You are what you eat: learning user tastes for rating prediction

Morgan Harvey[1], Bernd Ludwig[2], and David Elsweiler[2]

[1] Faculty of Informatics, University of Lugano, Lugano, Switzerland[1].
[2] Inst. for Info. and Media, Lang. and Culture, University of Regensburg, Germany[2]
morgan@derharvey.de, bernd.ludwig@ur.de, david@elsweiler.co.uk

Abstract. Poor nutrition is one of the major causes of ill-health and death in the western world and is caused by a variety of factors including lack of nutritional understanding and preponderance towards eating convenience foods. We wish to build systems which can recommend nutritious meal plans to users, however a crucial pre-requisite is to be able to recommend recipes that people will like. In this work we investigate key factors contributing to how recipes are rated by analysing the results of a longitudinal study (n=124) in order to understand how best to approach the recommendation problem. We identify a number of important contextual factors which can influence the choice of rating. Based on this analysis, we construct several recipe recommendation models that are able to leverage understanding of user's likes and dislikes in terms of ingredients and combinations of ingredients and in terms of nutritional content. Via experiment over our dataset we are able to show that these models can significantly outperform a number of competitive baselines.

1 Introduction

In the developed world people have the luxury of an abundance of choice with regard to the food they eat. While huge choice offers many advantages, making the decision of what to eat is not always straightforward, is influenced by several personal and social factors [9] and can be complex to the point of being overwhelming [12]. Therefore, many people would benefit from assistance that allows them to strike a balance between a diet that is healthy and will keep them well and one that is appealing and they will want to eat. After all, it is no good providing users with healthy diet plans if they do not cook and eat the recipes therein, but instead choose unhealthy meals which are more appealing to them.

This is a problem for which recommender systems (RS) are ideally suited: If systems can predict recipes that the user would actually *like to eat*, this could be combined within a system modelling expert nutritional knowledge to generate appealing meal recommendations that are also healthy and nutritious. A prerequisite, therefore, is an understanding of the factors that influence the decision of whether a recommended meal will be eaten and prepared or not. In this work we investigate these factors by analysing the results of a long-term user study, using the insights obtained to build new RS which are able to significantly outperform the current state-of-the-art in this field.

2 Related Work

RS provide suggestions, in the form of items, that are predicted to offer utility to the user. Such systems are particularly beneficial in situations where there is an overwhelming choice of alternatives and/or where the user lacks sufficient personal experience, competence or time to evaluate potential options [10]. Correspondingly, recommendations are usually made based on knowledge of the user's needs, preferences, and past behaviour. Many RS only use past ratings in order to predict ratings for previously unseen combinations of user and item. A common approach to generating recommendations is to mimic the natural human behaviour of making decisions based on recommendations from peers. More modern approaches [5] attempt to learn a model of how ratings are generated by breaking the rating process down into a number of components or "biases" which contribute to the final rating. In the case of recommending recipes there are many content-related features that could be used to base predictions on, e.g., cooking time, ingredients, nutritional properties, classification of dish, skills required. The open questions are: which content is useful and how can you best make use of this content in recommendation models?

While food recommendation is not frequently studied, there is a small body of appropriate related work. Early attempts to design automated systems using case-based planning to recommend meals include CHEF [4] and JULIA [7]. Hybrid recommenders have been presented [13] for recommending recipes and systems have been proposed based on grouping of users [14]. More recent efforts try to understand user's tastes, improving recommendations by breaking recipes down into individual ingredients, which has been demonstrated to work well [2, 3]. This work has shown that, in the case of recipes, new approaches to the RS problem are necessary. We hypothesise that the process of rating a recipe is complex and several factors will combine to determine the rating assigned, beyond purely the user's tastes and that these tastes must be carefully modelled. Factors such as how well the preparation steps are described and perhaps the nutritional properties of the dish and the availability of ingredients could have a bearing on the user's opinion of the recommendation [6]. We believe that by building recommender algorithms that incorporate or exploit these kinds of aspects we will be better able to accurately predict ratings. However we also believe that it is important that such factors can be automatically ascertained from ratings data rather than relying on the users themselves. The amount of information expected from users is therefore minimised. Below we describe how data was collected and analysed to understand how content and contextual factors may influence the way a recipe is rated.

3 Data Collection

To collect data we developed a simple food rating system, which selected recipes from a pool of 912 recipes sourced from a popular German recipe web site. While there is quite a strong emphasis on German food (which is beneficial as most

users were German), the web site also contains a large number of recipes from all of the major world cuisines. Users were given a personalised URL and when this was accessed, they were presented with a randomly selected recipe. The system did not attempt to perform any recommendation or try to match recipes to a user's tastes. The user was then asked to provide a rating for the recipe in context - either as a main meal or breakfast for the following day - by clearly stating which meal the user should have in mind when rating, e.g. Please rate this recipe as a breakfast for tomorrow. Recipe meta-data was used to determine which meals should be recommended for which time period. This is important because, in contrast to previous data collection methods, the user is not only rating the recipe with respect to how appealing it is, but also how suitable the recipe is given a specific context. In addition to collecting ratings, the web interface offered the user the chance to explain his rating by clicking appropriate check boxes representing reason. These check boxes were grouped into reasons to do with personal preferences, reasons related to the healthiness of the recipe and reasons related to the preparation of the recipe. The listed explanations were generated through a small user study, whereby 11 users rated recipes and explained their decisions in the context of an interview. The web interface also provided a free-text box for reasons not covered by the checkboxes [3].

After publicising the system on the Internet, through mailing lists and twitter, 124 users from 4 countries provided 4,472 ratings over a period of 9 months. We argue that although this is a relatively small and sparse dataset, it is an improvement on previous recipe ratings data collection methods, which have used mechanical turk, where there are no validity controls and users are incentivised to rate as many recipes a possible as they are being paid [2, 3] and surveys where participants rate large numbers of recipes or ingredients in a single session.

Our dataset also differs from previous work in terms of matrix density. The number of ratings per user is Zipfian (median = 7, mean = 29.93 max = 395 min =1; 18 users have 1, 52 have 10+). Whereas previous food recommender papers report user-ratings densities of 22%-35% [2, 3], our dataset exhibits a more realistic density of 3.95% and a median 3 ratings per recipe (mean = 4.04, max=14, min=2), more in line with collections such as movielens and netflix. Our dataset is, therefore, not only realistic, but also a challenging platform for experimentation as it is both sparse and variant in terms of ratings (sd = 1.43).

4 Exploratory Analysis

To learn about the decision process undertaken when users rate recipes, as well as the factors that influence this process, we statistically analysed the reasons provided by the users when they rated. The most common reasons for negatively rating a recipe were that the recipe contained a particular disliked ingredient, the combination of ingredients did not appeal, or the recipe would take too long to prepare and cook. The most common positive factors included ease or

[3] Screenshot of the interface - http://tinypic.com/r/1zx4p77/4

quickness of preparation, the type of dish or the recipe being novel or interesting. Health related reasons, such as the recipe containing too many calories, the recipe being perceived as unhealthy, or positive factors like the recipe being balanced or easily digestible were clicked less often overall. However, these were clicked very frequently for a particular subset of users; those who ever chose a health reason did so, on average, for 16.3% of the recipes they rated.

We trained a number of linear models to understand how relationships between factors contribute to a final rating. The final model (adj. R^2=0.329) shows that 17 factors were significant. Ingredient factors, such as the presence of particular ingredients or combination of ingredients and whether meat was in the recipe had particularly strong predictive power. Furthermore, the data show that ingredient factors can have both a positive and negative influence on ratings and that the combination of ingredients can be important, neither of which are considered by current models.

Although the health factors did not add significantly to the predictive power of the models, we wanted to understand if they might help predict ratings on a per-user basis. We looked at the correlation between calorie and fat content of recipes and the ratings provided by two groups of users, those who had clicked on a health related factor once or more (Care-about-Health, n= 54, 3130 ratings), and those who never clicked on a health reason (Don't-Care-About Health, n=70, 1342 ratings)[4]. Figure 1 shows clear differences between the rating behaviour exhibited in these groups. There is a strong trend that for the Care-about-Health group, the higher the fat (R^2=0.88, p=0.012) or calorific content (R^2=0.87,p=0.022) of the recipe, the lower the rating. However, this trend is not present in the second group. If anything, there seems to be a slight tendency toward the reverse trend, whereby recipes higher in fat (R^2=0.23,p=0.643) and calories (R^2=0.73, p=0.064) are assigned a higher rating. This analysis suggests that accounting for nutritional factors in recommendation models will allow more accurate predictions to be generated. To summarise, these analyses demonstrate

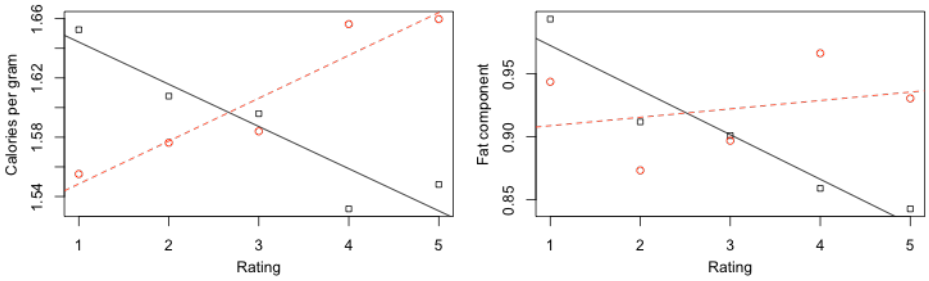

Fig. 1: Influence of Calorific Content and Fat on Ratings

the complexity of rating decision process. Even with 17 significant explanatory variables, the best model is still only able to return an adj. R^2 value of 0.329. Nevertheless, they hint that several factors could be exploited in recipe recommendation algorithms to improve accuracy. In the following section we describe

[4] Nutritional content of recipes was calculated using the system as described in [8]

models that exploit the factors and trends uncovered. As a starting point, we focus on building powerful ingredient based models, as these were shown to be important and have been emphasised in previous work. We then extend these ingredient models to take account of nutritional aspects in terms of fat and energy, which aligns well with our long-term research aims.

5 Recommendation Models

Before describing the new models, we first introduce appropriate notation in Table 1. Note for the purposes of this discussion recipes may also be referred to as "items" and ingredients as "features", these terms will be used interchangeably.

Symbol	Description	Symbol	Description
$d \in \mathbf{d}$	set of items (recipes)	Φ	item feature weights **d** x **f**
$u \in \mathbf{u}$	set of users	$\phi_{d,f}$	weight of feature f in item d
$f \in \mathbf{f}$	set of features (ingredients)	Ψ	user feature weights **u** x **f**
R	ratings matrix **u** x **d**	$\psi_{u,f}$	weight of feature f for user u
b_u	bias due to user u	Ψ^+	matrix of positive user features
b_d	bias due to item (recipe) d	Ψ^-	matrix of negative user features
$r_{u,d}$	rating for item d by user u	IUF_f	inverse user frequency of feature f
IDF_f	inverse item frequency of f		

Table 1: List of notation for recommender models

Ingredients contained in recipes are like words in documents and can be referred to as features. Based on this assumption, we can build an item-feature matrix Φ which can be either binary, indicating the presence or absence of an ingredient in a recipe, or the relative weight of each ingredient in the recipe. The weight of feature f in item d is $\phi_{d,f}$. To compute a similarity between users and items, we can construct a similar feature matrix for users. Such a matrix Ψ (where $\psi_{u,f}$ is the weight of feature f for user u) can be constructed by considering the ingredients contained within the recipes rated by the user. Concretely: $\psi_{u,f} = \sum_{d \in D} \phi_{d,f}\ I\{r_{u,d} > 0\}\ \pi$, where π is an additional weighting factor that may be item, user or feature-dependent (in the un-weighted case this defaults to 1) and $I\{\}$ is the indicator function which is 1 when the condition within the braces - in this case that user u has rated item d - is satisfied.

From our analysis we know that the contextual factors which affect ratings can be both positive and negative and vary in their influence. In the case of ingredients, it was shown that users have a set of ingredients (and combinations thereof) that they like as well as a set of those that they do not like. Previous work has attempted to incorporate this observation into the modelling process by weighting ingredient-features by the rating assigned to the recipes containing that ingredient [3]. In one model, ratings are exploited by assigning weighting to ingredients based on their parent recipe's rating (i.e. π is set to $r_{u,d}$). The problem with this approach is that it implicitly assigns some positive rating to

ingredients which the user dislikes, particularly compared to ingredients which they have not yet rated.

We take a different approach by using two separate user-feature matrices Ψ^+ and Ψ^- containing weighted values for ingredients the users like and those that they do not like. Ψ^+ is derived from recipes to which the user assigned a rating of 4 or 5 and Ψ^- from those that the user assigned a score of 1 or 2. Here we utilise the weighting factor π; for Ψ^+ we can assign a weighting of 1 for those rated 4 and a weighting of 2 for those rated 5, for Ψ^- recipes rated 1 receive a weighting of 2 and those rated 2, a weight of 1. This preserves the idea that a rating of 5 indicates a stronger positive preference than a rating of 4 whereas a rating of 1 should be more strongly negative than a rating of 2.

5.1 Predicting ratings

We now need a metric to determine how similar (or dissimilar) an item and a user are. We use a variation on TF-IDF weighting [11] as this will give high weights to ingredients that are frequently rated positively by the user of interest, but not generally by all users. Similarity between two items can be computed using the cosine similarity metric, resulting in a vector space (VS) model:

$$sim_{VS}(u,d) = \sum_{f \in \mathbf{f}} \frac{(\psi_{u,f}\ IUF_f)(\phi_{d,f}\ IDF_f)}{\sum_{\mathbf{f}} \sqrt{(\psi_{u,f} IUF_f)^2} \sum_{\mathbf{f}} \sqrt{(\phi_{d,f} IDF_f)^2}} \tag{1}$$

When analysing the ratings matrix we noted that it was rather sparse, particularly on a per-user basis with many users having only rated a small number of recipes. This introduces problems for the basic TF-IDF model as a large number of users will have very sparse feature vectors. Our analyses also show that people like types of ingredients and specific combinations of them and therefore performance may be improved by trying to learn which ingredients are similar through their co-occurrence in recipes. This can be achieved implicitly by the use of dimensionality reduction techniques on the feature matrix.

We can therefore apply a Singular Value Decomposition (SVD) to the feature matrices in a similar fashion to previous work in information retrieval [1] where this method has been used successfully to improve accuracy for document retrieval. SVD is commonly used to reduce the amount of noise within matrices and can uncover relationships between variables that are not obvious from the explicit first-order co-occurence data. The reader is referred to [1] for a more detailed treatment of the subject. Given a reduced-dimensionality representation of the original feature matrices, a similarity metric between two items is simply the cosine of the angle between their vectors over the new feature space.

5.2 User and item (recipe) biases

As noted in the related work section, many modern RS estimate ratings based on a number of biases. Two sets of biases which have been shown to have a large impact on the rating ultimately given are dependent on each individual user

and on the item (in this case recipe) being rated. For example, some users may naturally rate items higher than others and some may naturally choose from a lower baseline score. Similarly some items are intrinsically better than others and are therefore likely to be rated higher by all users. By calculating these biases as part of our model, we can effectively remove these eccentricities from the ratings. This gives the ingredient similarity measures the freedom to deal purely with the variations caused by each user's tastes. The bias due to user u is denoted b_u and the bias for recipe/item d is denoted b_d.

These biases can be calculated by means of iterating fixed-point gradient descent optimisation routine based on the training ratings until convergence is observed via the following update rules:

$$\hat{b_u} = b_u - \lambda(eb_d - \alpha b_u) \tag{2}$$

$$\hat{b_d} = b_d - \lambda(eb_u - \alpha b_d) \tag{3}$$

where $\hat{b_u}$ and $\hat{b_d}$ are the updated values for the parameters, λ is a fixed scalar parameter which determines the learning rate of the optimiser, α is a regularisation parameter to prevent over-fitting and e is the error of the following simple model estimate $\hat{r}_{u,d} = \mu + b_u + b_d$ where μ is the mean training rating.

5.3 Including nutritional information

Our analysis indicate that there is a notable split between users who appear to care about the healthiness of a dish and those to whom this factor is perhaps not so important. We know, for example, that those users for whom nutrition is important will, in the mean, rate items with high levels of fat and calories lower than other recipes. This information could be used to introduce an additional bias into the model in order to improve prediction performance. To model this bias, we first split the recipes into "bins" based on their calorific and fat content. Bins were chosen by calculating the q quantiles of the calories and fat respectively and assigning each recipe to its corresponding bin. We separated users into "healthy" and "unhealthy" groups based on their use of the *calories* and *healthy* checkboxes in the training ratings. For each of the two groups a vector of biases was computed for all bins over both the calories and the fat content, where the biases are simply the expected mean-normalised change in the rating for rated items within the bins. These biases are then included as additional explanatory variables in the linear model. Due to the splitting of users it is necessary to calculate two separate models, one for each user group, since the coefficients for both the calories and fat biases will be different for the two distinct groups.

To predict a rating $\hat{r}$ for user u given a recipe (or item) d we can learn a linear weighted model based on the output from the similarity metrics over both positive and negative feature matrices and the biases:

$$\begin{aligned}\hat{r}_{u,d} \; &= \theta_0 + \theta_1 sim^+(u,d) + \theta_2 sim^-(u,d) \\ &+\theta_3 b_c(u,d) + \theta_4 b_f(u,r) + \theta_5(b_u + b_d)\end{aligned}$$

where $b_c(u,d)$ and $b_f(u,d)$ are the predicted calorie and fat biases for user u and recipe d (based on the calorie and fat bins d belongs to). The terms in this

linear equation can actually be seen as the factors that combine to bias the rating in either a positive or negative direction, thus perturbing the rating from some baseline "standard" rating. θ_0 can be seen as approximating a standard or average rating, θ_1 is the factor biasing the rating in a positive direction and θ_2 biases in the opposite direction. θ_3 and θ_4 represent the biases due to nutritional content and θ_5 encodes the influence of the user and item-specific biases. These weights can be optimised using a large number of numerical optimisation procedures including gradient descent, neural networks and generalised linear models. Due to its stability and relative simplicity we use the latter method in this work.

6 Experimental Results

To test the performance of our models for recipe recommendation we must ascertain how well they are able to predict ratings for unknown pairs of users and recipes. To do so we randomly separated our dataset into 5 equal partitions and conducted split-fold testing where for each test 4 of the partitions is used for training the models and the remaining partition is used to test performance. resulting in a total of 3,624 training ratings and 848 test ratings.

The prediction problem is best described by saying that we would like to "fill in" the sparse ratings matrix, extrapolating (or predicting) a rating $\hat{r}_i$ for every possible user-item pair from the limited data available. More practically we wish to define some function or model which will minimise the root mean squared prediction error over the test data $RMSE = \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (r_i - \hat{r}_i)^2}$. The RMSE is commonly used in statistics for measuring the difference between the set of values predicted by a model and the values actually observed from the system being modelled. We also report the Mean Absolute Error (MAE) which is simply the mean absolute difference between the predicted rating and the actual rating, over the whole test set. We report both metrics as they provide different information regarding the performance of predictions: the RMSE penalises large errors much more than small errors while the MAE penalises all errors equally relative to their size.

6.1 Models and Parameters

We compare the performance of our models against 3 baselines from the CF literature, including the state-of-the-art recipe recommendation model:

mean-r naïve baseline, returns the mean rating as an estimate for all u, d pairs.
CF nearest-neighbour method, Pearson correlation coefficient similarity metric.
CB best-performing content-based algorithm by Freyne et al [3].

In this section we evaluate the performance of the following 4 recipe recommendation models as described in Section 5:

VS weighted model with VS similarity measure
VS+n weighted model, VS similarity measure, nutritional biases

VS+n+b weighted model, VS similarity measure, all biases
SVD weighted model, SVD-based similarity measure
SVD+n weighted model, SVD-based similarity measure, nutritional biases
SVD+n+b weighted model, SVD-based similarity measure, all biases

For CF we use a maximum of 10 neighbours ignoring those with low similarity (<0.2). Both SVD models were trained over 100 dimensions. For the +n models q was set to 20 quantiles. The user and item optimiser converges as would be expected, with major gains being made over the earlier iterations and becoming smaller as the optimal values are reached, completely flattening out near the end. This hints that the algorithm has fully converged by this point. The learning rate λ was set to 0.001 and the regularisation parameter α was set to 0.05 as this resulted in the fastest convergence times and best held-out likelihood.

6.2 Average Performance

Table 2: Best results from each model. % indicate improvement over mean baseline. ***** indicates statistically significant improvement over mean, † over CB model

	Prediction error		**Improvement**	
Model	**MAE**	**RMSE**	**MAE**	**RMSE**
mean	1.180	1.383	-	-
CF	1.175	1.379	0.42%	0.28%
CB	1.154	1.347	2.2%	2.6%
VS	1.115 *	1.308 *	5.5%	5.4%
VS+n	1.109 *	1.299 *	6%	6.1%
VS+n+b	1.079 * †	1.269 * †	8.6%	8.2%
SVD	1.095 * †	1.296 * †	7.2%	6.3%
SVD+n	1.086 * †	1.289 * †	8%	6.8%
SVD+n+b	1.072 * †	1.256 * †	9.2%	9.2%

Table 2 shows the average performance figures yielded by the models. Significance is determined based on the p-value returned by a paired Student's-t test. Exact p-values were: SVD-CB = 0.02, SVD+n-CB = 0.011, mean-CB = 0.39. The p-values comparing the mean with all of the models presented in this paper were $\ll 0.01$. The results indicate that all the content-based recommenders are able to outperform both the mean rating and the neighbourhood-based algorithm, which returns particularly poor figures for this dataset. This is likely due to the sparsity of the data making it difficult for the algorithm to find suitable neighbours from which to derive its estimates. Among the content-based methods it is clear that the VS method outperforms the CB method and that the SVD method in turn outperforms VS.

Addition of the nutritional information into the model improves performance for both the VS and SVD variants, however in neither case is this improvement

significant. The addition of the individual user and item biases is, however, significant and increases the performance of both the VS and SVD-based models. In fact, the performance gain is such that the VS model with the biases is even able to beat both of the SVD models without the biases. As would be expected, the performance of the SVD algorithms are somewhat dependent on the number of dimensions. Performance with a small number of dimensions (i.e. 10) is poor, but increases consistently until it reaches an informational saturation point at approximately 100 dimensions, after which performance gain is asymptotic. The performance of all of the trained models increases with the proportion of training data, however it appears that the newer models are better able to exploit the extreme case where 90% of the data is used for training. The errors returned at the other extreme (i.e. where only 50% of the data is kept for training) suggest that the SVD-based model is able to cope better in the case of sparse data than the VS-based one.

6.3 Standard deviation of errors

The RMSE and MAE provide useful information regarding the performance, and more specifically the expected error, of a given prediction algorithm. However, users do not want excessively large errors as this can rapidly destroy their trust in the system and therefore the standard deviation of the errors is also important.

The most variant errors are returned by the CF algorithm followed by the mean, with these returning 0.716 and 0.715 respectively. The content-based recommenders perform better: CB = 0.696, VS = 0.669, SVD = 0.665, VS+n = 0.66 and SVD+n= 0.659. By adding in the user and item biases the standard deviation is further reduced to 0.647 for SVD+n+b and 0.649 for VS+n+b. These results illustrate further that differences in performance suggested by the RMSE and MAE results are likely to make a tangible difference to the accuracy of the recommender. There is little difference between the performance of the 3 baselines, however there is a large step-up in the performance of the models outlined in this work. The fact that the improvements (over the baseline) for the RMSE scores are larger than for MAE also suggests that the models presented in this paper make fewer large errors. As discussed previously, this is advantageous as making large errors can have serious implications with regard to the trust of the user in the RS. The much lower standard deviations for the models that incorporate the user and item biases also illustrate how much extra prediction power and flexibility these are able to provide.

7 Discussion of Results and Conclusion

In this paper we have investigated the decisional process involved in rating recommended recipes. We described a large naturalistic data collection method with 124 users in which recipes were rated in context over a period of 9 months. This resulted in a realistic dataset for testing RS for this specific problem which

approximates well the kind of data that would be generated in a real recipe recommendation system, especially compared to recipe datasets used in the past. Analyses of the dataset underlined the complexity of the recipe rating process with 17 factors having a significant influence on the rating in the best linear model. Yet, this model is only able to explain about a third of the variance in the rating. However, based on insights obtained from analyses performed, we developed new models and showed empirically that these models offer performance improvement over strong baselines.

The results justify choices made in the modelling process; the new models offer improved performance both in terms of reducing error and the variance of the error. The results show that the ingredients contained within recipes are important and that this data can be better exploited by using models that account for positive and negative weighting and by applying dimensionality reduction techniques. Furthermore it is clear from the results obtained that training separate bias parameters for each individual user and item is extremely beneficial, particularly given the low cost in terms of increased model estimation time.

Including nutritional information in the model was also shown to be beneficial. The models incorporating calorie and fat data offered improved performance, although the differences were not significant. It is our intuition that as our dataset grows, the results of models exploiting nutritional information will improve. It should also be noted that our current method of incorporating recipe nutritional information in the rating prediction is quite simple and could certainly be improved. More sophisticated models might, for example, use a continuous function to estimate weights for calories or fat values rather than simply using bins. Moreover, future models may learn weights on a per user basis rather than relying on pre-defined groups as we do now.

The presented work represents a single component in a much larger project aimed at building RS that can promote healthier dietary choices. Our short term goals include continuing the work here to build models that better predict user food preferences using the ideas suggested above or, for example, by incorporating other content factors. We are continuing to collect data and hope to investigate how performance of models change as the collection size increases. For example, will the CF approaches eventually match content approaches when the collection achieves a certain density? We acknowledge that, in contrast to our long-term goals, the nutrition-aware models could be improving performance by offering unhealthy choices to users who prefer such recipes. It would be interesting to look at how these models influence the error for the two user groups. Further analysis might also provide an understanding of where nutritional content plays a role i.e. at what level of fat content do users start to rate differently? Our initial analyses in Section 4 suggest that there is also scope for further performance improvement by developing models that take other content-related factors into account.

In the longer term we plan to move beyond recommending recipes in isolation to recommending full dietary plans. This would involve recommending sequences of recipes under a number of constraints such as the daily recommended intake

suggested by the WHO, and user activity patterns. Achieving this will present several algorithmic and usability challenges and will necessitate the development of more sophisticated models. We are also interested in understanding how using this kind of RS can influence user behaviour and knowledge of nutritional principles. Will users learn about nutrition over time from the suggestions made or will they instead simply rely on the system to provide them with meal plans without truly understanding what they should be eating to keep healthy?

It is worth noting that the models presented in this paper could also be used for other RS problems where contextual data is available for items that could be used as features for the similarity matrices. For example, the models could be adapted to recommend movies by using features such as directors, actors and genres instead of ingredients.

References

1. S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R Harshman. Indexing by lsa. In *J. of the Am. Soc. of Inf. Sci.*, volume 41(6), pages 391–407, 1990.
2. J. Freyne and S. Berkovsky. Intelligent food planning: personalized recipe recommendation. In *15th Int. Conf. on Intelligent User Interfaces*, IUI '10, pages 321–324, New York, NY, USA, 2010. ACM.
3. J. Freyne, S. Berkovsky, and G. Smith. Recipe recommendation: accuracy and reasoning. In *Proc. UMAP*, pages 99–110. Springer-Verlag, 2011.
4. K. Hammond. Chef: A model of case-based planning. In *Proceedings of the National Conference on AI*, 1986.
5. M. Harvey, M. J. Carman, I. Ruthven, and F. Crestani. Bayesian latent variable models for collaborative item rating prediction. In *Proc. CIKM'11*, pages 699–708. ACM, 2011.
6. M. Harvey, B. Ludwig, and D. Elsweiler. Learning user tastes: a first step to generating healthy meal plans? In *ACM RecSys 2012 LifeStyle Workshop*, 2012.
7. T. Hinrichs. Strategies for adaptation and recovery in a design problem solver. In *Proceedings of the Workshop on Case-Based Reasoning*, 1989.
8. M. Mueller, M. Harvey, D. Elsweiler, and S. Mika. Ingredient matching to determine the nutr. properties of internet-sourced recipes. In *Pervasive Health*, 2012.
9. M. Nestle, R. Wing, L. Birch, L. DiSogra, A. Drewnowski, S. Middleton, M. Sigman-Grant, J. Sobal, M. Winston, and C. Economos. Behavioral and social influences on food choice. *Nutrition Reviews*, 56(5):50–64, 1998.
10. F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, editors. *Rec. Systems Handbook*. Springer, 2011.
11. G. Salton and C. Buckley. Weighting approaches in automatic text retrieval. *IP and M*, 24(5):513–523, 1988.
12. B. Scheibehenne, R. Greifeneder, and P. M. Todd. Can there ever be too many options? a meta-analytic review of choice overload. *J. of Consumer Rsrch.*, 37:409–425, 2010.
13. J. Sobecki, E. Babiak, and M. Słanina. Application of hybrid recommendation in web-based cooking assistant. In *Knowledge-Based Intelligent Info. and Engineering Sys.*, pages 797–804. Springer, 2006.
14. M. Svensson, J. Laaksolahti, K. Höök, and A. Waern. A recipe based on-line food store. In *5th Int. Conf. on Intelligent User Interfaces*, IUI '00, pages 260–263, New York, NY, USA, 2000. ACM.