge Belongie, Curtis Cole, and Deborah Estrin. 2017. Yum-me: A personalized nutrient-based meal recommender system. ACM Trans. Inf. Syst. 36, 1, Article 7 (July 2017), 31 pages.
DOI: http://dx.doi.org/10.1145/3072614

1. INTRODUCTION

Healthy eating plays a critical role in our daily well-being and is indispensable in preventing and managing conditions such as diabetes, high blood pressure, cancer, mental illnesses, asthma, and so on [5, 41]. In particular, for children and young people, the adoption of healthy dietary habits has been shown to be beneficial to early cognitive development [46]. Many applications designed to promote healthy behaviors have been proposed and studied [10, 12, 28, 29]. Among those applications, the studies and products that target healthy meal recommendations have attracted much attention [40, 56]. Fundamentally, the goal of these systems is to suggest food alternatives that cater to individuals' health goals and help users develop healthy eating behavior by following the recommendations [67]. Akin to most recommender systems, learning users' preferences is a necessary step in recommending healthy meals that users are more likely to find desirable [67]. However, the current food preference elicitation approaches, including (1) on-boarding surveys and (2) food journaling, still suffer from major limitations, as discussed below.

- **Preferences elicited by surveys are coarse-grained.** A typical on-boarding survey asks a number of multi-choice questions about general food preferences. For example, PlateJoy [40], a daily meal planner app, elicits preferences for healthy goals and dietary restrictions with the following questions:
  (1) How do you prefer to eat? No restrictions, dairy free, gluten free, kid friendly, pescatarian, paleo, vegetarian...
  (2) Are there any ingredients you prefer to avoid? avocado, eggplant, eggs, seafood, shellfish, lamb, peanuts, tofu....
  While the answers to these questions can and should be used to create a rough dietary plan and avoid clearly unacceptable choices, they do not generate meal recommendations that cater to each person's fine-grained food preferences, and this may contribute to their lower-than-desired recommendation-acceptance rates, as suggested by our user testing results.
- **Food journaling approach suffers from cold-start problem and is hard to maintain.** For example, Nutrino [38], a personal meal recommender, asks users to log their daily food consumption and learn users' fine-grained food preferences. As is typical of systems relying on user-generated data, food journaling suffers from the *cold-start* problem, where recommendations cannot be made or are subject to low accuracy when the user has not yet generated a sufficient amount of data. For example, a previous study showed that an active food-journaling user makes about 3.5 entries per day [13]. It would take a non-trivial amount of time for the system to acquire sufficient data to make recommendations, and the collected samples may be subject to sampling biases as well [13, 32]. Moreover, the photo food journaling of all meals is a habit difficult to adopt and maintain and therefore is not a generally applicable solution to generate complete food inventories [13].

To tackle these limitations, we develop *Yum-me*, a meal recommender that learns *fine-grained food preferences without relying on the user's dietary history*. We leverage people's apparent desire to engage with food photos[1] to create a more user-friendly medium for asking visually based diet-related questions. The recommender learns users' fine-grained food preferences through a simple quiz-based visual interface [59] and then attempts to generate meal recommendations that cater to the user's health goals, food restrictions, as well as personal appetite for food. It can be used by people who have food restrictions, such as vegetarian, vegan, kosher, or halal. Particularly, we focus on the health goals in the form of nutritional expectations, for example adjusting calories, protein, and fat intake. The mapping from health goals to nutritional expectations can be accomplished by professional nutritionists or personal coaches and is out of the scope of this article. We leave it as future work. In designing the visual interface [59], we propose a novel online learning framework that is suitable for learning users' potential preferences for a large number of food items while requiring only a modest number of interactions. Our online learning approach balances exploitation-exploration and takes advantage of food similarities through preference-propagation among locally connected graphs. To the best of our knowledge, this is the first interface and algorithm that learns users' food preferences through real-time interactions without requiring specific diet history information.

For such an online learning algorithm to work, one of the most critical components is a robust food image analysis model. Towards that end, as an additional contribution of this work, we present a novel, unified food image analysis model, called *FoodDist*. Based on deep convolutional networks and multi-task learning [6, 33], FoodDist is the best-of-its-kind Euclidean distance embedding for food images, in which similar food items have smaller distances while dissimilar food items have larger distances. FoodDist allows the recommender to learn users' fine-grained food preferences accurately via similarity assessments on food images. Besides preference learning, FoodDist can be applied to other food-image-related tasks, such as food image detection, classification, retrieval, and clustering. We benchmark FoodDist with the Food-101 dataset [6], the largest dataset for food images. The results suggest the superior performance of FoodDist over prior approaches [6, 35, 59]. FoodDist will be made available on Github on publication.

We evaluate our online learning framework in a field study of 227 anonymous users and we show that it is able to predict the food items that a user likes or dislikes with high accuracy. Furthermore, we evaluate the desirability of Yum-me recommendations end to end through a 60-person user study, where each user rates the meal recommendations made by Yum-me relative to those made using a traditional survey-based approach. The study results show that, compared to the traditional survey based recommender, our system significantly improves the acceptance rate of the recommended healthy meals by 42.63%. We see Yum-me as a complement to the existing food preference elicitation approaches that further filters the food items selected by a traditional onboarding survey based on users' fine-grained taste for food and allows a system to serve tailored recommendations on the first use of the system. We discuss some potential use cases in Section 7.

The rest of the article is organized as follows. After discussing related work in Section 2, we introduce the structure of *Yum-me* and our backend database in Section 3. In Section 4, we describe the algorithmic details of the proposed online learning algorithm, followed by the architecture of *FoodDist* model in Section 5. The evaluation results of each component, as well as the recommender are presented in Section 6.

[1] Collecting, sharing, and appreciating high-quality, delicious-looking food images is growing in popularity in our everyday lives. For example, food photos are immensely popular on Instagram (#*food* has over 177M posts and #*foodporn* has over 91M posts at the time of writing).

Finally, we discuss the limitations, potential impact, and real-world applications in Section 7 and conclude in Section 8.

2. RELATED WORK

Our work benefits from, and is relevant to, multiple research threads: (1) healthy meal recommender system, (2) cold-start problem and preference elicitation, (3) pairwise algorithms for recommendation, and (4) food image analysis, which will be surveyed in detail next.

2.1. Healthy Meal Recommender System

Traditional food and recipe recommender systems learn users' dietary preferences from their online activities, including ratings [17, 19, 20, 23], past recipe choices [21, 51], and browsing history [38, 54, 56]. For example, the authors of Reference [51] build a social navigation system that recommends recipes based on the previous choices made by the user; the authors of Reference [56] propose to learn a recipe similarity measure from crowd card-sorting and make recommendations based on the self-reported meals; and the authors of References [17, 23] generate healthy meal plans based on user's ratings towards a set of recipes and the nutritional requirements calculated for the person. In addition, previous recommenders also seek to incorporate users' food consumption histories recorded by the food logging and journaling systems (e.g. taking food images [13] or writing down ingredients and meta-information [56]).

The above systems, while able to learn users' detailed food preference, share a common limitation; that is, they need to wait until a user generates enough data before their recommendations can be effective for this user (i.e., the cold-start problem). Therefore, most commercial applications, for example, Zipongo [68] and Shopwell [47] adopt onboarding surveys to more quickly elicit users' *coarse-grained* food preferences. For instance, Zipongo's questionnaires [68] ask users about their nutrient intake, lifestyle, habits, and food preferences and then make day-to-day and week-to-week healthy meals recommendations; ShopWell's survey [47] is designed to avoid certain food allergens, for example, gluten, fish, corn, or poultry, and find meals that match to particular lifestyles, for example, healthy pregnancy or athletic training.

Yum-me fills a vacuum that the prior approaches were not able to achieve, namely a rapid elicitation of users' *fine-grained* food preferences for immediate healthy meal recommendations. Based on the online learning framework [59], Yum-me infers users' preferences for each single food item among a large food dataset and projects these preferences for general food items into the domain that meets each individual user's health goals.

2.2. Cold-Start Problem and Preference Elicitation

To alleviate the cold-start problem mentioned above, several models of preference elicitation have been proposed in recent years. The most prevalent method of elicitation is to train decision trees to poll users in a structured fashion [15, 22, 42, 50, 66]. These questions are either generated in advance and remain static [42] or change dynamically based on real-time user feedback [15, 22, 50, 66]. Also, another previous work explores the possibility of eliciting item ratings directly from the user [11, 63]. This process can either be carried at item level [63] or within category (e.g., movies) [11].

The preference elicitation methods we mentioned above largely focus on the domain of movie recommendations [11, 42, 50, 63] and visual commerce [15] (e.g., cars, cameras) where items can be categorized based on readily available metadata. When it comes to real dishes, however, categorical data (e.g., cuisines) and other associated information (e.g., cooking time) possess a much weaker connection to a user's food preferences.

Therefore, in this work, we leverage the visual representation of each meal to better capture the process through which people make diet decisions.

2.3. Pairwise Algorithms for Recommendation

Pairwise approaches [25, 39, 44, 45, 57, 58, 60] are widely studied in recommender system literature. For example, Bayesian Personalized Ranking (BPR) [44, 45] and Weighted Approximate-Rank Pairwise (WARP) loss [57], which learn users' and items' representations from user-item pairs, are two representative and popular approaches in this category. Such algorithms have successfully powered many state-of-the-art systems [25, 58]. In terms of the cold-start scenario, the authors of Reference [39] developed a pairwise method to leverage users' demographic information in recommending new items.

Compared to previous methods, our problem setting fundamentally differs in the sense that *Yum-me* elicits preferences in an active manner where the input is incremental and contingent on the previous decisions made by the algorithm, while prior work focuses on the static circumstances where the training data is available up-front, and there is no need for the system to actively interact with the user.

2.4. Food Image Analysis

The tasks of analyzing food images are very important in many ubiquitous dietary applications that actively or passively collect food images from mobile [13] and wearable [1, 36, 52] devices. The estimation of food intake and its nutritional information is helpful to our health [37] as it provides detailed records of our dietary history. Previous work mainly conducted the analysis by leveraging the crowd [37, 53] and computer vision algorithms [6, 35].

Noronha et al. [37] crowdsourced nutritional analysis of food images by leveraging the wisdom of untrained crowds. The study demonstrated the possibility of estimating a meal's calories, fat, carbohydrates, and protein by aggregating the opinions from a large number of people; the authors of Reference [53] elicit the crowd to rank the healthiness of several food items and validate the results against the ground truth provided by trained observers. Although this approach has been justified to be accurate, it inherently requires human resources that restrict it from scaling to large number of users and providing real-time feedback.

To overcome the limitations of crowds and automate the analysis process, numerous articles discuss algorithms for food image analysis, including classification [4, 6, 30, 35], retrieval [31], and nutrient estimation [9, 24, 35, 49]. Most of the previous work [6] leveraged hand-crafted image features. However, traditional approaches were only demonstrated in special contexts, such as in a specific restaurant [4] or for particular types of cuisine [30], and the performance of the models might degrade when they are applied to food images in the wild.

In this article, we designed FoodDist using deep convolutional neural network based multitask learning [8], which has been shown to be successful in improving model generalization power and performance in several applications [14, 64]. The main challenge of multitask learning is to design appropriate network structures and sharing mechanisms across tasks. With our proposed network structure, we show that FoodDist achieves superior performance when applied to the largest available real-world food image dataset [6] and when compared to prior approaches.

3. YUM-ME: PERSONALIZED NUTRIENT-BASED MEAL RECOMMENDATIONS

Our personalized nutrient-based meal recommendation system, Yum-me, operates over a given inventory of food items and suggests the items that will appeal to the users' palate and meet their nutritional expectations and dietery restrictions. A high-level

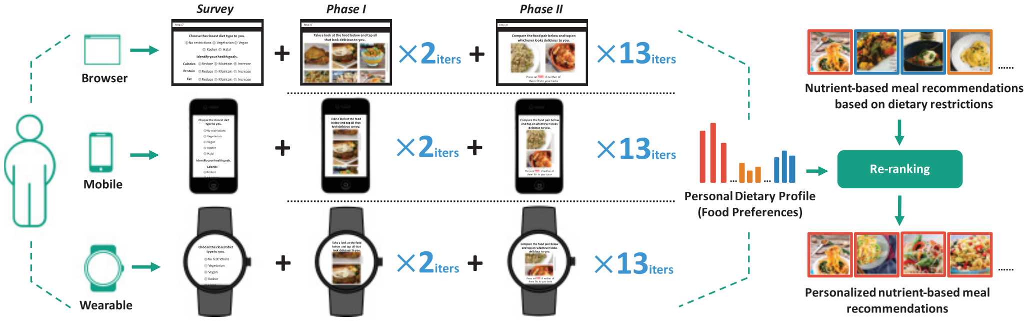


Fig. 1. Overview of Yum-me. This figure shows three sample scenarios in which Yum-me can be used: desktop browser, mobile, and smart watch. The fine-grained dietary profile is used to re-rank and personalize meal recommendations.

overview of Yum-me's recommendation process is shown in Figure 1 and briefly described as follows:

- *Step 1*: Users answer a simple survey to specify their dietary restrictions and nutritional expectations. This information is used by Yum-me to filter food items and create an initial set of recommendation candidates.
- *Step 2*: Users then use an adaptive visual interface to express their fine-grained food preferences through simple comparisons of food items. The learned preferences are used to further re-rank the recommendations presented to them.

In the rest of this section, we describe our backend large-scale food database and aforementioned two recommendation steps: (1) a user survey that elicits user's dietary restrictions and nutritional expectations and (2) an adaptive visual interface that elicits users' fine-grained food preferences.

3.1. Large Scale Food Database

To account for the dietary restrictions in many cultures and religions, or people's personal choices, we prepare a separate food database for each of the following dietary restrictions:

No restrictions, Vegetarian, Vegan, Kosher, Halal[2]

For each diet type, we pulled over 10,000 main dish recipes along with their images and metadata (ingredients, nutrients, tastes, etc.) from the Yummly API [62]. The total number of recipes is around 50,000. To customize food recommendations for people with specific dietary restrictions, for example, vegetarian and vegan, we filter recipes by setting the *allowedDiet* parameter in the search API. For kosher or halal, we explicitly rule out certain ingredients by setting *excludedIngredient* parameter. The lists of excluded ingredients are shown below:

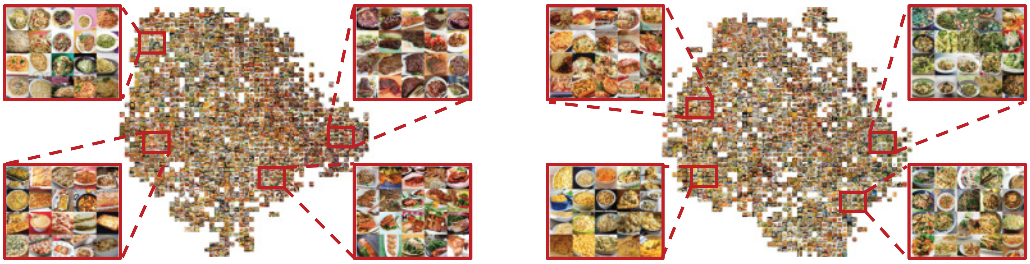
- **Kosher:** pork, rabbit, horse meat, bear, shellfish, shark, eel, octopus, octopuses, moreton bay bugs, frog.
- **Halal:** pork, blood sausage, blood, blood pudding, alcohol, grain alcohol, pure grain alcohol, ethyl alcohol.

One challenge in using a public food image API is that many recipes returned by the API contain non-food images and incomplete nutritional information. Therefore, we further filter the items with the following criteria: The recipe should have (1) nutritional

[2] Our system is not restricted to these five dietary restrictions, and we will extend the system functionalities to other categories in the future.

Table I. Sizes of Databases that Catered to Different Diet Types. Unit: Number of Unique Recipes

Database	Original size	Final size
No restriction	9,405	7,938
Vegetarian	10,000	6,713
Vegan	9,638	6,013
Kosher	10,000	4,825
Halal	10,000	5,002



(a) No restrictions

(b) Vegetarian

Fig. 2. Overview of two sample databases: (a) Database for users without dietary restrictions and (b) Database for vegetarian users.

information of calories, protein, and fat and (2) at least one food image. To automate this process, we build a binary classifier based on a deep convolutional neural network to filter out non-food images. As suggested in Reference [35], we treat the whole training set of Food-101 dataset [6] as one generic *food* category and sampled the same number of images (75,750) from the ImageNet dataset [16] as our *non-food* category. We took the pretrained VGG CNN model [48] and replaced the final 1,000-dimensional softmax with a single logistic node. For the validation, we use the Food-101 testing dataset along with the same number of images sampled from ImageNet (25,250). We trained the binary classifier using the Caffe framework [27], and it reached 98.7% validation accuracy. We applied the criteria to all the datasets, and the final statistics are shown in Table I.

Figure 2 shows the visualizations of the collected datasets. For each of the recipe images, we embed it into a 1,000-dimensional feature space using FoodDist (described later in Section 5) and then project all the images onto a two-dimensional (2D) plane using t-Distributed Stochastic Neighbor Embedding(t-SNE) [55]. For visibility, we further divide the 2D plane into several blocks, from each of which we sample a representative food image residing in that block to present in the figure. Figure 2 demonstrates the large diversity and coverage of the collected datasets. Also, the embedding results clearly demonstrate the effectiveness of FoodDist in grouping similar food items together while pushing dissimilar items away. This is important to the performance of Yum-me, as discussed in Section 6.3.

3.2. User Survey

The user survey is designed to elicit user's high-level dietary restrictions and nutritional expectations. Users can specify their dietary restrictions among the five categories mentioned-above and indicate their nutritional expectations in terms of the desired amount of *calories*, *protein*, and *fat*. We choose these nutrients for their high relevance to many common health goals, such as weight control [18], sports performance [7], and so on. We provide three options for each of these nutrients, including *reduce*, *maintain*, and *increase*. The user's diet type is used to select the appropriate food dataset, and the food items in the dataset are further ranked by their suitability to users' health goals based on the nutritional facts.

To measure the suitability of food items given users' nutritional expectations, we rank the recipes in terms of different nutrients in both *ascending* and *descending* order, such that each recipe is associated with six ranking values, that is, $r_{\text{calories},a}$, $r_{\text{calories},d}$, $r_{\text{protein},a}$, $r_{\text{protein},d}$, $r_{\text{fat},a}$, and $r_{\text{fat},d}$, where a and d stand for *ascending* and *descending*, respectively. The final suitability value for each recipe given the health goal is calculated as follows:

$$u = \sum_{n \in \mathbb{U}} \alpha_{n,a} r_{n,a} + \sum_{n \in \mathbb{U}} \alpha_{n,d} r_{n,d}, \tag{1}$$

where $\mathbb{U} = \{\text{calories, protein, fat}\}$. The indicator coefficient $\alpha_{n,a} = 1 \iff$ nutrient n is rated as *reduce* and $\alpha_{n,d} = 1 \iff$ nutrient n is rated as *increase*. Otherwise $\alpha_{n,a} = 0$ and $\alpha_{n,d} = 0$. If user's goal is to maintain all nutrients, then all recipes are given equal rankings. Eventually, given a user's responses to the survey, we rank the suitability of all the recipes in the corresponding database and select top-M items (around top 10%) as the candidate pool of healthy meals for this user. In our initial prototype, we set $M = 500$.

3.3. Adaptive Visual Interface

Based on the food suitability ranking, a candidate pool of healthy meals is created. However, not all the meals in this candidate pool will suit the user's palate. Therefore, we design an adaptive visual interface to further identify recipes that cater to the user's taste through eliciting their fine-grained food preferences. We propose to learn users' fine-grained food preferences by presenting users with food images and ask them to choose ones that look delicious.

Formally, the food preference learning task can be defined as follows: Given a large *target set* of food items $\mathbb{S}$, we represent user's preferences as a distribution over all the possible food items, that is, $\mathbf{p} = [p_1, \ldots, p_{|\mathbb{S}|}]$, $\sum_i p_i = 1$, where each element p_i denotes the user's favorable scale for item i. Since the number of items, $|\mathbb{S}|$, is usually quite large and intractable to elicit individually from the user,[3] the approach we take is to adaptively choose a specific and much smaller *subset* $\mathbb{V}$ to present to the user and propagate the users' preferences for those items to the rest items based on their visual similarity. Specifically, as Figure 1 shows, the preference elicitation process can be divided into two phases:

Phase I: In each of the first two iterations, we present 10 food images and ask users to tap on all the items that look delicious to them.

Phase II: In each of the subsequent iterations, we present a pair of food images and ask users to either compare the food pair and tap on the one that looks delicious to them or tap on "Yuck" if neither of the items appeal to their taste.

To support the preference elicitation process, we design a novel exploration-exploitation online learning algorithm built on a state-of-the-art food image embedding model, which will be discussed in the Section 4 and Section 5, respectively.

4. ONLINE LEARNING FRAMEWORK

We model the interaction between the user and our backend system at iteration t, ($t \in \mathcal{R}^+$, $t = 1, 2, \ldots, T$) as Figure 3 shows. The symbols that will be used in our algorithms are defined as follows:

[3] The target set is often the whole food database that different applications use. For example, the size of Yummly database can be up to 1 million [62].

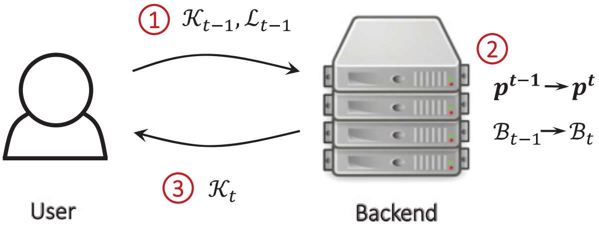

① $\mathcal{K}_{t-1}, \mathcal{L}_{t-1}$
② $p^{t-1} \to p^t$
$\mathcal{B}_{t-1} \to \mathcal{B}_t$
③ $\mathcal{K}_t$
User
Backend

Fig. 3. User-system interaction at iteration t.

- $\mathcal{K}_t$: Set of food items that are presented to user at iteration t ($\mathcal{K}_0 = \emptyset$). $\forall k \in \mathcal{K}_t, k \in \mathbb{S}$;
- $\mathcal{L}_{t-1}$: Set of food items that user *prefer(select)* among $\{k | k \in \mathcal{K}_{t-1}\}$. $\mathcal{L}_{t-1} \subseteq \mathcal{K}_{t-1}$;
- $\boldsymbol{p}^t = [p_1^t, \ldots, p_{|\mathbb{S}|}^t]$: User's preference distribution on all food items at iteration t, where $\|\boldsymbol{p}^t\|_1 = 1$. $\boldsymbol{p}^0$ is initialized as $p_i^0 = \frac{1}{|\mathbb{S}|}$;
- $\mathcal{B}_t$: Set of food images that have been already explored until iteration t ($\mathcal{B}_0 = \emptyset$). $\mathcal{B}_i \subseteq \mathcal{B}_j (i < j)$;
- $\mathcal{F} = \{f(x_1), \ldots, f(x_{|\mathbb{S}|})\}$: Set of feature vectors of food images $x_i (i = 1, \ldots, |\ \mathbb{S}\ |)$ extracted by a feature extractor, denoted by f. We use FoodDist as the feature extractor. More details about FoodDist appear in Section 5.

Based on the workflow depicted in Figure 3, for each iteration t, the backend system updates vector $\boldsymbol{p}^{t-1}$ to $\boldsymbol{p}^t$ and set $\mathcal{B}_{t-1}$ to $\mathcal{B}_t$ based on users' selections $\mathcal{L}_{t-1}$ and previous image set $\mathcal{K}_{t-1}$. After that, it decides the set of images that will be immediately presented to the user (i.e., $\mathcal{K}_t$). Our food preference elicitation framework can be formalized in Algorithm. 1. The core procedures are *update* and *select*, which will be described in the following subsections for more details.

ALGORITHM 1: Food Preference Elicitation Framework

```
Data: $\mathbb{S}, \mathcal{F} = \{f(x_1), \ldots, f(x_{|\mathbb{S}|})\}$
Result: $\boldsymbol{p}^T$
$\mathcal{B}_0 = \emptyset, \mathcal{K}_0 = \emptyset, \mathcal{L}_0 = \emptyset, \boldsymbol{p}^0 = [\frac{1}{|\mathbb{S}|}, \ldots, \frac{1}{|\mathbb{S}|}]$ ;
for $t \leftarrow 1$ to $T$ do
    $[\mathcal{B}_t, \boldsymbol{p}^t] \leftarrow$ update($\mathcal{K}_{t-1}, \mathcal{L}_{t-1}, \mathcal{B}_{t-1}, \boldsymbol{p}^{t-1}$) ;
    $\mathcal{K}_t \leftarrow$ select($t, \mathcal{B}_t, \boldsymbol{p}_t$) ;
    if $t$ equals $T$ then
        return $\boldsymbol{p}^T$
    else
        ShowToUser($\mathcal{K}_t$) ;
        $\mathcal{L}_t \leftarrow$ WaitForSelection() ;
```

4.1. User State Update

Based on user's selections $\mathcal{L}_{t-1}$ and the image set $\mathcal{K}_{t-1}$, the *update* module renews user's state from $\{\mathcal{B}_{t-1}, \boldsymbol{p}^{t-1}\}$ to $\{\mathcal{B}_t, \boldsymbol{p}^t\}$. Our intuition and assumption behind following algorithm design is that people tend to have close preferences for similar food items.

Preference vector $\boldsymbol{p}^t$: Our strategy of updating preference vector $\boldsymbol{p}^t$ is inspired by the Exponentiated Gradient Algorithm in bandit settings (EXP3) [3]. Specifically, at iteration t, each p_i^t in vector $\boldsymbol{p}^t$ is updated by

$$p_i^t \leftarrow p_i^{t-1} \times e^{\frac{\beta u_i^{t-1}}{p_i^{t-1}}}, \tag{2}$$

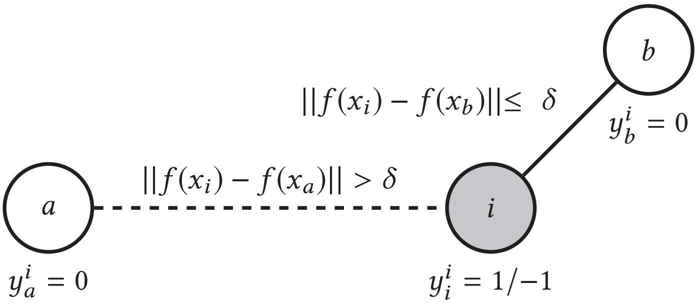

b
$||f(x_i) - f(x_b)|| \leq \delta$
$y_b^i = 0$
$||f(x_i) - f(x_a)|| > \delta$
a
i
$y_a^i = 0$
$y_i^i = 1/-1$

Fig. 4. Locally connected graph with item i.

where β is the exponentiated coefficient that controls update speed and $\boldsymbol{u}^{t-1} = \{u_1^{t-1}, \ldots, u_{|\mathbb{S}|}^{t-1}\}$ is the update vector used to adjust each preference value.

To calculate update vector $\boldsymbol{u}$, we formalize the user's selection process as a data labeling problem [65] where for item $i \in \mathcal{L}_{t-1}$, label $y_i^{t-1} = 1$, and for item $j \in \mathcal{K}_{t-1} \backslash \mathcal{L}_{t-1}$, label $y_j^{t-1} = -1$. Thus, the label vector $\boldsymbol{y}^{t-1} = \{y_1^{t-1}, \ldots, y_{|\mathbb{S}|}^{t-1}\}$ provided by the user is

$$y_i^{t-1} = \begin{cases} 1 & : i \in \mathcal{L}_{t-1} \\ 0 & : i \notin \mathcal{K}_{t-1} \\ -1 & : i \in \mathcal{K}_{t-1} \backslash \mathcal{L}_{t-1} \end{cases}. \tag{3}$$

For update vector $\boldsymbol{u}$, we expect that it is close to label vector $\boldsymbol{y}$ but with smooth propagation of label values to nearby neighbors (for convenience, we omit superscript that denotes current iteration). The update vector $\boldsymbol{u}$ can be regarded as a soften label vector compared with $\boldsymbol{y}$. To make the solution more computationally tractable, for each item i with $y_i \neq 0$, we construct a locally connected undirected graph G^i as Figure 4 shows: $\forall j \in \mathbb{S}$, add an edge (i, j) if $\| f(x_i) - f(x_j) \| \leq \delta$. The labels $\boldsymbol{y}^i$ for vertices s_j in graph G^i are calculated as $y_j^i = 0 (j = 1, \ldots, |\mathbb{S}| \setminus i), y_i^i = y_i$.

For each locally connected graph G^i, we fix u_i^i value as $u_i^i = y_i^i$ and propose the following regularized optimization method to compute other elements $(\forall u_j^i, j \neq i)$ of update vector $\boldsymbol{u}^i$, which is inspired by the traditional label propagation method [65]. Consider the problem of minimizing following objective function $Q(\boldsymbol{u}^i)$:

$$\min_{\boldsymbol{u}^i} \sum_{j=1, j\neq i}^{|\mathbb{S}|} w_{ij} \left(y_i^i - u_j^i\right)^2 + \sum_{j=1, j\neq i}^{|\mathbb{S}|} (1 - w_{ij}) \left(u_j^i - y_j^i\right)^2. \tag{4}$$

In Equation (4), w_{ij} represents the similarity measure between food item s_i and s_j:

$$w_{ij} = \begin{cases} e^{-\frac{1}{2\alpha^2} \| f(x_i) - f(x_j) \|^2} & : \| f(x_i) - f(x_j) \| \leq \delta \\ 0 & : \| f(x_i) - f(x_j) \| > \delta \end{cases}, \tag{5}$$

where $\alpha^2 = \frac{1}{|\mathbb{S}|^2} \sum_{i,j \in \mathbb{S}} \| f(x_i) - f(x_j) \|^2$

The first term of the objective function $Q(\boldsymbol{u}^i)$ is the *smoothness constraint*, as the update value for similar food items should not change too much. The second term is the *fitting constraint*, which makes $\boldsymbol{u}^i$ close to the initial labeling assigned by user (i.e., $\boldsymbol{y}^i$). However, unlike in Reference [65], in our algorithm, the tradeoff between these two constraints is dynamically adjusted by the similarity between item i and j where similar pairs are weighed more with smoothness and dissimilar pairs are forced to be close to initial labeling.

ALGORITHM 2: User State Update Algorithm

```
Function update($\mathcal{K}_{t-1}$, $\mathcal{L}_{t-1}$, $\mathcal{B}_{t-1}$, $\boldsymbol{p}^{t-1}$)
    input: $\mathcal{K}_{t-1}$, $\mathcal{L}_{t-1}$, $\mathcal{B}_{t-1}$, $\boldsymbol{p}^{t-1}$
    output: $\mathcal{B}_t$, $\boldsymbol{p}^t$
    $\boldsymbol{u} = [0, \ldots, 0], \mathcal{B}_t = \mathcal{B}_{t-1}, \boldsymbol{p}^t = \boldsymbol{p}^{t-1}$
    for $i \leftarrow 1$ to $\mid \mathbb{S} \mid$ do
        // preference update
        for $s_j$ in $\mathcal{K}_{t-1}$ do
            $u_i \leftarrow u_i + (-1)^{\mathbf{1}(j \in \mathcal{L}_{t-1})-1} w_{ij}$
        $p_i^t = p_i^{t-1} e^{\frac{\beta u_i}{p_i^{t-1}}}$
        // explored image set update
        if min($\| f(x_i) - f(x_j) \|, \forall j \in \mathcal{K}_{t-1}$) $\leq \delta$ then
            $\mathcal{B}_t \leftarrow \mathcal{B}_t \cup \{i\}$
    // normalize $\boldsymbol{p}^t$ s.t. $\|\boldsymbol{p}^t\|_1 = 1$
    normalize($\boldsymbol{p}^t$)
```

With Equation (4) being defined, we can take the partial derivative of $\boldsymbol{Q}(\boldsymbol{u}^i)$ with respect to different u_j^i as follows:

$$\frac{\partial \boldsymbol{Q}(\boldsymbol{u}^i)}{u_{j,j\neq i}^i} = 2w_{ij}\big(u_j^i - u_i^i\big) + 2(1 - w_{ij})\big(u_j^i - y_j^i\big) = 0. \tag{6}$$

As $u_i^i = y_i^i$, then

$$u_j^i = w_{ij} u_i^i = w_{ij} y_i^i (j = 1, 2, \ldots, \mid \mathbb{S} \mid). \tag{7}$$

After all $\boldsymbol{u}^i$ are calculated, the original update vector $\boldsymbol{u}$ is then the sum of $\boldsymbol{u}^i$, that is, $\boldsymbol{u} = \sum_i \boldsymbol{u}^i$. The pseudo code for the algorithm of updating preference vector is shown in Algorithm.2 for details.

Explored food image set $\mathcal{B}_t$: To balance the *exploitation* and *exploration* in image selection phase, we maintain a set $\mathcal{B}_t$ that keeps track of all similar food items that have already been visited by user and the updating rule for $\mathcal{B}_t$ is as follows:

$$\mathcal{B}_t \leftarrow \mathcal{B}_{t-1} \cup \{i \in \mathbb{S} | min_{j \in \mathcal{K}_{t-1}} \| f(x_i) - f(x_j) \| \leq \delta\}. \tag{8}$$

With the algorithms designed for updating preference vector $\boldsymbol{p}^t$ and explored image set $\mathcal{B}_t$, the overall functionality of procedure *update* is shown in Algorithm 2.

4.2. Images Selection

After updating user state, the *select* module then picks food images that will be presented in the next round. To trade off between exploration and exploitation in our algorithm, we propose different images selection strategies based on current iteration t.

4.2.1. Food Exploration. For each of the first two iterations, we select ten different food images by using *k-means*++ [2] algorithm, which is a seeding method used in *k-means* clustering and can guarantee that selected items are evenly distributed in the feature space. For our use case, the *k-means*++ algorithm is summarized in Algorithm 3.

4.2.2. Food Exploitation-Exploration. Starting from the third iteration, users are asked to make pairwise comparisons between food images. To balance the Exploitation and Exploration, we always select one image from the area with higher preference value based on current $\boldsymbol{p}^t$ and another one from *unexplored* area, that is, $\mathbb{S} \backslash \mathcal{B}_t$. (Both selections

ALGORITHM 3: *k*-means++ Algorithm for Exploration

```
Function k-means-pp($\mathbb{S}$, $n$)
    input: $\mathbb{S}$, $n$
    output: $\mathcal{K}_t$
    $\mathcal{K}_t = \texttt{random}(\mathbb{S})$
    while $|\ \mathcal{K}_t\ | < n$ do
        prob $\leftarrow [0, \dots, 0]_{|\mathbb{S}|}$
        for $i \leftarrow 1$ to $|\ \mathbb{S}\ |$ do
            prob$_i \leftarrow min(\| f(x_i) - f(x_j)\|^2 | \forall j \in \mathcal{K}_t)$
        sample $m \in \mathbb{S}$ with probability $\propto$ prob$_m$
        $\mathcal{K}_t \leftarrow \mathcal{K}_t \cup \{m\}$
```

ALGORITHM 4: Images Selection Algorithm - *select*

```
Function select($t$, $\mathcal{B}_t$, $\boldsymbol{p}^t$)
    input: $t$, $\mathcal{B}_t$, $\boldsymbol{p}^t$
    output: $\mathcal{K}_t$
    $\mathcal{K}_t = \emptyset$
    if $t \leq 2$ then
        $\mathcal{K}_t \leftarrow$ k-means-pp($\mathbb{S}$, 10) // K-means++
    else
        // 99th percentile (top 1%)
        threshold $\leftarrow$ percentile($\boldsymbol{p}^t$, 99)
        topSet $\leftarrow \{s_i \in \mathbb{S} | p_i^t \geq$ threshold$\}$
        $\mathcal{K}_t \leftarrow$ [random(topSet), random($\mathbb{S} \backslash \mathcal{B}^t$)]
```

are *random* in a given subset of food items.) With the above explanations, the image selection method we propose in this application is shown in Algorithm 4.

5. FOODDIST: FOOD IMAGE EMBEDDING

Formally, the goal of FoodDist is to learn a *feature extractor (embedding) f* such that given an image x, $f(x)$ projects it to an N dimensional feature vector for which the Euclidean distance to other such vectors will reflect the similarities between food images, as Figure 5 shows. Formally speaking, if image x_1 is more similar to image x_2 than image x_3, then $\| f(x_1) - f(x_2)\| < \| f(x_1) - f(x_3)\|$.

We build FoodDist based on recent advances in deep Convolutional Neural Networks (CNN), which provide a powerful framework for automatic feature learning. Traditional feature representations for images are mostly hand-crafted and were used with feature descriptors, such as Scale Invariant Feature Transform [34], which aims for invariance to changes in object scale and illumination, thereby improving the generalizability of the trained model. However, in the face of highly diverse image characteristics, the one-size-fits-all feature extractor performs poorly. In contrast, deep learning adapts the features to particular image characteristics and extracts features that are most discriminative in the given task [43].

As we present below, a feature extractor for food images can be learned through classification and metric learning, or through multitask learning, which concurrently performs these two tasks. We demonstrate that the proposed multitask learning approach enjoys the benefits of both classification and metric learning and achieves the best performance.

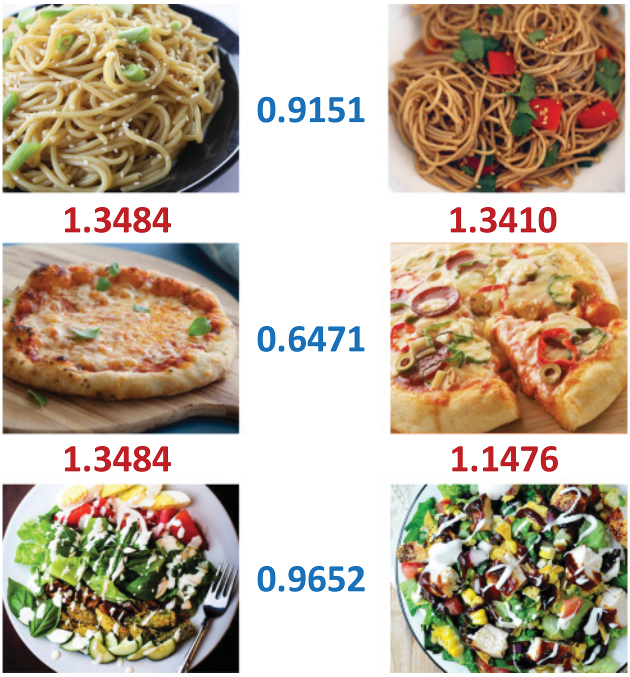


Fig. 5. Euclidean embedding of FoodDist. This figure shows the pairwise Euclidean distances between food images in the embedding. A distance of 0.0 means two food items are identical and a distance of 2.0 represents that the image contents completely differ. For this example, if the threshold is set to 1.0, then all the food images can be correctly classified.

5.1. Learning with Classification

One common way to learn a feature extractor for labeled data is to train a neural network that performs classification (i.e., mapping input to labels) and takes the output of a hidden layer as the feature representations; specifically, using a feedforward deep CNN with n-layers (as the upper half of the Figure 6 shows),

$$F(x) = g_n\left(g_{n-1}\left(\ldots g_i(\ldots g_1(x)\ldots)\right)\right), \tag{9}$$

$$f(x) = g_{n-1}\left(\ldots g_i(\ldots g_1(x)\ldots)\right). \tag{10}$$

Usually, the last few layers will be fully connected layers, and the last layer $g_n(.)$ is roughly equivalent to a linear classifier that is built on the features $f(x)$ [26]. Therefore, $f(x)$ is discriminative in separating instances under different categorical labels, and the Euclidean distances between normalized feature vectors can reflect the similarities between images.

5.2. Metric Learning

Differing from the classification approach, where the feature extractor is a by-product, metric learning proposes to learn the distance embedding directly from the paired inputs of similar and dissimilar examples. Prior work [59] used a *Siamese network* to learn a feature extractor for food images. The structure of a Siamese network resembles that in Figure 6 but without the *Class label*, *Fully connected, 101*, and *Softmax Loss* layers. The inputs to the Siamese network are pairs of food images x_1, x_2. The images pass through CNNs with shared weights, and the output of each network is regarded as the feature representation, that is, $f(x_1)$ and $f(x_2)$, respectively. Our goal is for $f(x_1)$

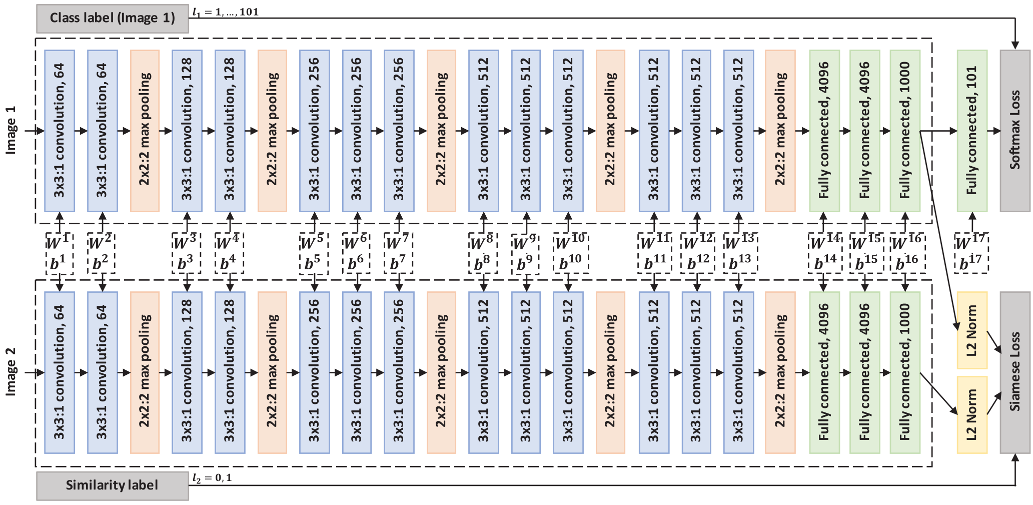


Fig. 6. Multitask learning structure of FoodDist. Different types of layers are denoted by different colors. The format of each type of layer: Convolution layer: [receptive field size:step size ..., #channels]; Pooling layer: [pooling size:step size ...]; Fully connected layer: [..., output dimension].

and $f(x_2)$ to have a small distance value (close to 0) if x_1 and x_2 are similar food items; otherwise, they should have a larger distance value. The value of contrastive loss is then back-propagated to optimize the Siamese network:

$$\mathcal{L}(x_1, x_2, l) = \frac{1}{2} l D^2 + \frac{1}{2}(1 - l) \max(0, m - D)^2, \tag{11}$$

where similarity label $l \in \{0, 1\}$ indicates whether the input pair of food items x_1, x_2 are similar ($l = 1$ for similar, $l = 0$ for dissimilar), $m > 0$ is the margin for dissimilar items, and D is the Euclidean distance between $f(x_1)$ and $f(x_2)$ in embedding space. Minimizing the contrastive loss will pull similar pairs together and push dissimilar pairs farther away (larger than a margin m), and it exactly matches the goal.

The major advantage of metric learning is that the network will be directly optimized for our final goal, that is, a robust distance measure between images. However, as shown in the model benchmarks, using the pairwise information alone does not improve the embedding performance as the process of sampling pairs loses the label information, which is arguably more discriminative than (dis)similar pairs.

5.3. Multitask Learning: Concurrently Optimize Both Tasks

Both methods above have their pros and cons. Learning with classification leverages the label information, but the network is not directly optimized to our goal. As a result, although the feature vectors are learned to be separable in the linear space, the intra- and inter- categorical distances might still be unbalanced. On the other hand, metric learning is explicitly optimized for our final objective by pushing the distances between dissimilar food items apart beyond a margin m. Nevertheless, sampling the similar or dissimilar pairs loses valuable label information. For example, given a pair of items with different labels, we only consider the dissimilarity between the two categories they belong to, but overlook the fact that each item also differs from the remaining $n - 2$ categories, where n is the total number of categories.

To leverage the benefits of both tasks, we propose a multitask learning design [26] for FoodDist. The idea of multitask learning is to share part of the model across tasks to improve the generalization ability of the learned model [26]. In our case, as Figure 6 shows, we share the parameters between the classification network and Siamese network and optimize them simultaneously. We use the base structure of the Siamese network and share the upper CNN with a classification network where the output of the CNN is fed into a cascade of a fully connected layer and a softmax loss layer. The final loss of the whole network is the weighted sum of the softmax loss $\mathcal{L}_{\text{softmax}}$ and contrastive loss $\mathcal{L}_{\text{contrastive}}$:

$$\mathcal{L} = \omega\mathcal{L}_{\text{softmax}} + (1-\omega)\mathcal{L}_{\text{contrastive}} \tag{12}$$

Our benchmark results (Section 6.2) suggest that the feature extractor built with multitask learning achieves the best of both worlds: It achieves the best performance for both classification and Euclidean distance-based retrieval tasks.

6. EVALUATION

- **H1:** Our online learning framework learns more accurate food preference profile than baseline approaches.
- **H2:** FoodDist generates better similarity measure for food images than state-of-the-art embedding models.
- **H3:** Yum-me makes more accurate nutritionally appropriate meal recommendations than traditional surveys, as it integrates coarse-grained item filtering (provided by a survey) with fine-grained food preference learned through adaptive elicitation.

In this section, we first present user testing results for the online learning framework in Section 6.1 and then the offline benchmark *FoodDist* model with a large-scale real-world food image dataset in Section 6.2 and, finally, discuss the results of end-to-end user testing in Section 6.3.

6.1. User Testing for Online Learning Framework

To evaluate the accuracy of our online learning framework, we conducted a field study among 227 anonymous users recruited from social networks and university mailing lists. The experiment was approved by Institutional Review Board (ID: 1411005129) at Cornell University. All participants were required to use this system independently 3 times. Each time the study consisted of following two phases:

- *Training Phase.* Users conducted the first T iterations of food image comparisons, and the system learnt and elicited preference vector $\boldsymbol{p}^T$ based on the algorithms proposed in this article or baseline methods, which will be discussed later. We randomly picked T from set $\{5, 10, 15\}$ at the beginning but made sure that each user experienced different values of T only once.
- *Testing Phase.* After T iterations of training, users entered the testing phase, which consisted of 10 rounds of pairwise comparisons. We picked testing images based on preference vector $\boldsymbol{p}^T$ that were learned from online interactions: One of them was selected from food area that user liked (i.e., item with top 1% preference value) and the other one from the area that user disliked (i.e., item with bottom 1% preference value). Both of the images were picked *randomly* among *unexplored* food items.

6.1.1. Prediction Accuracy. To evaluate the effectiveness of *user state update* and *images selection* methods, respectively, we conduct a 2-by-2 experiment in this section. For the *user state update* method, we compare the proposed *Label propagation, Exponentiated Gradient* (**LE**) algorithm to *Online Perceptron* (**OP**), and for the *images selection* method, we compare the proposed *Exploration-Exploitation* (**EE**) algorithm to the *Random Selection* (**RS**). Specifically, four frameworks presented below are evaluated. Users encountered them *randomly* when they logged into the system.

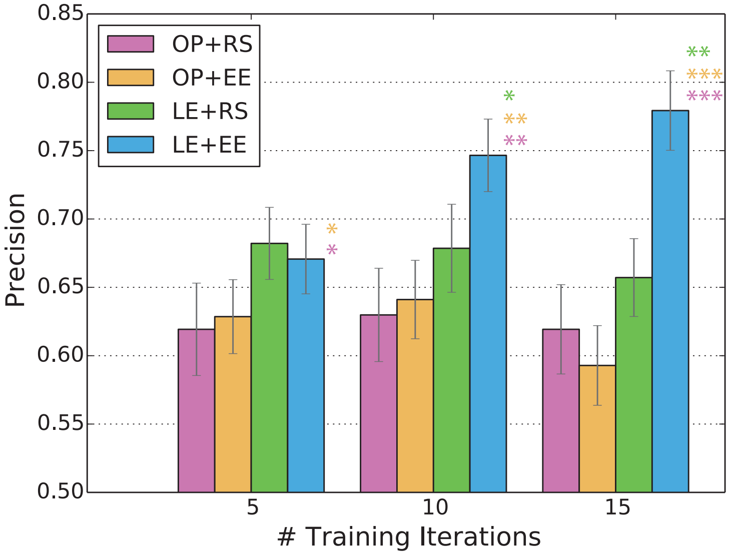


Fig. 7. Prediction accuracy for different algorithms in various training settings (asterisks represent different levels of statistical significance: $***: p < 0.001$, $**: p < 0.01$, $*: p < 0.05$).

LE+EE: This is the online learning algorithm proposed in this article that combines the ideas of **L**abel propagation, the **E**xponentiated Gradient algorithm for user state update, and **E**xploitation-**E**xploration strategy for images selection.

LE+RS: This algorithm retains our method for user state update (**LE**) but **R**andom **S**elect images to present to user without any exploitation or exploration.

OP+EE: As each item is represented by a *1,000-dimensional* feature vector, we can adopt the idea of regression to tackle this online learning problem (i.e., learning weight vector $\boldsymbol{w}$ such that $\boldsymbol{w} f(x_i)$ is higher for item i that user prefer). Hence, we compare our method with **O**nline **P**erceptron algorithm that updates $\boldsymbol{w}$ whenever it makes error, that is, if $y_i \boldsymbol{w} f(x_i) \leq 0$, then assign $\boldsymbol{w} \leftarrow \boldsymbol{w} + y_i \boldsymbol{w} f(x_i)$, where y_i is the label for item i (pairwise comparison is regarded as binary classification such that the food item that user select is labeled as +1 and otherwise -1). In this algorithm, we retain our strategy of images selection (i.e., **EE**).

OP+RS: The last algorithm is the **O**nline **P**erceptron mentioned above but with **R**andom images **S**election strategy.

Of the 227 participants in our study, 58 of them finally used algorithm **LE+EE** and 57 used **OP+RS**. For the rest of the users (112), half of them (56) tested **OP+EE** and the other half (56) tested **LE+RS**. Overall, the participants for different algorithms are totally random so the performances of different models are directly comparable.

After all users go through the training and testing phases, we calculate the prediction accuracy of each *individual user* and aggregate them based on the context that they encountered (i.e. the number of training iterations T and the algorithm settings mentioned above). The prediction accuracies and their cumulative distributions are shown in Figures 7, 8, and 9 respectively.

Length effects of training iterations. As shown in Figure 7 and Figure 8, the prediction accuracies of our online learning algorithm are all significantly higher than the baselines.The algorithm performance is further improved with longer training period. As is clearly shown in Figure 8, when the number of training iterations reaches 15, about half of the users will experience the prediction accuracy that exceeds 80%, which is fairly promising and decent considering the small number of interactions that the

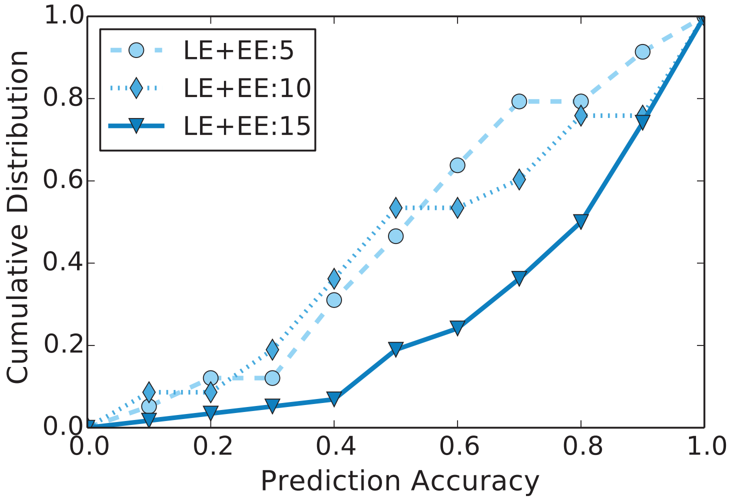
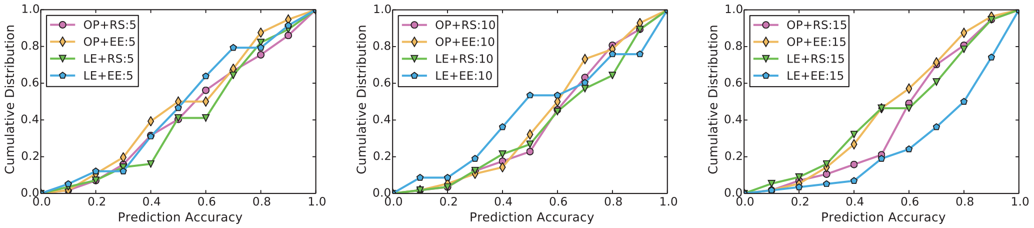


Fig. 8. Cumulative distribution of prediction accuracy for *LE+EE* algorithm (Numbers in the legend represent the number of training iterations (i.e., values of T)).



(a) # training iterations: 5

(b) # training iterations: 10

(c) # training iterations: 15

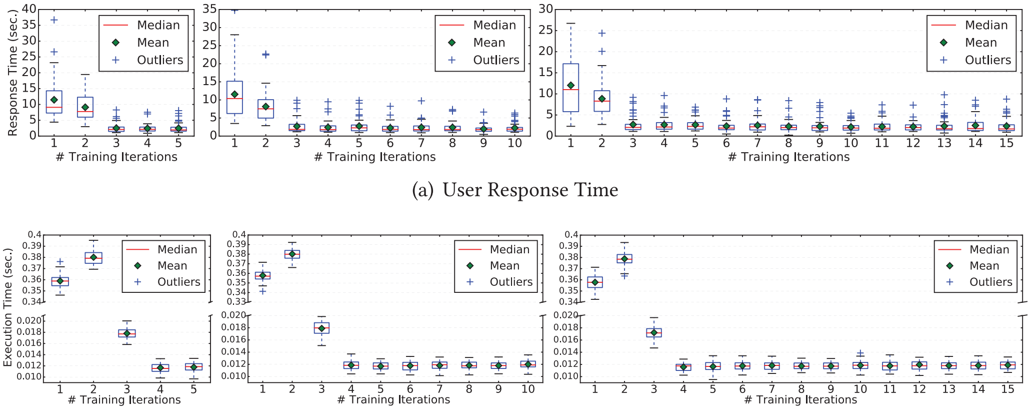
Fig. 9. Comparison of cumulative distribution of prediction accuracy across different algorithms.

system elicited from scratch. The results above justify that the online preference learning algorithm can adjust itself to explore users' preference area as more information is available from their choices. For the task of item-based food preference bootstrapping, our system can efficiently balance the exploration-exploitation while providing reasonably accurate predictions.

Comparisons across different algorithms. As mentioned previously, we compared our algorithm with several obvious alternatives. As shown in Figure 7 and Figure 9, none of these algorithms works very well and the accuracy of prediction is actually decreasing as the user provides more information. Additionally, as is shown in Figure 9, our algorithm has particular advantages when users are making progress (i.e., the number of training iterations reaches 15). The reason why these techniques are not suited for our application is mainly due to the following limitations:

Random Selection. Within a limited number of interactions, random selection cannot maintain the knowledge that it has already learned about the user (exploitation) or explore unknown areas (exploration). In addition, it is more likely that the system will choose food items that are very similar to each other and thus hard for the user to make decisions. Therefore, after short periods of interactions, the system is messed up, and the performance degrades.

Underfitting. The algorithm that will possibly have the underfitting problem is the online perceptron (**OP**). For our application, each food item is represented by a



(a) User Response Time

(b) System Execution Time

Fig. 10. Timestamp records for user response time and system execution time.

Table II. Average Time to Complete Training Phase

# Iter: 5	# Iter: 10	# Iter: 15
28.75s	39.74s	53.22s

1,000-dimensional feature vector, and **OP** is trying to learn a separate hyperplane based on a limited number of training data. As each single feature is directly derived from a deep neural network, the linearity assumptions made by the perceptron might yield wrong predictions for the dishes that have not been explored before.

6.1.2. System Efficiency. As another two aspects of online preference elicitation system, computing efficiency and user experience are also very important metrics for system evaluation. Therefore, we recorded the program execution time and user response time as a lens into the real-time performance of the online learning algorithm. As shown in Figure 10(b), the program execution time is about 0.35s for the first two iterations and less than 0.025s for the iterations afterwards.[4] Also, according to Figure 10(a), the majority of users can make their decisions in less than 15s for the task of comparison among 10 food images while the payload for the pairwise comparison is less than 2 to 3s. As a final cumulative metric for the system overhead, it is shown in Table II that even for 15 iterations of training, users can typically complete the whole process within 53s, which further justify that our online learning framework is lightweight and user friendly in efficiently eliciting food preference.

6.1.3. User Qualitative Feedback. After the study, some participants send us emails regarding their experiences towards the adaptive visual interface. Most of the comments reflect the participants' satisfactions and that our system is able to engage the user throughout the elicitation process, for example, "*Now I'm really hungry and want a grilled cheese sandwhich!*," "*That was fun seeing tasty food at top of the morning*," and "*Pretty cool tool.*" However, they also highlight some limitations of our current prototype, for example, "*I am addicted to spicy food and it totally missed it. There may just not be enough spicy alternatives in the different dishes to pick up on it*" points out that the prototype is limited in the size of the food database.

[4] Our web system implementation is based on Amazon EC2 t2-micro Linux 64-bit instance

Table III. Model Performance of Classification Task. ∗ Represents State-of-the-Art Approach and Bold Text Indicates the Method with the Best Performance

Method	Top-1 ACC (%)	Top-5 ACC(%)
RFDC∗ [6]	50.76%	—
GoogleLeNet∗ [35]	79%	—
AlexNet+CL	67.63%	89.02%
AlexNet+MT	70.50%	90.36%
VGG+CL	82.48%	95.70%
VGG+MT **(FoodDist)**	**83.09%**	**95.82%**

6.2. Offline Benchmarking for FoodDist

We develop FoodDist and baseline models (Section 5) using the Food-101 training dataset, which contains 75,750 food images from 101 food categories (750 instances for each category) [6]. To the best of our knowledge, Food-101 is the largest and most challenging publicly available dataset for food images. We implement models using Caffe [27] and experiment with two CNN architectures in our framework: AlexNet [33], which won first place at the ILSVRC2012 challenge, and VGG [48], which is the state-of-the-art CNN model. The inputs to the networks are image crops of sizes 224×224 (VGG) or 227 × 227 (AlexNet). They are randomly sampled from a pixelwise mean-subtracted image or its horizontal flip. In our benchmark, we train four different feature extractors: AlexNet+Learning with classification (**AlexNet+CL**), AlextNet+Multitask learning (**AlexNet+MT**), VGG+Learning with classification (**VGG+CL**), and VGG+Multitask learning (**VGG+ML, FoodDist**). For the multitask learning framework, we sample the similar and dissimilar image pairs with 1:10 ratio from the Food-101 dataset based on the categorical labels to be consistent with the previous work [59]. The models are fine-tuned based on the networks pre-trained with the ImageNet data. We use Stochastic Gradient Decent with a mini-batch size of 64, and each network is trained for 10×10^4 iterations. The initial learning rate is set to 0.001, and we use a weight decay of 0.0005 and momentum of 0.9.

We compare the performance of four feature extractors, including FoodDist, with the state-of-the-art food image analysis models using the Food-101 testing dataset, which contains 25,250 food images from 101 food categories (250 instances for each category). The performance for the *classification* and *retrieval* tasks are evaluated as follows:

- **Classification:** We test the performance of using learned image features for classification. For the classification deep neural network in each of the models above, we adopt the standard 10-crop testing. that is, the network makes a prediction by extracting 10 patches (the four corner patches and the center patch in the original images and their horizontal reflections) and averaging the predictions at the softmax layer. The metrics used in this article are Top-1 accuracy and Top-5 accuracy.
- **Retrieval:** We use a retrieval task to evaluate the quality of the Euclidean distances between extracted features. Ideally, the distances should be smaller for similar image pairs and larger for dissimilar pairs. Therefore, as suggested by previous work [59, 61], we check the nearest k-neighbors of each test image, for $k = 1, 2, \dots, N$, where $N = 25,250$ is the size of the testing dataset, and calculate the Precision and Recall values for each k. We use mean Average Precision (mAP) as the evaluation metric to compare the performance. For every method, the Precision/Recall values are averaged over all the images in the testing set.

The classification and retrieval performance of all models are summarized in Table III and Table IV, respectively. FoodDist performs the best among four models and is significantly better than the state-of-the-art approaches in both tasks. For the classification

Table IV. Model Performance of Retrieval Task. ∗ Represents State-of-the-Art Approach and Bold Text Indicates the Method with the Best Performance

Method	mean Average Precision (mAP)
Food-CNN∗ [59]	0.3084
AlexNet+CL	0.3751
AlexNet+MT	0.4063
VGG+CL	0.6417
VGG+MT **(FoodDist)**	**0.6670**

Note: The mAP value that we report for food-cnn is higher because we use pixel-wise mean subtraction while the original paper only used per-channel mean subtraction.

task, the classifier built on FoodDist features achieves 83.09% Top-1 accuracy, which significantly outperforms the original RFDC [6] model and the proprietary GoogLeNet model [35]; for the retrieval task, FoodDist doubles the mAP value reported by previous work [59] that only used the AlexNet and Siamese network architecture. The benchmark results demonstrate that FoodDist features possess high generalization ability and the Euclidean distances between feature vectors reflect the similarities between food images with great fidelity. In addition, as we can observe from both tables, the multitask learning-based approach always performs better than learning with classification for both tasks no matter which CNN is used. This further justifies the proposed multitask learning approach and its advantage of incorporating both label and pairwise distance information that makes the learned features more generalizable and meaningful in the Euclidean distance embedding.

6.3. End-to-End User Testing

We conducted end-to-end user testing to validate the efficacy of Yum-me recommendations. We recruited 60 participants through the university mailing list, Facebook, and Twitter. The goal of the user testing was to compare Yum-me recommendations with a widely used user onboarding approach, that is, a traditional food preference survey (a sample survey used by PlateJoy is shown in Figure 13). As Yum-me is designed for scenarios where no rating or food consumption history is available (which is common when the user is new to a platform or is visiting a nutritionist's office), a collaborative filtering algorithm that has been adopted by many state-of-the-art recommenders is not directly comparable to our system.

In this study, we used a within-subjects study design in which each participant expressed their opinions regarding the meals recommended by both of the recommenders, and the effectiveness of the systems were compared on a per-user basis.

6.3.1. Study Design. We created a traditional recommendation system by randomly picking N of M meals in the candidate pool to recommend to the users. The values of N and M are controlled such that $N = 10, M = 500$ for both Yum-me and the traditional baseline. The user study consists of three phases, as Figure 11 shows: (1) Each participant was asked to indicate his or her diet type and health goals through our basic user survey. (2) Each participant was then asked to use the visual interface. (3) Twenty meal recommendations were arranged in a random order and presented to the participant at the same time, where 10 of them are made by Yum-me, and the other 10 are generated by the baseline. The participant was asked to express their opinion by dragging each of the 20 meals into either the *Yummy* or the *No way* bucket. To overcome the fact that humans would tend to balance the buckets if their previous

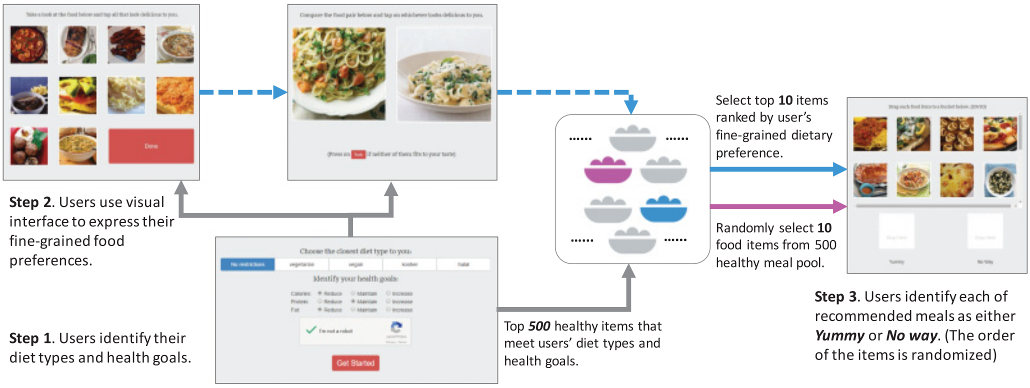

Step 2. Users use visual interface to express their fine-grained food preferences.
Step 1. Users identify their diet types and health goals.
Choose the closest diet type to you:
Identify your health goals:
Get Started
Top 500 healthy items that meet users' diet types and health goals.
Select top 10 items ranked by user's fine-grained dietary preference.
Randomly select 10 food items from 500 healthy meal pool.
Step 3. Users identify each of recommended meals as either Yummy or No way. (The order of the items is randomized)

Fig. 11. User study workflow for personalized nutrient-based meals recommendation system. We compare Yum-me (blue arrows) with the baseline method (violet arrow) that makes recommendations solely based on nutritional facts and dietary restrictions.

Table V. Statistics of Health Goals Among 60 Participants. Unit: Number of Participants

Nutrient	Reduce	Maintain	Increase
Calories	30	28	2
Protein	1	44	15
Fat	23	36	1

choices were shown, the food item disappeared after the user dragged it into a bucket. In this way, users were not reminded of how many meals they had put into each bucket.

The user study systems were implemented as web services and participants accessed the study from desktop or mobile browsers. We chose a web service for its wide accessibility to the population, but we could easily fit Yum-me into other ubiquitous devices, as mentioned earlier.

6.3.2. Participants. The most common dietary choice among our 60 participants was *No restrictions* (48), followed by *Vegetarian* (9), *Halal* (2), and *Kosher* (1). No participants chose *Vegan*. Participant preferences in terms of nutrients are summarized in Table V. For *Calories* and *Fat*, the top two goals were *Reduce* and *Maintain*. For *Protein*, participants tended to choose either *Increase* or *Maintain*. For health goals, the top four participant choices were *Maintain calories-Maintain protein-Maintain fat* (20), *Reduce calories-Maintain protein-Reduce fat* (10), *Reduce calories-Maintain protein-Maintain fat* (10) and *Reduce calories-Increase protein-Reduce fat* (5). The statistics match well with the common health goals among the general population, that is, people who plan to control weight and improve sports performance tend to reduce the intake calories and fat and increase the amount of protein.

6.3.3. Quantitative Analysis. We use a quantitive approach to demonstrate that (1) Yum-me recommendations yield higher meal acceptance rates than traditional approaches and (2) meals recommended by Yum-me satisfy users' nutritional needs.

To show higher meal acceptance rates, we calculated the participant acceptance rate of meal recommendations as

$$\frac{\text{\# Meals in Yummy bucket}}{\text{\# Recommended meals}}.$$

The cumulative distribution of the acceptance rate is shown in Figure 12, and the average acceptance rate, Mean Absolute Error (MAE), and Root Mean Square Error

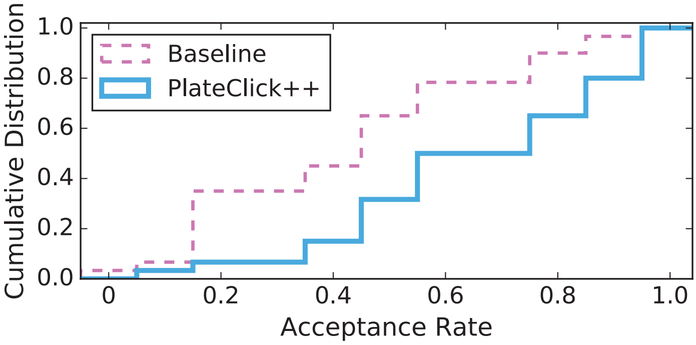


Fig. 12. Cumulative distribution of acceptance rate for both recommender systems.

Table VI. Average Acceptance Rates (Avg. Acc.), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) between two Systems. Paired *t*-test P-value (Avg. Acc.): 8.58×10^{-10}

Metric	Mean	SEM
Yum-me Avg. Acc.	**0.7250**	**0.0299**
Baseline Avg. Acc.	0.5083	0.0341
Yum-me MAE	**0.2750**	**0.0299**
Baseline MAE	0.4916	0.0341
Yum-me RMSE	**0.4481**	**0.0355**
Baseline RMSE	0.6649	0.0290

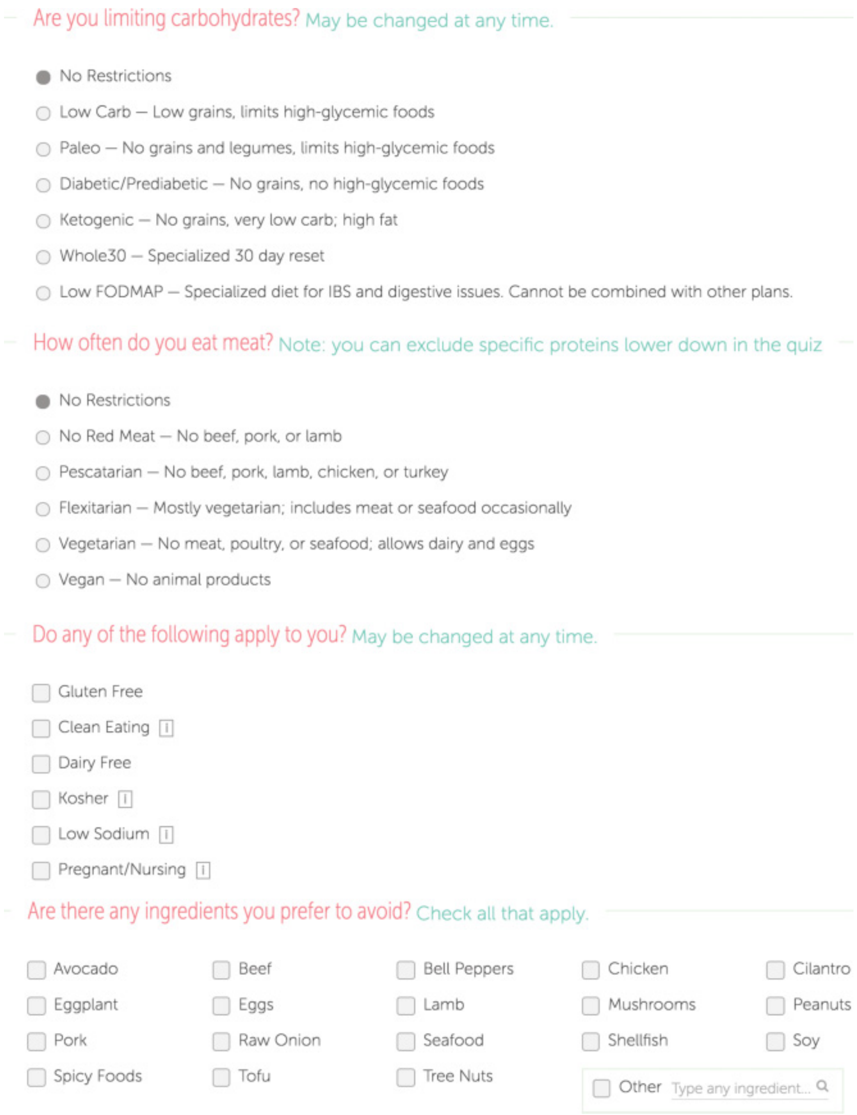
(RMSE) of each approach are presented in Table VI. The results demonstrate that Yum-me significantly improves the quality of the presented food items. The per-user acceptance rate difference between two approaches was normally distributed,[5] and a paired Student's t-test indicated a significant difference between the two methods ($p < 0.0001$).[6]

To quantify the improvement provided by *Yum-me*, we calculated the difference between the acceptance rates of the two systems, that is, *difference* = *Yum-me acceptance rate* − *baseline acceptance rate*. The distribution and average values of the differences are presented in Figure 14 and Table VI, respectively. It is noteworthy that Yum-me outperformed the baseline by 42.63% in terms of the number of preferred recommendations, which demonstrates its utility over the traditional meal recommendation approach. However, another observed phenomenon in Figure 14 is that there are 12 users (20%) with zero acceptance rate differences, which may due to the following two reasons: (1) Yum-me is not effective to this set of users, and it does not improve their preferences towards recommended food items. (2) As we did not conduct participant control and filtering, some participants may not be well involved in the study and randomly select or drag items.

To examine meal nutrition, we compare the nutritional facts of paticipants' favorite meals with those of meals recommended (by Yum-me) and accepted (items dragged into the *yummy* bucket) by the user. As shown in Figure 15, for users with the same nutritional needs and no dietary restrictions, we calculate the average amount of protein, calories, and fat (per-serving) in (1) their favorite 20 meals (as determined by our

[5] A Shapiro Wilk W test was not significant ($p = 0.12$), which justifies that the difference is normally distributed.

[6] We also performed a non-parametric Wilcoxon signed-rank test and found a comparable result.



Are you limiting carbohydrates? May be changed at any time.

● No Restrictions
○ Low Carb — Low grains, limits high-glycemic foods
○ Paleo — No grains and legumes, limits high-glycemic foods
○ Diabetic/Prediabetic — No grains, no high-glycemic foods
○ Ketogenic — No grains, very low carb; high fat
○ Whole30 — Specialized 30 day reset
○ Low FODMAP — Specialized diet for IBS and digestive issues. Cannot be combined with other plans.

How often do you eat meat? Note: you can exclude specific proteins lower down in the quiz

● No Restrictions
○ No Red Meat — No beef, pork, or lamb
○ Pescatarian — No beef, pork, lamb, chicken, or turkey
○ Flexitarian — Mostly vegetarian; includes meat or seafood occasionally
○ Vegetarian — No meat, poultry, or seafood; allows dairy and eggs
○ Vegan — No animal products

Do any of the following apply to you? May be changed at any time.

□ Gluten Free
□ Clean Eating
□ Dairy Free
□ Kosher
□ Low Sodium
□ Pregnant/Nursing

Are there any ingredients you prefer to avoid? Check all that apply.

□ Avocado □ Beef □ Bell Peppers □ Chicken □ Cilantro
□ Eggplant □ Eggs □ Lamb □ Mushrooms □ Peanuts
□ Pork □ Raw Onion □ Seafood □ Shellfish □ Soy
□ Spicy Foods □ Tofu □ Tree Nuts □ Other Type any ingredient...

Fig. 13. The survey used for user onboarding of PlateJoy. The questions are up to date at the time of the writing of this article, and we only include top four questions for illustration purpose.

online learning algorithm) and (2) their recommended and accepted meals, respectively. The mean values presented in Figure 15 are normalized by the average amount of corresponding nutrients in their favorite meals. The results demonstrate that by using a relatively simple nutritional ranking approach, Yum-me is able to satisfy most of the nutritional needs set by the users, including *reduce, maintain and increase calories*, *increase protein*, and *reduce fat*. However, our system fails to meet two nutritional requirments, that is, *maintain protein* and *maintain fat*. Our results also show where Yum-me recommendations result in unintended nutritional composition. For example, the goal of *reducing fat* results in the reduction of *protein* and *calories*, and the goal of *increasing calories* ends up increasing the *protein* in meals. This is partially due to

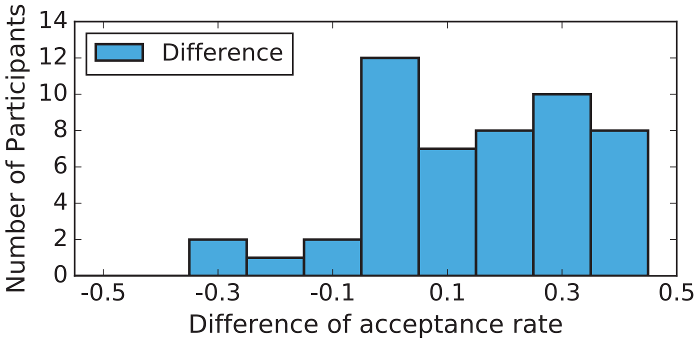


Fig. 14. Distribution of the acceptance rate differences between two recommender systems.

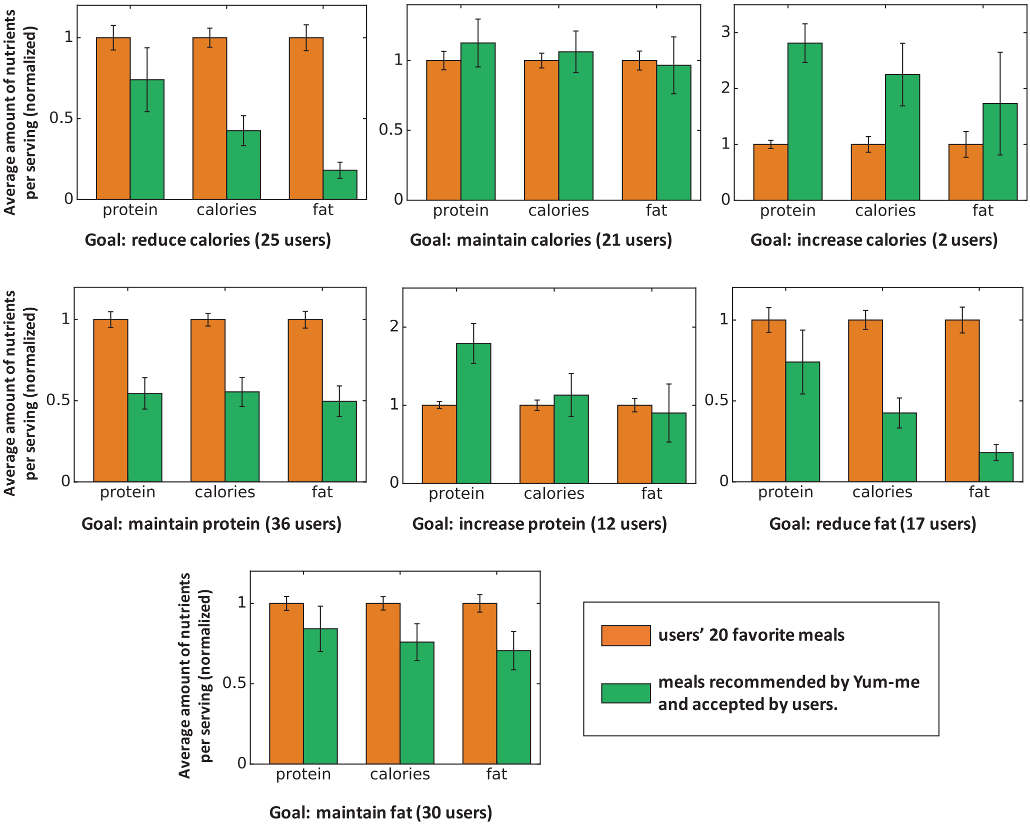


Fig. 15. Nutritional facts comparison between paticipants' favorite meals and recommended (Yum-me) and accepted meals. The meal is accepted if it is dragged into the *yummy* bucket. The mean values are normalized by the average amount of corresponding nutrient in the favorite meals (orange bar). (Only seven of nine nutritional goals are used by at least one partipant.)

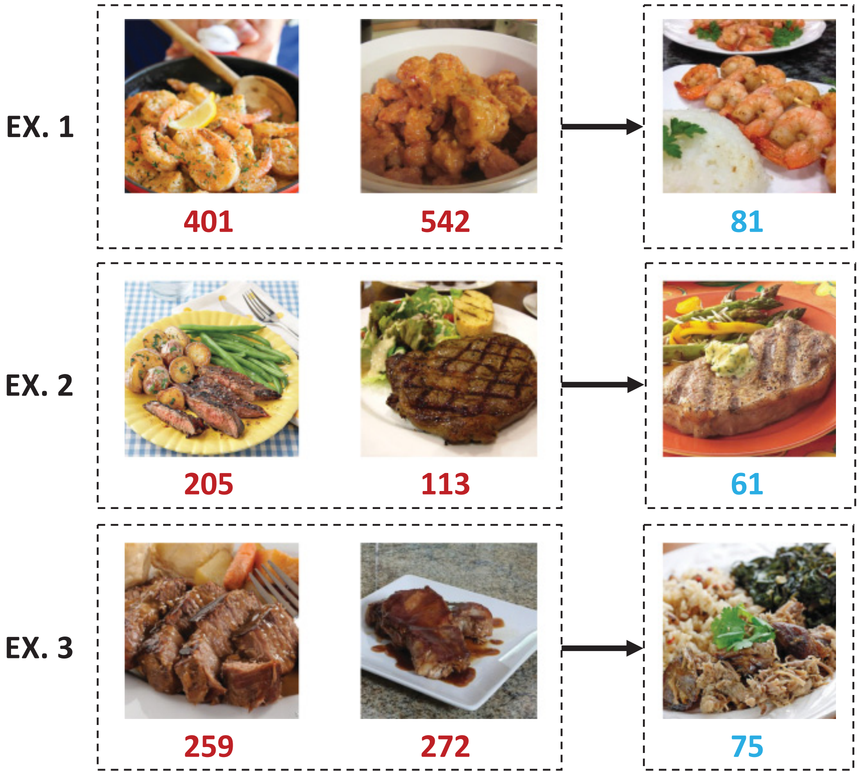


Fig. 16. Qualitative analysis of personalized healthy meal recommendations. Images on the left half are sampled from users' top-20 favorite meals learned from Yum-me; images on the right half are the meals presented to the user. The number under each food image represents the amount of calories for the dish, unit: $kcal/serving$.

the inherent inter-dependence between nutrients, and we leave further investigation of this issue to future work.

6.3.4. Qualitative Analysis. To qualitatively understand the personalization mechanism of Yum-me, we randomly pick three participants with no dietary restrictions and with the health goal of reducing calories. For each user, we select top-20 general food items the user likes most (inferred by the online learning algorithm). These food items played important roles in selecting the healthy meals to recommend to the user. To visualize this relationship, among these top-20 items, we further select two food items that are most similar to the healthy items Yum-me recommended to the users and present three such examples in Figure 16. Intuitively, our system is able to recommend healthy food items that are *visually similar* to the food items a user like, but the recommended items are of lower calories due to the use of healthier ingredients or different cooking styles. These examples showcase how Yum-me can project users' general food preferences to the domain of the healthy options and find the ones that can most appeal to users.

6.3.5. Error Analysis. Through a closer examination of the cases where our system performed, or did not perform, well, we observed a negative correlation between the entropy of the learned preference distribution $\mathbf{p}$[7] and the improvement of Yum-me over

[7] Entropy of preference distribution: $H(\mathbf{p}) = -\sum_i p_i \log p_i$.

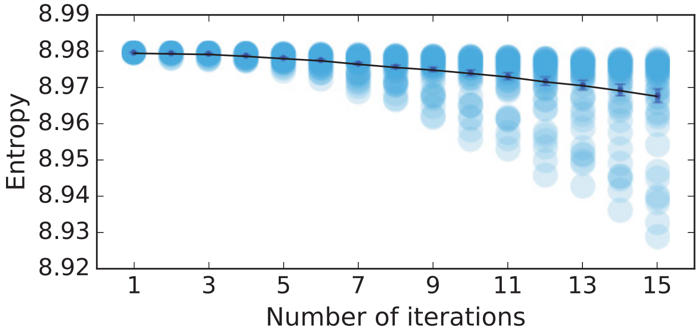


Fig. 17. Entropy of preference distributions in different iterations of online learning. (Data are from 48 users with no dietary restrictions.)

the baseline ($r = -0.32$, $p = 0.026$). This correlation suggests that when user's preference distributions are more concentrated, the recommended meals tend to perform better. This is not too surprising, because the entropy of the preference distribution roughly reflects the degree of confidence the system has in the users' preferences, where the confidence is higher if the entropy is lower and vice versa. In Figure 17, we show the evolution of the entropy value as the users are making more comparisons. The results demonstrate that the system becomes more confident about user's preferences as users provide more feedback.

7. DISCUSSION

In this section, we discuss the limitations of the current prototype and study and present real-world scenarios where *Yum-me* and its sub-modules can be used.

7.1. Limitations of the Evaluations

In evaluating the online learning framework, because there is no previous algorithm that can end-to-end solve our preference elicitation problem, the baselines are constructed by combining methods that intuitively fit *user state update* and *images selection* modules, respectively. This introduces potential biases in baseline selections. Additionally, in the end-to-end user testing, the participants' judgements of whether the food is *Yummy* or *No way* is potentially influenced by the image quality and the health concerns. These may be confounding factors in measuring users' preferences towards food items and can be eliminated by explicitly instructing the participants to not consider these factors. We leave further evaluations as future work.

7.2. Limitations of Yum-me in Recommending Healthy Meals

The ultimate effectiveness of *Yum-me* in generating healthy meal suggestions is contingent on the appropriateness of the nutritional needs input by the user. To conduct such recommendations for people with different conditions, Yum-me could be used in the context of personal health coaches, nutritionists, or coaching applications that provide reliable nutritional suggestions based on the user's age, weight, height, exercise, and disease history. For instance, general nutritional recommendations can be calculated using online services built on the guidelines from National Institutes of Health, such as *weight-success*[8] and *active*.[9] Also, although we have demonstrated the feasibility of building a personalized meal recommender catering to people's fine-grained food

[8] http://www.weighing-success.com/NutritionalNeeds.html.

[9] http://www.active.com/fitness/calculators/nutrition.

preference and nutritional needs, the current prototype of Yum-me assumes a relatively simple strategy to rank the nutritional appropriateness, and is limited in terms of the available options for nutrition. Future work should investigate more sophisticated ranking approaches and incorporate options relevant to the specific application context.

7.3. Yum-Me for Real-World Dietary Applications

We envision that Yum-me has the potential to power many real-world dietary applications. For example, (1) *User onboarding*. Traditionally, food companies, for example, Zipongo and Plated, address the cold start problem by asking each new user to answer a set of pre-defined questions, as shown in Section 6.3, and then recommend meals accordingly. Yum-me can enhance this process by eliciting user's fine-grained food preference and informing an accurate dietary profile. Service providers can customize Yum-me to serve their own businesses and products by using a specialized backend food item database and then use it as a step after the general questions. (2) *Nutritional assistants.* While visiting a doctor's office, patients are often asked to fill out standard questionnaires to indicate food preferences and restrictions. Patients' answers are then investigated by the professionals to come up with effective and personalized dietary suggestions. In such a scenario, the recommendations made by Yum-me could provide a complementary channel for communicating the patient's fine-grained food preferences to the doctor to further tailor suggestions.

7.4. FoodDist for a Wide Range of Food Image Analysis Tasks

FoodDist provides a unified model to extract features from food images so they are discriminative in the classification and clustering tasks, and its pairwise Euclidean distances are meaningful in reflecting similarities. The model is rather efficient ($<$0.5s/f on eight-core commodity processors) and can be ported to mobile devices with the publicly available caffe-android-lib framework.[10]

In addition to enabling Yum-me, we released the FoodDist model to the community (https://github.com/ylongqi/FoodDist), so it can be used to fuel other nutritional applications. For the sake of space, we only briefly discuss two sample use cases below:

- **Food/Meal recognition:** Given a set of labels, for example, food categories, cuisines, and restaurants, the task of food and meal recognition could be approached by first extracting food image features from FoodDist and then training a linear classifier, for example, logistic regression or SVM, to classify the food images that are beyond the categories given in the Food-101 dataset.
- **Nutrition Facts estimation:** With the emergence of large-scale food item or recipe databases, such as Yummly, the problem of nutritional fact estimation might be converted to a simple nearest-neighbor retrieval task: Given a query image, we find its closest neighbor in the FoodDist based on Euclidean distance and use that neighbor's nutritional information to estimate the nutrition facts of the query image [35].

8. CONCLUSION AND FUTURE WORK

In this article, we propose Yum-me, a novel nutrient-based meal recommender that makes meal recommendations catering to users' fine-grained food preferences and nutritional needs. We further present an online learning algorithm that is capable of efficiently learning food preference and FoodDist, a best-of-its-kind unified food image analysis model. The user study and benchmarking results demonstrate the effectiveness of Yum-me and the superior performance of the FoodDist model.

[10] https://github.com/sh1r0/caffe-android-lib.

Looking forward, we envision that the idea of using visual similarity for preference elicitation may have implications to the following research areas. (1) **User-centric modeling**: the fine-grained food preference learned by Yum-me can be seen as a general dietary profile of each user and be projected to other domains to enable more dietary applications, such as suggesting proper meal plans for diabetes patients. Moreover, a personal dietary API can be built on top of this profile to enable sharing and improvementment across multiple dietary applications. (2) **Food image analysis API for deeper content understanding**: With the release of the FoodDist model and API, many dietary applications, in particular the ones that capture a large number of food images, might benefit from a deeper understanding of their image contents. For instance, food journaling applications could benefit from the automatic analysis of food images to summarize the day-to-day food intake or trigger timely reminders and suggestions when needed. (3) **Fine-grained preference elicitation leveraging visual interfaces.** The idea of eliciting users' fine-grained preference via visual interfaces is also applicable to other domains. The key insight here is that visual contents capture many subtle variations among objects that text or categorical data cannot capture, and the learned representations can be used as an effective medium to enable fine-grained preferences learning. For instance, the IoT, wearable, and mobile systems for entertainments, consumer products, and general content deliveries might leverage such an adaptive visual interface to design an onboarding process that learn users' preferences in a much shorter time and potentially provide a more pleasant user experience than traditional approaches.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments and thank Yin Cui, Fan Zhang, Tsung-Yi Lin, and Dr. Thorsten Joachims for discussions of machine-learning algorithms.

REFERENCES

[1] Lenore Arab, Deborah Estrin, Donnie H. Kim, Jeff Burke, and Jeff Goldman. 2011. Feasibility testing of an automated image-capture method to aid dietary recall. *Eur. J. Clin. Nutr.* 65, 10 (2011), 1156–1162.

[2] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*.

[3] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.* 32, 1 (2002), 48–77.

[4] Oscar Beijbom, Neel Joshi, Dan Morris, Scott Saponas, and Siddharth Khullar. 2015. Menu-match: restaurant-specific food logging from images. In *Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 844–851.

[5] Lisa M. Bodnar and Katherine L. Wisner. 2005. Nutrition and depression: implications for improving mental health among childbearing-aged women. *Biol. Psychiat.* 58, 9 (2005), 679–685.

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*. Springer International Publishing, 446–461.

[7] J. R. Brotherhood. 1984. Nutrition and sports performance. *Sports Med.* 1, 5 (1984), 350–389.

[8] Rich Caruana. 1997. Multitask learning. *Mach. Learn,* 28, 1 (1997), 41–75.

[9] Junghoon Chae, Insoo Woo, SungYe Kim, Ross Maciejewski, Fengqing Zhu, Edward J. Delp, Carol J. Boushey, and David S. Ebert. 2011. Volume estimation using food specific shape templates in mobile image-based dietary assessment. In *Computational Imaging.* 78730K.

[10] Kerry Shih-Ping Chang, Catalina M. Danis, and Robert G. Farrell. 2014. Lunch Line: Using public displays and mobile devices to encourage healthy eating in an organization. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 823–834.

[11] Shuo Chang, F. Maxwell Harper, and Loren Terveen. 2015. Using groups of items for preference elicitation in recommender systems. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'15)*.

[12] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y. Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, et al. 2008. Activity sensing in the wild: aA field trial of ubifit garden. In *Proceedings of the Special Interest Group on Computer-Human Interaction Conference on Human Factors in Computing Systems*. ACM, 1797–1806.

[13] Felicia Cordeiro, Elizabeth Bales, Erin Cherry, and James Fogarty. 2015. Rethinking the mobile food journal: Exploring opportunities for lightweight photo-based capture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 3207–3216.

[14] Jifeng Dai, Kaiming He, and Jian Sun. 2016. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3150–3158.

[15] Mahashweta Das, Gianmarco De Francisci Morales, Aristides Gionis, and Ingmar Weber. 2013. Learning to question: leveraging user preferences for shopping advice. In *Proceedings of the Conference Knowledge Discovery in Databases* (*KDD'13)*.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009 (CVPR 2009)*. IEEE, 248–255.

[17] David Elsweiler and Morgan Harvey. 2015. Towards automatic meal plan recommendations for balanced nutrition. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 313–316.

[18] Leonard H. Epstein, Rena R. Wing, Barbara C. Penner, and Mary Jeanne Kress. 1985. Effect of diet and controlled exercise on weight loss in obese children. *J. Pediatr.* 107, 3 (1985), 358–361.

[19] Peter Forbes and Mu Zhu. 2011. Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation. In *Proceedings of the 5th ACM Conference on Recommender Systems*. ACM, 261–264.

[20] Jill Freyne and Shlomo Berkovsky. 2010. Intelligent food planning: personalized recipe recommendation. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*. ACM, 321–324.

[21] Gijs Geleijnse, Peggy Nachtigall, Pim van Kaam, and Luciënne Wijgergangs. 2011. A personalized recipe advice system to promote healthful choices. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*. ACM, 437–438.

[22] Nadav Golbandi, Yehuda Koren, and Ronny Lempel. 2011. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM'11)*.

[23] Morgan Harvey, Bernd Ludwig, and David Elsweiler. 2013. You are what you eat: Learning user tastes for rating prediction. In *Proceedings of the International Symposium on String Processing and Information Retrieval*. Springer, 153–164.

[24] Ye He, Chang Xu, Neha Khanna, Carol J. Boushey, and Edward J. Delp. 2013. Food image analysis: Segmentation, identification and weight estimation. In *Proceedings of the 2013 IEEE International Conference on Multimedia and Expo (ICME'13)*. IEEE, 1–6.

[25] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative Metric Learning. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee.

[26] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. 2016. Deep Learning. Retrieved from http://goodfeli.github.io/dlbook/.

[27] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 675–678.

[28] Azusa Kadomura, Cheng-Yuan Li, Yen-Chang Chen, Hao-Hua Chu, Koji Tsukada, and Itiro Siio. 2013. Sensing fork and persuasive game for improving eating behavior. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*. ACM, 71–74.

[29] Azusa Kadomura, Cheng-Yuan Li, Koji Tsukada, Hao-Hua Chu, and Itiro Siio. 2014. Persuasive technology to improve eating behavior using a sensor-embedded fork. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 319–329.

[30] Yoshiyuki Kawano and Keiji Yanai. 2014. Food image recognition with deep convolutional features. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 589–593.

[31] Keigo Kitamura, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2009. FoodLog: capture, analysis and retrieval of personal food images via web. In *Proceedings of the ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating Activities*. ACM, 23–30.

[32] Robert C. Klesges, Linda H. Eck, and JoAnne W. Ray. 1995. Who underreports dietary intake in a dietary recall? Evidence from the Second National Health and Nutrition Examination Survey. *J. Consult. Clinical Psychol.* 63, 3 (1995), 438.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.

[34] David G. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60, 2 (2004), 91–110.

[35] Austin Meyers, Nick Johnston, Vivek Rathod, Anoop Korattikara, Alex Gorban, Nathan Silberman, Sergio Guadarrama, George Papandreou, Jonathan Huang, and Kevin P. Murphy. 2015. Im2Calories: towards an automated mobile vision food diary. In *Proceedings of the International Conference on Computer Vision (ICCV'15)*. 1233–1241.

[36] Kher Hui Ng, Victoria Shipp, Richard Mortier, Steve Benford, Martin Flintham, and Tom Rodden. 2015. Understanding food consumption lifecycles using wearable cameras. *Pers. Ubiq. Comput.* 19, 7 (2015), 1183–1195.

[37] Jon Noronha, Eric Hysen, Haoqi Zhang, and Krzysztof Z Gajos. 2011. Platemate: crowdsourcing nutritional analysis from food photographs. In *Proceedings of the ACM Symposium of User Interface Software and Technology (UIST'11)*. ACM, 1–12.

[38] nutrino. 2016. nutrino. Retrieved from http://nutrino.co/.

[39] Seung-Taek Park and Wei Chu. 2009. Pairwise preference regression for cold-start recommendation. In *Proceedings of the 3rd ACM Conference on Recommender Systems*. ACM, 21–28.

[40] platejoy. 2016. PlateJoy. Retreived from https://www.platejoy.com/.

[41] Rachel Clare Povey and David Clark-Carter. 2007. Diabetes and healthy eating: A systematic review of the literature. *Diabetes Educator* 33, 6 (2007), 931–959.

[42] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. McNee, Joseph A. Konstan, and John Riedl. 2002. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the ACM Intelligent User Interfaces Conference (IUI'02)*.

[43] Ali Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 806–813.

[44] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. ACM, 273–282.

[45] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461.

[46] Jonathan Shepherd, Angela Harden, Rebecca Rees, Ginny Brunton, Jo Garcia, Sandy Oliver, and Ann Oakley. 2006. Young people and healthy eating: a systematic review of research on barriers and facilitators. *Health Educ. Res.* 21, 2 (2006), 239–257.

[47] Shopwell. 2016. Retrieved from http://www.shopwell.com/.

[48] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[49] Kyoko Sudo, Kazuhiko Murasaki, Jun Shimamura, and Yukinobu Taniguchi. 2014. Estimating nutritional value from food images based on semantic segmentation. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 571–576.

[50] Mingxuan Sun, Fuxin Li, Joonseok Lee, Ke Zhou, Guy Lebanon, and Hongyuan Zha. 2013. Learning multiple-question decision trees for cold-start recommendation. In *Proceedings of the ACM International Conferene on Web Search and Data Mining (WSDM'13)*.

[51] Martin Svensson, Kristina Höök, and Rickard Cöster. 2005. Designing and evaluating kalas: A social navigation system for food recipes. *ACM Trans. Comput.-Hum. Interact.* 12, 3 (2005), 374–400.

[52] Edison Thomaz, Aman Parnami, Irfan Essa, and Gregory D Abowd. 2013. Feasibility of identifying eating moments from first-person images leveraging human computation. In *Proceedings of the 4th International SenseCam and Pervasive Imaging Conference*. ACM, 26–33.

[53] Gabrielle M Turner-McGrievy, Elina E Helander, Kirsikka Kaipainen, Jose Maria Perez-Macias, and Ilkka Korhonen. 2015. The use of crowdsourcing for dietary self-monitoring: crowdsourced ratings of food pictures are comparable to ratings by trained observers. *J. Am. Med. Inf. Assoc.* 22, e1 (2015), e112–e119.

[54] Mayumi Ueda, Syungo Asanuma, Yusuke Miyawaki, and Shinsuke Nakajima. 2014. Recipe recommendation method by considering the users preference and ingredient quantity of target recipe. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 1.

[55] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 2579–2605 (2008), 85.

[56] Youri van Pinxteren, Gijs Geleijnse, and Paul Kamsteeg. 2011. Deriving a recipe similarity measure for recommending healthful meals. In *Proceedings of the 16th ACM International Conference on Intelligent User Interfaces*. 105–114.

[57] Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: learning to rank with joint word-image embeddings. *Mach. Learn.* 81, 1 (2010), 21–35.

[58] Jason Weston, Hector Yee, and Ron J Weiss. 2013. Learning to rank recommendations with the k-order statistic loss. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 245–248.

[59] Longqi Yang, Yin Cui, Fan Zhang, John P. Pollak, Serge Belongie, and Deborah Estrin. 2015. PlateClick: Bootstrapping Food Preferences Through an Adaptive Visual Interface. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 183–192.

[60] Longqi Yang, Chen Fang, Hailin Jin, Matt Hoffman, and Deborah Estrin. 2017. Personalizing Software and Web Services by Integrating Unstructured Application Usage Traces. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee.

[61] Longqi Yang, Cheng-Kang Hsieh, and Deborah Estrin. 2015. Beyond classification: Latent user interests profiling from visual contents analysis. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW'15)*. IEEE, 1410–1416.

[62] Yummly. 2016. Retrieved from http://developer.yummly.com.

[63] Xi Zhang, Jian Cheng, Shuang Qiu, Guibo Zhu, and Hanqing Lu. 2015. Dualds: A dual discriminative rating elicitation framework for cold start recommendation. *Knowledge-Based Systems* 73 (2015), 161–172.

[64] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2014. Facial landmark detection by deep multi-task learning. In *Computer Vision–ECCV 2014*. Springer, 94–108.

[65] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. In *NIPS*. Vol. 16, no. 16, 321–328.

[66] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. 2011. Functional matrix factorizations for cold-start recommendation. In *Special Interest Group on Information Retrieval (SIGIR'11)*.

[67] zipongo. 2015. Personalizing Food Recommendations with Data Science. Retrieved from http://blog.zipongo.com/blog/2015/8/11/personali zing-food-recommendations-with-data-science.

[68] zipongo. 2016. Retrieved from https://zipongo.com/.

Received July 2016; revised March 2017; accepted March 2017