Rating Bias and Preference Acquisition

JILL FREYNE, CSIRO, ICT Centre
SHLOMO BERKOVSKY, NICTA and CSIRO, ICT Centre
GREGORY SMITH, CSIRO, ICT Centre

Personalized systems and recommender systems exploit implicitly and explicitly provided user information to address the needs and requirements of those using their services. User preference information, often in the form of interaction logs and ratings data, is used to identify similar users, whose opinions are leveraged to inform recommendations or to filter information. In this work we explore a different dimension of information trends in user bias and reasoning learned from ratings provided by users to a recommender system. Our work examines the characteristics of a dataset of 100,000 user ratings on a corpus of recipes, which illustrates stable user bias towards certain features of the recipes (cuisine type, key ingredient, and complexity). We exploit this knowledge to design and evaluate a personalized rating acquisition tool based on active learning, which leverages user biases in order to obtain ratings bearing high-value information and to reduce prediction errors with new users.

Categories and Subject Descriptors: H.1.2 [**Models and Principles**]: User/Machine Systems

General Terms: Design, Algorithms, Human Factors

Additional Key Words and Phrases: Decision support, reasoning, collaborative filtering, content-based, machine learning, personalization

ACM Reference Format:
Freyne, J., Berkovsky, S., and Smith, G. 2013. Rating bias and preference acquisition. *ACM Trans. Interact. Intell. Syst.* 3, 3, Article 19 (October 2013), 21 pages.
DOI:http://dx.doi.org/10.1145/2499673

1. INTRODUCTION

The success of a Web or mobile application is often dependent on early user satisfaction, service expectation meeting, and general usability. Adaptive services, such as recommender and personalized systems, have the added pressure of intelligent service performance, which is particularly challenging when user information is scarce or nonexistent. Practicality demands that functionality and performance must impress users early, to secure loyalty and retain customers, but often this may be difficult due to sparse user data [Rashid et al. 2002]. Thus, much work has been done on the "cold start problem", with some systems relying on domain dependent algorithms that convert item ratings into domain preferences [Berkovsky et al. 2009]. Adaptive learning systems [Rubens et al. 2011] are a typical example of this type of system. These learners aim to exploit trends in a dataset to maximise both the volume of information gathered from a user by considering the user's ability to provide ratings for requested

This work is funded by the CSIRO Food and Nutritional Sciences.
Authors' addresses: J. Freyne, Information Engineering Laboratory, ICT Centre, CSIRO; email: jill.freyne@csiro.au; S. Berkovsky, Information Engineering Laboratory, ICT Centre, CSIRO and Networks Research Group, NICTA; email: shlomo.berkovsky@csiro.au; G. Smith, Intelligent Sensing and Systems Laboratory, ICT Centre, CSIRO; email: gregory.smith@csiro.au.

DOI:http://dx.doi.org/10.1145/2499673

items, and the value of the information provided by examining the rating patterns of the community to identify controversial items or those with high rating diversity.

Our work combines these concepts and takes them beyond the state-of-the-art works in the area. While focussing on a content based approach to recommendations, we argue that even scarce rating information provides insight into a user's biases toward certain characteristics of the items being rated. We propose that there is additional value hidden in user ratings, which reaches beyond a user's preference for an item and the features of the item, and tells us about bias toward some characteristics of the items, that is the human decision making process. Our work, applied to the domain of food and recipes, shows that we can ascertain how important certain domain features, for instance, ingredients, cuisine, or a cooking complexity, are to a user rating recipes. For example, we show that ratings of some users are reflective of their reasoning behaviour when the cuisine type of the rated recipes is examined, but for others the ratings may be reflective of reasoning when the ingredients, cuisine type, or complexity are examined. Knowing what is important to the user and whether they are deliberate in their rating provision, offers many opportunities for adaptive systems to enhance their services. We illustrate a practical use of this knowledge in the design of an active learning algorithm that analyses the input of users, determines how they are reasoning, and responds by suggesting to rate items, which contribute high-value information to the system.

The contributions of this work are twofold. First, we provide an in-depth analysis of the bias or reasoning trends identified in our dataset of recipe ratings gathered from the users of Amazon's Mechanical Turk. We examine the stability, predictability, and accuracy of these trends applied for prediction generation. We detail the groups of reasoners that our data points to, groups of users, who provide ratings on a small number of domain characteristics, and those, whose biases are rooted in complex combinations of features. Second, we detail the design and evaluation of a personalized active learning tool. This tool is used for new users of a system, to influence the items that they are asked to rate according to their observed biases and the repository of items in question. The tool aims to gather a rich data from a small number of ratings, to generate accurate predictions for the user.

The article is structured as follows. Section 2 positions this work in relation to other works in the field of human decisions, meal recommendation strategies, and active learners in the recommender space. Section 3 provides an overview of the analysis undertaken and the subsequent discovery of trends in user reasoning. Section 4 details a personalized active learning tool, which seeks to exploit this information in the rating acquisition process. We conclude with a thorough discussion of our findings and implications for other domains and contexts.

2. RELATED WORK

The area of content based recommender systems is well studied and documented [Lops et al. 2011]. The primary goal of the approaches is to identify items or content that users are likely to be interested in, based on the characteristics of items that they have been interested in previously. This process involves matching the attributes stored in a user profile to the attributes of the candidate items for recommendation. In some cases, this involves mining the content of previously viewed web pages, translating rating information to classifications or features of the rated items, or asking users explicitly for interest areas or indicators. The creation of a user profile in a content based recommender is the job of the profile learner. Typically, a supervised machine learning algorithm is employed to judge positive and negative feedback from users in order to ascertain their preferences in relation to a new or previously unrated item [Pazzani and Billsus 2007]. A drawback of using content based algorithms is that in domains,

where there is ample information on which to build the user and item attribute profiles, additional domain and expert knowledge is often required by the profile learner. The work presented here is unusual in that our profile learner relies only on the available features of the items rather than on additional domain knowledge.

It is well accepted that in most adaptive systems the opportunities to gather input from users are scarce and that users are typically reluctant to provide high volumes of input. It is, thus, a priority to design interfaces that seek out feedback of high quality or value when appropriate, and develop algorithms that accurately process this feedback. Much research has been done into the impact of context on user feedback [Adomavicius and Tuzhilin 2011; Adomavicius et al. 2011; Baltrunas et al. 2012]. It has been shown that user feedback is strongly affected by contextual parameters. Some of them, for instance, time and weather, are extrinsic to the user; some are intrinsic, for instance, mood; while others can be categorised as social parameters, for instance, presence of other users. An important factor that was discovered to affect user feedback is the exact way the questions are asked. To name a few, these factors include the ordering/grouping of items [Amatriain et al. 2009], rating scale [Kuflik et al. 2012], and the availability of explanations [Tintarev and Masthoff 2011]. Thus, factoring out the impact of those factors and interpreting the captured feedback in order to create an accurate and up-to-date user profile is a challenging task.

More insights into interpreting user feedback and understanding their preferences when interacting with computer systems come from the decision making research [de Gemmis et al. 2012; Jameson 2011, 2012]. Specifically, Jameson identifies seven influential aspects pertaining to preferential choice and decision making [Jameson 2012]: (1) goal/value, the goal driving the decision making process or the outcome that can be obtained by making the choice; (2) situation, presentation of the scenario/setting and the available options; (3) consequences, future implications of the possible options; (4) temporal dimension, short vs. long term benefit and impact of the decisions; (5) reuse, tendency to select options similar to the previously selected ones; (6) social, examples, norms, and expectations set by others, and (7) learning, new experience that can be obtained through the available options. Although not all of these factors are applicable to the feedback provision scenario, some of them can underpin user reasoning processes when rating items. In our discussion we will address these factors.

An alternative approach to capturing informative user feedback comes from the area of Active Learning, where the profile learners that model the users in recommender systems adapt to the domain and community by identifying high-value items which should be rated by users so that informative user models can be constructed [Chen and Pu 2004; Elahi et al. 2011; Golbandi et al. 2010; Rashid et al. 2008; Rubens et al. 2011]. These learners rank items using various parameters, such the likelihood of a user providing a rating is maximised (thus, maximising the number of ratings in a profile). Other approaches, in particular those exploiting neighbourhood formation in their recommendation process, often focus on items of high entropy (diverse ratings from users, but lower chances of users being able to rate the items) in order to maximise the differentiators in a user's profile.

In the area of food recommendations, wide and varied approaches to making recommendations have been undertaken. Early efforts resulted in systems, such as Chef [Hammond 1986] and Julia [Hinrichs 1989], which rely heavily on domain knowledge for recommendations. More recently, works concentrating on social navigation, ingredient representation and recipe modeling have come to the fore. A recipe recommender system based on user browsing patterns is presented by Svensson et al. [2001]. They use social navigation techniques and apply collaborative filtering to predict ratings. While users reported liking the system, formal analysis of its predictive power is not reported.

Zhang et al. [2008] also make use of an ingredient representation but, in contrast, distinguish between three levels of ingredient importance, which are manually assigned. Using this mechanism, ingredients that are considered to be more important, have the largest contribution to the similarity score. Once again, a level of domain expertise is required for this process. We would argue that the importance of an ingredient in a recipe is likely to be user dependent rather than generic. Van Pinxteren et al. do take a user-centred approach to recipe modeling, rather than make a priori assumptions about the characteristics that determine the perceived similarity, such as ingredients or directions [van Pinxteren et al. 2011]. They derive a measure, which models the perceived similarity between recipes by identifying and extracting important features from the recipe text. Based on these features, a weighted similarity measure between recipes is determined.

Common to all of these approaches is a requirement for domain knowledge and input, a factor which we would ideally avoid in order for our meal planning technique to be applied to large repositories of recipes and user contributed content, which have become more and more common since the arrival of the Social Web. Thus, our algorithms focus on exploiting the information that is already contained in a recipe or that can be generated from the characteristics of the recipe content (complexity, the number of ingredients/steps, etc).

3. UNCOVERING PATTERNS IN USER PREFERENCE DATA

3.1. Identifying Predictive Features

Our previous work focussed on the exploration algorithms for accurate recommendations in the food domain [Freyne and Berkovsky 2010a, 2010b; Freyne et al. 2011a, 2011b]. We have analysed and compared several algorithms, such as collaborative filtering, content based filtering, and a variety of hybridizations, in an effort to understand the strengths of these techniques when applied for recipe recommendation generation. We have assessed these approaches in terms of their prediction accuracy, classification accuracy, and coverage [Herlocker et al. 2004].

An analysis of the use of the M5P algorithm [Quinlan 1992; Wang and Witten 1996] that generates a pruned logistical decision tree based on the features of a recipe and the scores provided, has led us to investigate potential reasoning patterns of users providing our data. The M5P algorithm generates a binary tree classifier, where where each leaf predicts a numeric quantity using linear regression, and then computes the scores based on the recipe content and the metadata associated with a recipe. Each data instance is a set of features $\{a_1, \ldots, a_{N+1}\}$, where each feature may be numeric or nominal, but a_{N+1} is the class label and must be numeric. M5P exploits a feature selection algorithm to prescribe the features on which the algorithm runs by identifying the best features for a given user profile. We employed a correlation-based feature selection algorithm (CFS) to compute a heuristic measure of the *merit* of the selected features from pair-wise feature correlations. The merit is quantified by

$$M_S = \frac{k\overline{r_{cS}}}{\sqrt{k + k(k-1)\overline{r_S}}}, \tag{1}$$

where k is the number of features in the selected set S, $\overline{r_{cS}}$ is the mean feature-class correlation over class c and set of features S, and $\overline{r_S}$ is the average feature-feature inter-correlation over S. The correlation is calculated using symmetrical uncertainty:

$$u(X,Y) = 2\left[\frac{g(X,Y)}{h(Y) + h(X)}\right], \tag{2}$$

Fig. 1. Ratings solicitation interface displayed to Mechanical Turk users.

where h is entropy of a feature and g is information gain of a class given a feature [Hall 1999]. We examined the use of other feature selection algorithms that produced similar patterns, but for reasons of brevity we supply the details of only the CFS algorithm in this article.

The selection of a feature as a predictor depends on the extent to which it predicts classes in areas of the instance space not yet predicted by other features. The output of this process is a set of predictive features and the merit associated with each grouping. Thus, for each user and their ratings, we can ascertain the features that are predictive of their provided ratings. The CFS algorithm identifies the features or characteristics of recipes, on which the M5P algorithm is basing its predictions from our set of possible features (ingredients, broad category, cuisine general, cuisine specific, number of ingredients, and number of steps). It identifies input patterns across the various values of features for each individual rater in our dataset, the number of and the set of features on which patterns of ratings can be found. We hypothesise that these patterns reflect user decision processes involved in providing ratings, in this case on recipes, as detailed in the next section.

3.2. Dataset

When investigating the development of a food recommender, we gathered a corpus of recipes and solicited ratings on these recipes, to allow us to find suitable algorithms to generate recommendations. The recipe corpus consisted of 343 recipes obtained from the CSIRO Total Wellbeing Diet books [Noakes and Clifton 2005, 2006] and from the meal planning website Mealopedia.[1] Online surveys, each containing 35 randomly selected recipes, were posted to Mechanical Turk, Amazon's online task facilitator.[2] Responses for each of the 35 recipes displayed were required and users could answer as many of the published surveys as they wished. Each question asked users to report how much a recipe appealed to them on a 5-Likert scale, spanning from "not at all" to "a lot," as shown in Figure 1. Sometimes, a user might not be familiar with an ingredient or method of cooking included in a recipe, but we assume that in most cases they could provide an answer based on their background knowledge.

[1]http://www.mealopedia.com
[2]http://www.mturk.com

Table I. Rating Spread

	not at all	not really	neutral	a little	a lot
count	15191	14425	19840	25593	26508
percentage	15%	14%	20%	25%	26%

Table II. Metadata Features and Values

General Cuisine	Specific Cuisine	Category
African, American, Asian, European, International, Oceania	African, Australian, Chinese, Eastern European, French, German, Greek, Indian, International, Italian, Japanese, Mexican, Middle Eastern, South East Asian, Southern, Spanish, UK&Ireland	beef, pork, lamb, chicken, veal, fish, vegetables, fruit

Much discussion has been seen in the literature on the appropriateness, quality and accuracy of the use of crowd-sourced data through Mechanical Turk [Buhrmester et al. 2011; Kittur et al. 2008; Paolacci et al. 2010]. Many strategies for quality control have been experimented with, including using minimum task duration thresholds, comparison of answers to expert users, etc. In our work, there are no correct or wrong answers, as we are seeking Turkers' feedback and opinion. Thus, it is inappropriate to compare the ratings given by Turkers to domain experts or apply similar strategies. We are nonetheless interested in quality control, as we aim to bootstrap future systems with the data gathered as well as to run our analyses. Our first check verified that the responses to a series of questions relating to the size of the household in which the Turker lived is consistent. It compared the total number of residents stated to the sum of the residents in certain age brackets. We further employed two strategies for the exclusion of data from the rating dataset. We set a minimum task duration threshold, a period of time deemed suitable to rate, with thought, the 35 recipes required per task. If a user completed the survey in a time under this threshold, their data was excluded. If all the ratings from a user had the same score, their data was excluded.

In total, we gathered 101,557 ratings of 917 users, such that the density of the obtained ratings matrix was about 33%. The distribution of ratings across the rating scale is presented in Table I. On average, each recipe was made up of 9.52 ingredients (stdev = 2.63) and the average number of recipes that each ingredient was found in was 8.03 (stdev = 19.8). On average, each user rated 109 recipes (stdev = 81.9), with the minimum number of user ratings being 35 and the maximum being 336.

Each recipe has a fixed structure that includes a *title*, *ingredient list*, *instructions*, and *image* that was shown to the raters. We automatically extracted two additional indicators of recipe complexity: the *number of ingredients* and the *number of steps* required to complete the recipe. In addition, we manually annotated each recipe with simple domain knowledge in the form of a *general cuisine type*, a *specific cuisine type*, and a *broad category*, containing options traditionally used to classify food. The options for cuisine types and categories are shown in Table II.

3.3. Predictive Feature Identification on Recipe Ratings

We began our analysis by contrasting the performance of two implementations of the M5P algorithm on our dataset. In the first instance, we used the full dataset to generate a model using the M5P algorithm, with no feature selection algorithm in place. This resulted in a generic, one-size-fits-all model for our data. Running a 10-fold cross validation analysis with 90% training and 10% test set yielded an Mean Average Error (MAE) of 1.175. In the second instance, we generated an individual M5P model

Table III. Distribution of Predictors

	1 predictor	2 predictors	3 predictors	4 predictors
profiles	172	327	187	147
% of total	20.6%	39.2%	22.4%	17.7%

Table IV. Combinations and Dominance of Features when Two Predictive Features Exist

Predictive features (feat1,feat2)	% of profiles applicable	most predictive feat1	most predictive feat2
(broad category, general cuisine)	48.62%	57.2%	42.8%
(broad category, specific cuisine)	37.31%	81.9%	18.1%
(broad category, number of ingredients)	10.40%	26.4%	74.6%
other	5.37%		

for each user using only their ratings, and then used the CFS feature selection algorithm detailed above. Once again using a 10-fold validation, we obtained a lower MAE of 0.977, showing that the individual models that exploit feature selection are more accurate than a single global model.

We analyzed the set of predictive features selected for each user and noted significant differences in the number and variation of the predictive features identified by the algorithm. 20.6% of users have one predictive feature, 39.2% have two, 22.4% have three and 17.7% have four predictive features, as seen in Table III. We hypothesise that the different number of predictors reflects different levels of reasoning and decision making employed by users when rating recipes. Since we acquired this data through Amazon's Mechanical Turk, which rewards users financially for their ratings and the ratings do not assist users in any way, we hypothesise that users could base their opinions on various aspects of the recipes and that some users were more thorough with the provided answers than others.

As mentioned, 20.6% of users have one predictive feature. For 93% of users in this group, the feature identified was the *broad category* feature, that is, the presence of a certain ingredient. We assume that users in this group assigned ratings to recipes based primarily on the main ingredient of the recipe. Simple rational following this reasoning is: "I like chicken recipes, I dislike fish recipes, and I love beef recipes."

39.2% of users have two predictive features selected and, we assume, were reasoning on two levels. In 96% of these profiles, the *broad category* feature was again selected; this time in conjunction with an additional feature. The additional feature selected was the *general cuisine* in 48.6% of cases, the *specific cuisine* in 37.3% of cases, and the *number of ingredients* in 10.4% of cases. Table IV shows how this breaks down for the various combinations of features. The dominance of the *broad category* feature varied, depending on its coupling with other features. For example, when coupled with *general cuisine*, the *broad category* feature was most predictive in 57.2% of cases. So, with respect to the *broad category* and *general cuisine* features, 57.2% of users were rationalising according to statements like "I like beef and I love it when its included in a Chinese style dish" and 42.8% of according to "I love Chinese dishes, especially ones which contain beef." When the *specific cuisine* feature was a predictor in conjunction with the *broad category*, in 81.9% of cases the *broad category* was the most predictive feature and only in 18.1% of cases the *specific cuisine* feature was most predictive. The opposite is the case when the *number of ingredients* feature was present. It was the dominant feature in 74.6% of cases, while the *broad category* was most predictive in 26.4% of cases.

Table V. Combinations and Dominance of Features when Three Predictive Features Exist

Predictive features (feat1, feat2, feat3)	% of profiles applicable
(number of ingredients, general cuisine, specific cuisine)	43.28%
(number of ingredients, specific cuisine, broad category)	20.90%
(number of ingredients, general cuisine, broad category)	18.51%
(general cuisine, specific cuisine, broad category)	11.94%
other	5.37%

Table VI. Combinations and Dominance of Features when Four Predictive Features Exist

Predictive features (feat1, feat2, feat3, feat4)	% of profiles applicable
(number of ingredients, number of steps, general cuisine, specific cuisine)	3.4%
(number of ingredients, number of steps, specific cuisine, broad category)	1.4%
(number of ingredients, number of steps, specific cuisine, broad category)	4.0%
(number of ingredients, general cuisine, specific cuisine, broad category)	91.2%

22.4% of users have three predictive features selected. When users were reasoning on three features, the *broad category* was not predictive in 43.3% of cases. This suggests that when users applied complex reasoning processes to provide well-thought ratings or when their tastes were more refined, their focus was on the fine grained details of cuisine type and cooking complexity, rather than simply on the main ingredient of the recipe. These users were likely to reason along the lines of "I like Asian dishes, in particular Thai dishes, but only ones with a small number of ingredients." Table V shows the break down of the three predictive features.

Table VI shows the break down of the 17.7% of users who were found to have four predictive features. In over 90% of cases, this was a combination of the following features: *number of ingredients*, *general cuisine*, *specific cuisine*, and *category*. The *number of ingredients* was the most predictive feature for 68% of users in this category; for a further 20.5% of users the *category* was the most predictive; for 10.2% the *general cuisine* was the most predictive, while only 1.4% of users favoured the *specific cuisine*.

We examined whether the number of selected features was related to the size of a user profile, that is, number of ratings provided. We calculated the correlation between the density of a user's ratings vector and the number of selected features. The correlation coefficient was –0.031, showing no correlation between the number of recipe ratings provided and the number of predictive features selected. Thus, the varying features were primarily based on the distribution of ratings across recipes with common characteristics.

Our analysis indicates that users answering our recipe rating survey have consistent patterns with respect to the recipe characteristics. For most users, the broad category (or the core ingredient of a recipe) affects the ratings heavily, whether alone or coupled with additional characteristics, such as the type of cuisine or the number of ingredients.

3.3.1. Stability of Feature Selection. This knowledge and awareness could impact many recommendation processes, for instance, rating solicitation, recommender generation, and confidence in user input. However, before we progress with enhancements based on this understanding, we must examine the stability and variability of the discovered trends and verify the accuracy of the selection algorithms. We have previously ruled out the correlation between the number of ratings in the user profile and the number of features on which users are deemed to reason. However, we must consider that the features selection process may produce varying results when presented with various subsections of user profiles. Figure 2 shows the variability of the selected set of features

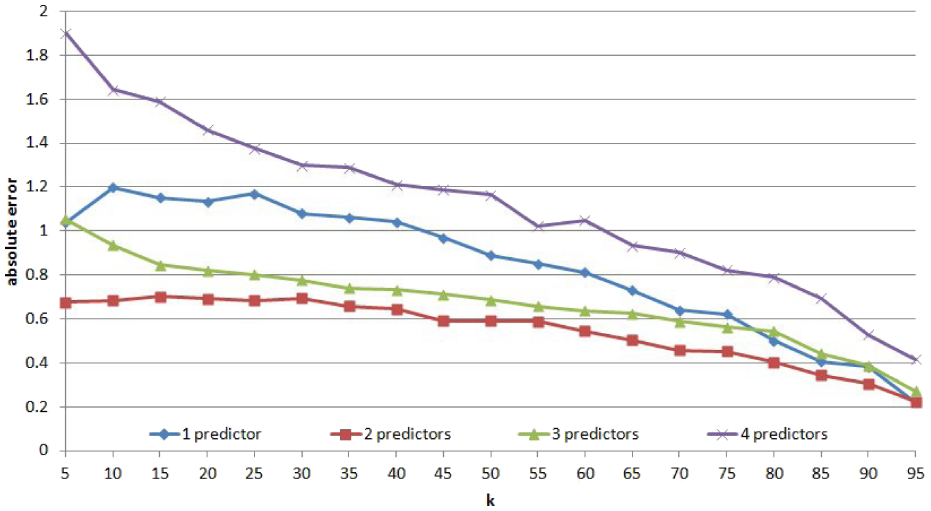

Fig. 2. Predictor stability over time.

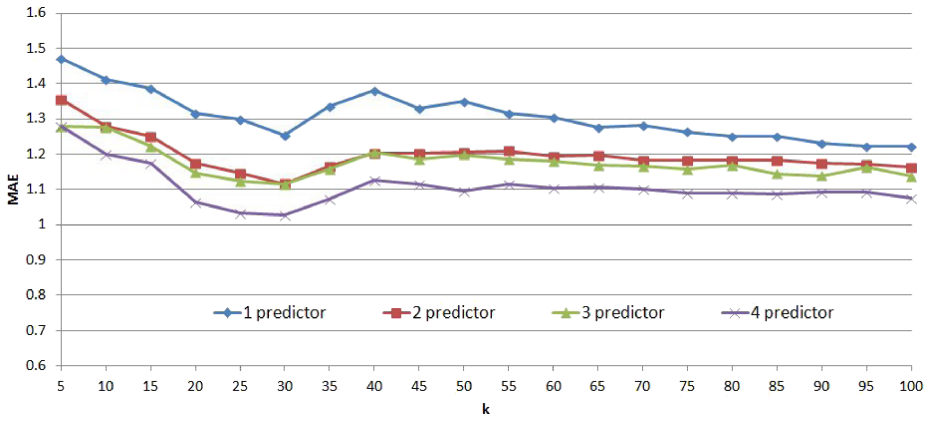

Fig. 3. MAE of predictions made using feature selection at various k.

for the above groups of users (1, 2, 3, or 4 features selected), when subsets of ratings in the profiles were used by the CFS algorithm. Only users having more than 100 rated items in their profiles were considered in this analysis, and for these users the first 100 ratings were considered.

We increase the number of ratings in the profile, k, from 5 to 100 in randomly selected increments of 5 ratings. For each k, we carry out the feature selection process and compare the number of selected features to the number of features selected when 100 ratings in the profile are considered. We repeat this process 10 times and report on the average error between the two. We compute the error separately for groups of users reasoning on 1, 2, 3, and 4 features. Figure 2 shows the error in identifying the correct number of predictive features in each group, for various values of k.

The highest error is obtained for users reasoning on 4 features. We observe an error rate of 1.9 for $k = 5$, followed by a steady decline. The same trend is seen for users reasoning on 3 features, although the error at $k = 5$ is half that of the previous group.

This curve levels off at 0.8 when $k = 25$. A consistent error curve is observed for users reasoning on 2 features, showing that the feature selection is accurate even when a small number of ratings is available. In contrast to the emerging trend, the error rates are high for users reasoning on 1 feature. The error hovers around 1 until $k = 40$ and then steadily decreases. Note that when a user is reasoning on 1 feature, the inaccuracies leading to errors can only be overestimations (i.e., CFS selects more than 1 feature), whereas in other cases it could over- or underpredict (when 2 features are predictive, the system could select 1, 3, or 4 predictors yielding high errors). Thus, the feature selection is mostly predicting that the users are reasoning on two features rather than one for $k < 40$. Similarly, the inaccuracies observed for users reasoning on 4 features can only be underestimations.

Figure 3 shows the MAE of predictions made using the selected features for user profiles of different sizes. For each value of k, feature selection was completed on 90% of the user profiles and the selected features were used to predict the remaining 10% of ratings. 10 runs of each were carried out and the average MAE across users in each group is reported. Note that a similar MAE is obtained for users reasoning on two and three features when $k > 5$. However, there is a pronounced difference in the accuracy of predictions for users reasoning on one feature and four features. We will focus on the analysis of this difference.

These groups had similar errors in the number of predictive features (see Figure 2), but that error has affected the accuracy of the predictions in different ways. The error in the number of selected features across users reasoning on one feature at $k = 10$ was around 1.2. This error was always positive and the number of selected features was overpredicted. Similarly, at $k = 10$ the average error in the number of selected features for users reasoning on four features was around 1.6, and this was always negative, such that the number of selected features was underpredicted. In the overestimated cases, noise was added and irrelevant features were selected, whereas in the underestimated cases some relevant features were not selected.

We examined the changes of merit scores when additional noisy data was added and when some information was missing. The analysis shows a 10% reduction in merit score when an additional feature was selected. Thus, the correlation between the selected features and the ratings is 10% lower. However, missing information has a weaker effect. In this case, the information loss associated with one missing feature is only 2% and with two missing features it is 4%. Thus, it appears better to underestimate the number of predictors rather than overestimate them. Hence, the MAE scores obtained for users reasoning on 4 predictive features are lower than those obtained for users reasoning on one predictive feature.

The accuracy of the predictions when the approach in identifying and applying the correct features for the different groups of users, will impact on overall system performance according to the user coverage of each group, that is, the percentage of users who have 1, 2, 3, and 4 predictive features. In this case we have the lowest performance for those with only 1 predictor (21% of users), followed more positively by those with 2 and 3 predictors (41.6% of users) with the best accuracy for those with 4 predictors (17%). It seems almost logical if you consider that those for which we have information on more features, we can serve best in terms of accuracy that those who appear to have simple tastes, or those who put little effort into their ratings decisions.

3.3.2. Accuracy of Feature Selection. The number of correct features is only a portion of the problem, while the accuracy of the features selected is also important. To this end, we examined the precision, recall, and F1 of the feature selection using different profile sizes. *Precision* measures the ratio of selected items among all the items relevant to the user. *Recall* measures the ratio of relevant items which were selected. The *F-measure*

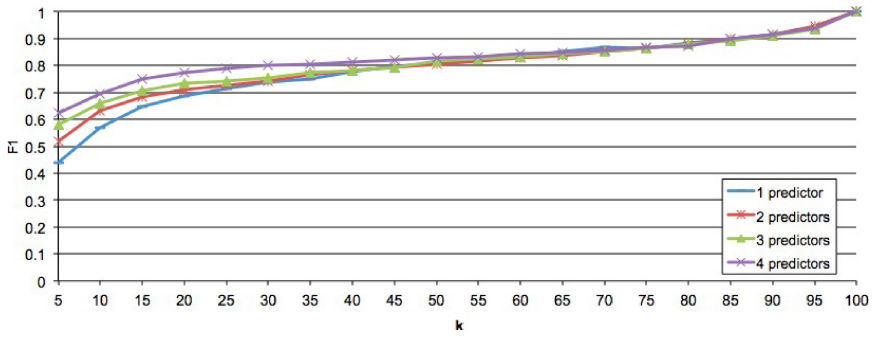

Fig. 4. F1 of attribute predictors at various k.

metric represents their harmonic mean assigning them equal weights, as shown in Equations (3)–(5).

$$precision = \frac{N_{rs}}{N_s} \tag{3}$$

$$recall = \frac{N_{rs}}{N_r} \tag{4}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}. \tag{5}$$

We start with a general performance indicator for each group and report the F1 scores in Figure 4. We note a difference across the groups for $k < 50$, but equivalent performance for $k > 50$. Focusing on the smaller profiles with low k, we note an inverse relationship between the F1 scores and the number of predictive features in the groups. The highest F1 score is obtained for users reasoning on 4 features followed by the scores for those reasoning on 3 and 2 features, and finally, those reasoning on 1 feature. This shows that overall, that the CFS algorithm is more accurate when identifying a high number of reasoning features.

Investigating further, we calculated the precision and recall scores for each group, as presented in Figures 5–8. Again, we observed different patterns for different groups of users. Starting with the performance of CFS for users reasoning on 1 feature, recall of 0.9 is obtained with $k = 20$ ratings and 1 with $k = 50$, while precision increases steadily until reaching 1 only for 100 ratings. Thus, for users with simple patterns of reasoning, the algorithm correctly identifies the predictive feature, although it appears to also identify another incorrect feature, resulting in the lower precision. For users with 2 predictive features, we note an improvement in precision as fewer incorrect features are identified, but also a decrease in recall. The correct 2 features are identified less often than in the previous group. We note precision and recall values close to 0.5 when $k = 5$ ratings are provided, with a steady increase in both as k increases, reaching recall of 0.8 with 35 ratings and precision of 0.8 with 60 ratings.

For users reasoning on 3 features, we note that the level of precision observed outperforms that of recall for the first time. We observe high and steadily growing precision scores of over 0.7 for very low k, whereas recall starts at 0.5. This is even more exaggerated in the last precision/recall graph of users reasoning on 4 features, as shown Figure 8. We observe very high precision scores as the algorithm correctly identifies

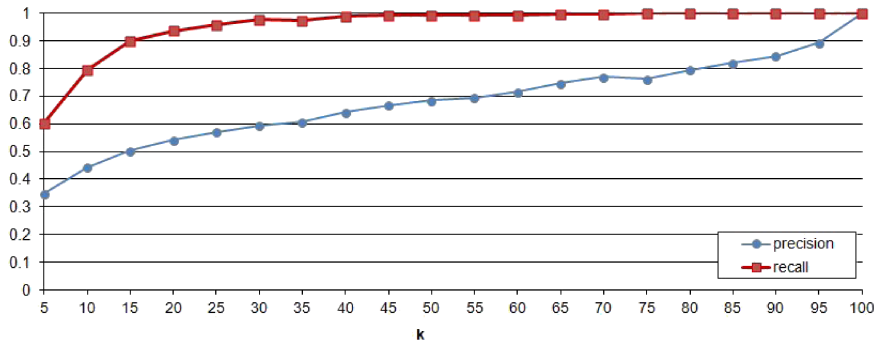

Fig. 5. Precision and Recall of 1 predictor users.

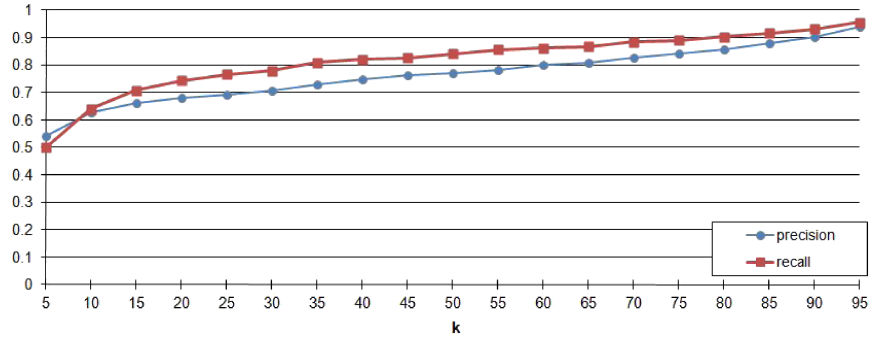

Fig. 6. Precision and Recall of 2 predictor users.

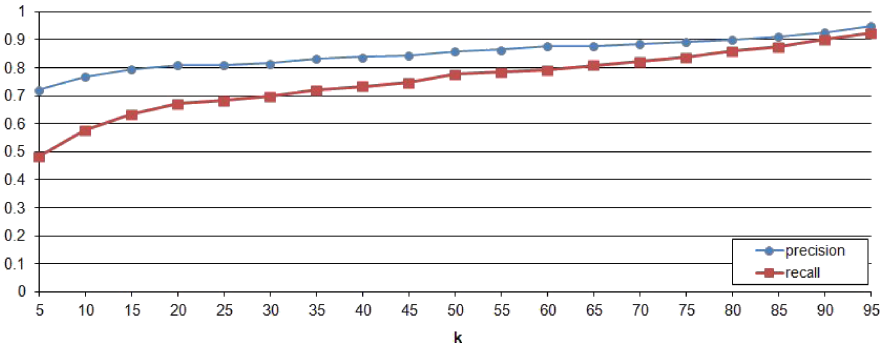

Fig. 7. Precision and Recall of 3 predictor users.

relevant features; precision of over 0.95 is obtained with only $k = 5$ ratings. However, the recall scores are lower, as not all of the features are identified, such that recall reaches the 0.8 mark only when $k = 85$ ratings are available.

These results show that even with a small number of available ratings, the CFS algorithm performs with high accuracy, although the accuracy varies according to the number of predictive features. When a user is reasoning on 1 feature, this feature is correctly identified when only 15 ratings are provided, but the algorithm can also return additional incorrect features. When users exhibit more complex reasoning

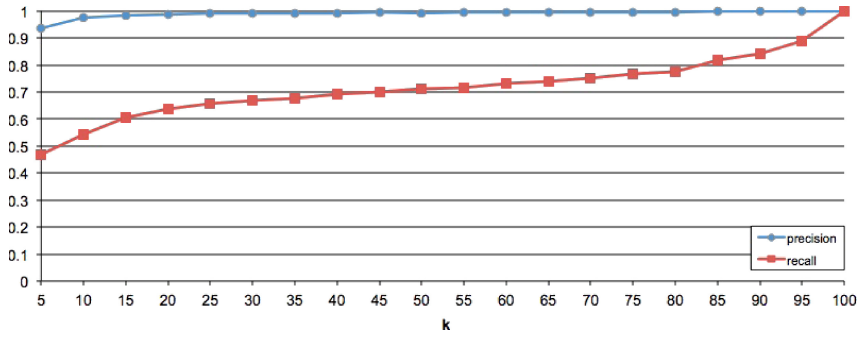

Fig. 8. Precision and Recall of 4 predictor users.

patterns, the CFS algorithm correctly identifies most of the predictive features. However, when only a low number of ratings is available, some predictive features may not be identified. These results correlate with the absolute error in predicting the number of features, which was discussed previously.

To summarise, this section has shown that patterns of reasoning exist, and that users can be grouped according to their reasoning patterns. We have further shown that the accuracy of a recommender system – in this case the M5P algorithm – varies depending on the user group receiving the recommendations. Those who reason on 4 features and, thus, provide a high level of detail, receive the most accurate recommendations, while those who reason on one feature receive the least accurate recommendations.

4. PERSONALIZED ACTIVE LEARNING BASED ON REASONING PREFERENCES

In the presented analyses we used a new dataset of recipe ratings to analyse the reasoning patterns of users. These patterns consider different domain features and characteristics of the recipes; these vary not only in terms of the features themselves, but also in terms of the number of considered features. The CFS algorithm has successfully selected the most predictive features in most cases, even for sparse user profiles containing a small number of ratings. This included both the number of features (i.e., on how many levels a user is reasoning) and the set of features (i.e., what features are important for each user). The number of ratings required for accurate feature selection fluctuates across the groups of users, as more evidence is needed to accurately select features for users applying complex reasoning processes when rating recipes.

The knowledge about users apparent reasoning processes, whether deliberate or implied from their actions, is a valuable asset for a personalized system. The understanding of what is important to a user allows systems to provide accurate personalized services. There are many opportunities where this knowledge could be applied, for instance, to identify collaborative filtering neighbours, to resolve conflicts in group recommendations, or to gather high-value user information. In what follows, we present an investigation into one application that exploits user reasoning patterns to build an adaptive ratings acquisition engine (or active learner), which considers user reasoning processes to request ratings that allow it to build rich user profiles. We include a detailed discussion on the implications of this knowledge in the concluding section of the article.

One of the challenges of recommender systems is the “cold start problem,” where the amount of available user information is insufficient to generate accurate recommendations. One way of combating this is to gather ratings that are seen to attract highly varied ratings (some love and others hate, rather than items that most users like).

Since our analysis has shown that user reasoning processes can be estimated and the important characteristics of recipes can be identified, we suggest to assess each recipe in terms of the contribution that a rating on the recipe can make to the construction of a rich user profile. We refer to this as *active profile learning*. Thus, the aim of the active learner is to prioritise the collection of high-value information over low-value information, with the aim of generating more accurate recommendations, thus, increasing user satisfaction and engagement.

4.1. Active Learning for Recipe Preference Acquisition

Our approach is motivated by ideas incorporated into ensemble based active learning, where the model applied to the data and the data points that serve as input into the model are personalized to each user [Rubens et al. 2011]. This approach recognises that different models are applicable to different users, and that different data points are useful to different models. Since we can identify the predictive features of each user, the active learner can determine areas of data sparsity within a user's profile and suggest the most valuable recipes on which to seek ratings.

For example, consider a user who has provided the system with 5 ratings and has been identified as reasoning on 1 feature, the recipe *category*. If the user provides consistent ratings for recipes of varying values within this category, for instance, high ratings for beef recipes and low ratings for fish recipes, the recommendation algorithm can compute accurate predictions for these categories. However, it is unlikely to accurately predict ratings for recipes that belong to unrated categories, for instance, chicken or lamb. Our active learner identifies values for the selected predictive features that the system is unaware of, and ranks these by the volume of information gained by acquiring ratings for these values.

To test our hypothesis we developed a feature based active learning approach for recipe rating acquisition. The active learner examines the seed set of recipes R_{seed}, for which user ratings are available, and selects a set of yet unrated recipes R_{AL}, for which ratings will be gathered. Firstly, the CFS algorithm analyses the seed set to identify the features on which the user is basing their ratings, as described in Section 3. Once the set of predictive features is known, the algorithm identifies the set of possible values for these features. This allows to identify feature values that are relevant to the user's rating bias, but the model has no information about these values. This is equivalent to discovering the reasoning tree and identifying the nodes for which we currently have no ratings.

The information gain associated with obtaining ratings for the identified feature values is determined by the number of recipes associated with this feature or feature combination. The more recipes associated with the feature-value combination, the more information will be added to the model by obtaining user ratings. Thus, the learner ranks the identified feature-values according to the number of recipes associated with the value. R_{AL} is generated by looping through the ranked features and randomly selecting recipes that match the feature selection criteria. In our case, 5 recipes are added to R_{AL} at each stage, with the process of feature selection, feature ranking, and recipe selecting being completed after every 5 ratings. Thus, the steps involved are as follows.

(1) Run the CFS algorithm on R_{seed} to identify predictive features F_1,..., F_n.
(2) Determine all feature-value combinations for the predictive features F_1,..., F_n.
(3) Filter out feature-value combinations already included in R_{seed}.
(4) Rank feature-value combinations according to the number of recipes represented by the combination.
(5) Add one random recipe per feature-value combination until R_{AL} contains 5 recipes.

(6) Present R_{AL} to user for rating.
(7) Update R_{seed}.
(8) Repeat (1)–(7).

4.2. Evaluation

The methodology employed to evaluate the active learning component mirrors the previously detailed analysis, which identified the predictive features and used these to predict recipe ratings. The analysis simulated the process typically seen at sign up or registration to a recommender system, where ratings are sought on batches of items to inform the user profile. In this analysis, we are particularly interested in the prediction accuracy of the M5P recommendation algorithm in early stages of user membership, when R_{seed} is small and a few user ratings are available. We focus again on the 349 users, who provided in excess of 100 recipe ratings.

For each user, 20 recipe ratings were randomly selected as test items R_{test}, on which the accuracy of the algorithms was assessed. In addition, 5 ratings were randomly selected as the seed set R_{seed} for the learner. The user profile was gradually increased in batches of 5 ratings (determined by the appropriate active learner algorithms), and the accuracy of the predicted ratings for recipes in R_{test} was recorded based on the user profile being built. Three active learning algorithms were evaluated. The first is the *personalized* active learner detailed in this section. In this instantiation, a batch of 5 recipes R_{AL} was identified by the active learner, with 1 randomly selected recipe from each of the top ranked feature-values being added to R_{AL}. The second is a standard *entropy* based learner, which ranked each recipe according to the variation of ratings obtained across the entire set of users, prioritising those with the highest variability of ratings. Finally, a *random* algorithm selected 5 random recipes to be added to the user profile. For the sake of simplicity, in all cases the recipes selected for addition to the profile were limited to those for which we a real rating was available. As in the previous analyses, we evaluated the accuracy of the M5P prediction generation algorithm [Freyne et al. 2011b].

4.3. Results

Overall, the impact of the personalized active learner was low. While we observed more accurate results, similar accuracy was obtained by using the nonpersonalized entropy based approach, as can be seen in Figure 9, where for clarity we concentrate on profiles smaller than 50 ratings. We note that the performance of both nonrandom active learners, which identify high-value ratings on which to solicit information, surpasses the performance of the random recipe selection. We note that the most pronounce differences are observed when the number of available ratings k is low, but the performance of the algorithms gets closer once the user profile contains more ratings. We also note the highly similar performance of the personalized active learner and the entropy based algorithm. Overall, altering the order of recipe ratings allows to build richer user profiles and improve the accuracy of the generated predictions.

Delving deeper, we examined the performance of the active learner for various groups of users. Figures 10–13 show the performance of the three learners for each group. Figure 10 focuses on the group of users reasoning on 1 predictive features shows that similar MAE values are obtained for $k > 25$, but clear differences between the two intelligent and the random learner are observed for low values of k, which is where we are most likely to see the impact of the active learner. The MAE of the random profile additions is high and it even increases as the user profile grows from 5 to 15 ratings. On the contrary, the intelligent learners provide the prediction algorithm with ratings that reduce MAE and improve accuracy, with the addition of each batch of recipe ratings.

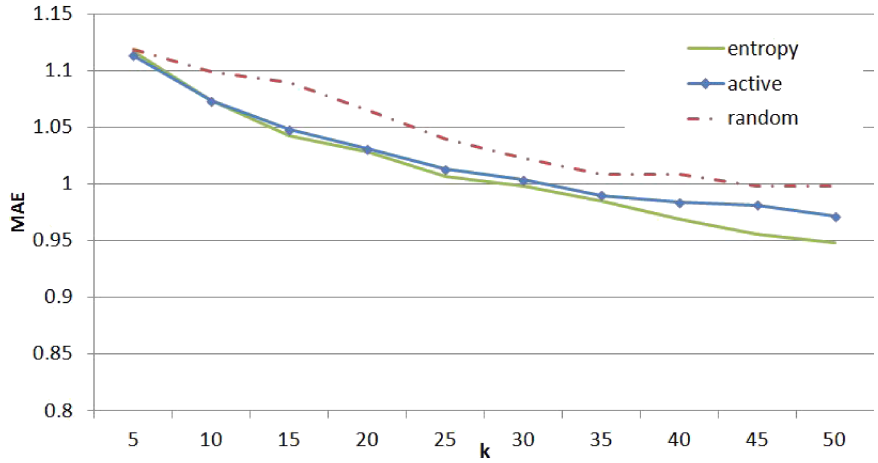

Fig. 9. MAE of active learner strategies.

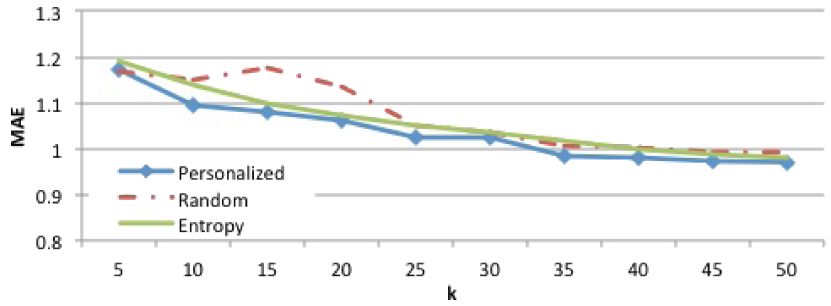

Fig. 10. MAE of 1 predictor users.

In Figures 11 and 12, which correspond to groups of users reasoning on 2 and 3 features, respectively, we observe the improvement in the accuracy of the predictions for small values of k. Both the personalized and entropy based learner outperforms the random learner for $k < 35$, with the differences being more clearly pronounced for $k < 30$ in Figure 11 and $k < 25$ in 12. Again, the random learner provides very little information when growing the profiles from 5 to 10 ratings, whereas the intelligent learners both provide valuable information that improves the accuracy of the predictions. In Figure 13, the impact is less clear and we observe all three algorithms obtaining similar MAE scores. We suggest that the decision tree constructed by the M5P algorithm is sufficiently broad that there are too many feature-value combinations providing valuable information to the algorithm, such that a random selection will in most cases provide a similar information as the one that can be provided by an intelligent learner.

These results show that the intelligent selection of recipes on which to solicit ratings has a profound impact on the accuracy of the generated predictions when limited user information is available. We have shown that in the early stages of user membership it is advantageous to use either a personalized active learner or an entropy based learner over the random addition of recipes. We had intended to use the entropy based learner as a second baseline, but the benefits of this approach were comparable to those of the personalized active learner. Requesting ratings for recipes with high entropy was found to be as informative as ratings for recipes covering a variety of feature-values

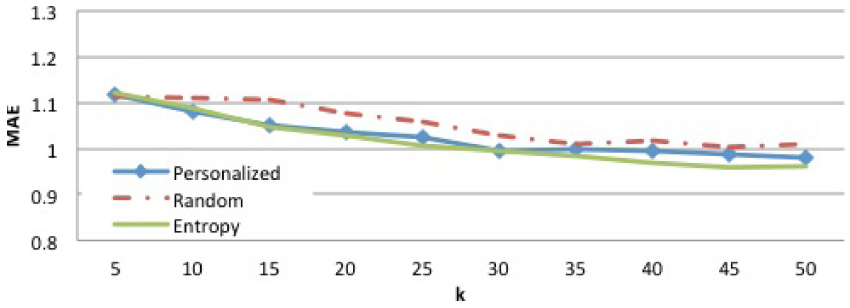

Fig. 11. MAE of 2 predictor users.

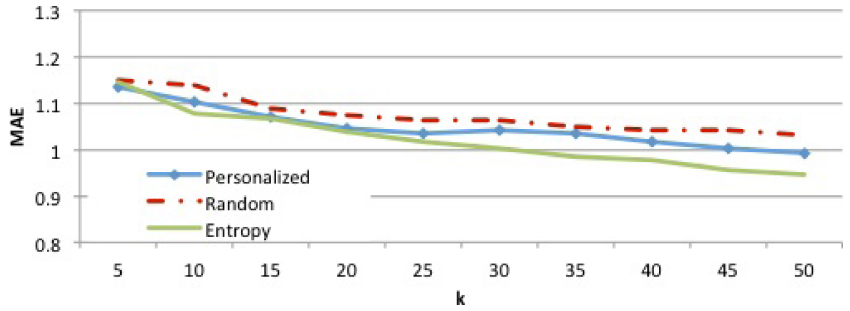

Fig. 12. MAE of 3 predictor users.

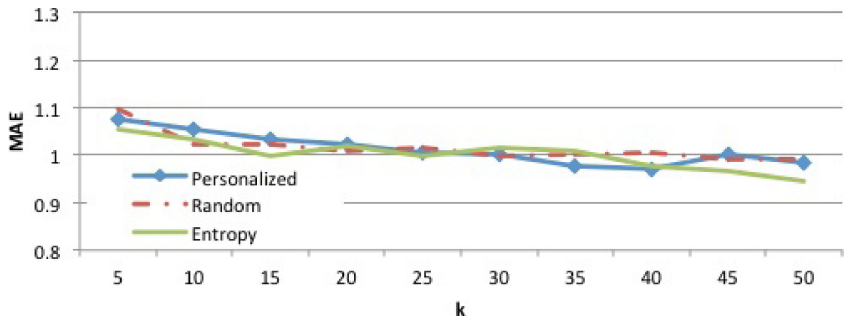

Fig. 13. MAE of 4 predictor users.

for the active learner. Overall, as k increased beyond the point where the personalized active learner is showing enhancements over the random learner (around $k = 30$), the entropy based learner could still facilitate additional increases in the accuracy of the predictions.

5. DISCUSSION

The work presented in this article points to varying reasoning patterns employed by users when providing ratings and preference information in an online system. It assumes that users apply different reasoning patterns for decision making and that these differences are apparent in the captured profile information and extractable using feature selection algorithms.

We have illustrated the user reasoning mining processes in the domain of food, but can this knowledge and learned lessons be transferred beyond this dataset and

domain? We posit that one of the key factors in using this technique in other domains is in the possible dominance of features in the predictive feature set. In our dataset, there were several features that we would describe as core features and fundamental component of a recipe, for instance, the category and the cuisine type, and there also were rather peripheral features, for instance, the cooking complexity. In the food domain people tend to reason base their reasoning on the core features, along the lines of "I love Thai food" or "I dislike seafood". This is reflected in our feature selection, such that a large variety and subtle combinations of these features exist across our users.

Other domains such as fashion or design, where the intricacies and linkages of items that make up an outfit or look are balanced and highly interconnected, are not as likely to benefit from obtaining user reasoning knowledge. If the features of the items in question are imbalanced in terms of their impact on user feedback, then it is likely that the selected features will dominated the reasoning process. This trend is expected in datasets such as TV or movie watching, where the genre of programs is likely to be the sole most informative feature for the majority of users. Features like the country where a program was recorded or the cinematographer, are not expected to be as influential, resulting in a prominent dominance of a single feature, and reducing the value of uncovering predictive features.

If in the food and similar domains, where the ratings correctly reflect user reasoning processes, then this knowledge could be exploited by a recommender system. Unlike many domains, where consumption of recommendations is a single-shot event, the context of the sequencing of recommendations is important in meal planning. A recipe recommender must be responsive to a user's preferences in order to identify recipes that they will enjoy, but also to the user's preferences for consumption frequencies of ingredients and recipes. However, it may take a long time to learn these trends. An understanding of user preferences may allow us to group recipes by predictive features as a proxy of the consumption frequency. In a similar vein, recipe similarity computation could incorporate the predictive features of each user, in order to suggest diverse meal plans. When planning for groups, the knowledge of the predictive features for each group member may be taken into account when resolving conflicts. For example, it would be interesting to assess whether the recommendations can be influenced not only by the preferences of users and their social roles, but also by their predictive features, such that each user is satisfied by the features of the recommended items.

It is also important to highlight the speculative nature of the ratings in our dataset. In most recommender system datasets, users rate items that they consumed or experienced in the past. For example, they rate movies that they watched, books that they read, or hotels in which they stayed. This is probably not the case for the Total Wellbeing Diet recipes. The recipes included in the diet are high-protein and peculiar to the Australian cuisine, whereas most Mechanical Turk respondents were from Asia. Thus, it is quite likely that most of them have never eaten the rated recipes and provided ratings based on their expected appreciation of the meal, rather than based on their past experience. In this sense, the nature of the gathered recipe ratings based on the expected utility may apparently differ from the other recommender datasets based on recollected utility of past experiences. How does the different nature affect the uncovered reasoning processes? Would similar patterns be uncovered for recollected ratings?

Another question of suitable merit that requires discussion here is whether the user reasoning patterns and the selected predictive features could be affected by the way the users were asked and the information was gathered. As discussed previously, there are many factors that are seen to influence the information given by a user to a system: social pressure, visibility of contributions, layout and prompts, return on effort, task complexity, system reputation, and others [Cosley et al. 2003; Jameson 2012].

Our analysis has shown that within a single system with identical rewards, task complexity, and visibility, a variety of apparent reasoning or human decision processes are identifiable in the resulting dataset. What we do not know yet is whether the input from users is the "ground truth" or whether different ratings would have been provided had we paid more, provided personalized meal recommendations in return for ratings, or asked for ratings in a different manner. We know from our dataset that users are reasoning differently, but we cannot tell if their reasoning would be impacted by the context of the experiment.

Recent work in the area of persuasive technologies [Berkovsky et al. 2012; Kaptein et al. 2012], has shown that users can be persuaded by different forms of persuasion (e.g., authority, peer pressure, conformity to normd). This highlights the need to understand the target user and to personalise the persuasive features being applied for maximum impact. We believe that this theory is likely to apply to the influence that a system has on the quality of information provided by the users. Some people will care and potentially change their input if this is seen by others. Similarly, some users may be affected by "social influence," conform to the behaviour of others and only like things approved by their friends. In the same vein, some users are likely to want the highest quality recommendations in a movie recommender system (movie buffs), whereas others may be satisfied with good recommendations (movie fans with some time to spare). In the case of crowd-sourced information, some people may have a better work ethics than others and put more effort into providing ratings regardless of the level of payment. Although our work did not address these intricacies, the question of determining the individual impact of the conditions and data collection rewards on the quality of the gathered data remains open.

The variability in data quality and accuracy is well documented in crowd-sourcing services like Mechanical Turk. Crowd-sourcing is popular and trusted in the domains of machine translation, text mining, and image analysis, where human judgment is necessary for metadata gathering and training of learning algorithms. We predict an increased usage of crowd-sourced data to supplement the generation of recommendations for emerging domains, like wellbeing and lifestyle. The drive to generate valuable rather than only accurate recommendations in these domains dictates moving beyond the tried and tested datasets centred around movies, books, and music. We predict that many research and industry parties may turn to crowd-sourcing, in order to acquire data, investigate algorithms, bootstrap new systems, and conduct online user studies. Best crowd-sourcing practices suggest the implementation of simple quality controls like time thresholds and consistency checks to identify bogus users. Beyond the exclusion of obvious noise, can we tell anything about the effort invested in the data provision or the quality of data? If so, how can this information by leveraged by recommendation algorithms?

6. CONCLUSION

In this work, we have investigated the applicability of recommender techniques to generate recipe recommendations and identified the performance enhancements achieved by machine learning techniques. Analyses of the results have shown that users appear to reason on various levels when rating recipes and that various combinations of metadata are seen to have different predictive qualities for different users. This information has assisted us in understanding how users provide recipe ratings and suggests opportunities for ways, in which this knowledge could be used to benefit the performance and acceptance of recommender systems.

We have shown one example of the exploitation of this information by an active learning algorithm, which identified high-value items in order to maximise the accuracy of the recommendations in early stages. We developed a personalized active

learner, which exploits a user's apparent decision making patterns to rank items in the repository in terms of their importance to the identified reasoning process and their coverage of the repository. By asking users to rate highly informative items, in contrary to random items, we have shown an increase in the predictive accuracy of the M5P rating prediction algorithm for user profiles containing a low number of ratings.

Recommending food and recipes is a complex domain, with multiple factors impacting a user's decision to take up a recommendation. As a reflective step, we aim to revisit the metadata included on our recipe dataset, to ensure that the included metadata are sufficiently broad to capture a user's motivation, paying particular attention to cooking times, ingredient costs, and product availability. Moving forward, we aim to consider other opportunities for the exploitation of user reasoning within a food recommender system, in order to increase the accuracy and enhance the quality of the recommendations provided to users. In particular, we foresee three opportunities beyond the data acquisition process evaluated here: in the sequencing and diversification of recommendations, and in group recommendations.

ACKNOWLEDGMENTS

The authors acknowledge Mealopedia.com and Penguin Group (Australia) for permission to use their data.

REFERENCES

Adomavicius, G. and Tuzhilin, A. 2011. Context-aware recommender systems. In *Recommender Systems Handbook*. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor Eds., Springer, 217–253.

Adomavicius, G., Mobasher, B., Ricci, F., and Tuzhilin, A. 2011. Context-aware recommender systems. *AI Magazine 32*, 3, 67–80.

Amatriain, X., Pujol, J., Tintarev, N., and Oliver, N. 2009. Rate it again: Increasing recommendation accuracy by user re-rating. In *Proceedings of the 3rd ACM Conference on Recommender Systems*. ACM, 173–180.

Baltrunas, L., Ludwig, B., Peer, S., and Ricci, F. 2012. Context relevance assessment and exploitation in mobile recommender systems. *Personal Ubiq. Comput. 16*, 5, 507–526.

Berkovsky, S., Kuflik, T., and Ricci, F. 2009. Cross-representation mediation of user models. *User Model. User-Adapt. Interact. 19*, 1–2, 35–63.

Berkovsky, S., Freyne, J., and Oinas-Kukkonen, H. 2012. Influencing individually: Fusing personalization and persuasion. *ACM Trans. Interact. Intell. Syst. 2*, 2, 9.

Buhrmester, M., Kwang, T., and Gosling, S. 2011. Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspect. Psychol. Sci. 6*, 1, 3–5.

Chen, L. and Pu, P. 2004. Survey of preference elicitation methods. Tech. rep. No. IC/200467, Swiss Federal Institute of Technology in Lausanne.

Cosley, D., Lam, S., Albert, I., Konstan, J., and Riedl, J. 2003. Is seeing believing?: How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 585–592.

de Gemmis, M., Felfernig, A., Lops, P., Ricci, F., Semeraro, G., and Willemsen, M. C. 2012. Recsys '12 workshop on human decision making in recommender systems. In *Proceedings of the ACM International Conference on Recommender Systems*. P. Cunningham, N. J. Hurley, I. Guy, and S. S. Anand Eds., ACM, 347–348.

Elahi, M., Repsys, V., and Ricci, F. 2011. Rating elicitation strategies for collaborative filtering. In *Ecommerce and Web Technologies*, 160–171.

Freyne, J. and Berkovsky, S. 2010a. Intelligent food planning: Personalized recipe recommendation. In *Proceedings of the International Conference on Intelligent User Interfaces*.

Freyne, J. and Berkovsky, S. 2010b. Recommending food: Reasoning on recipes and ingredients. In *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*. 381–386.

Freyne, J., Berkovsky, S., Baghaei, N., Kimani, S., and Smith, G. 2011a. Personalized techniques for lifestyle change. In *Proceedings of the Conference on Artificial Intelligence in Medicine*. 139–148.

Freyne, J., Berkovsky, S., and Smith, G. 2011b. Recipe recommendation: Accuracy and reasoning. In *Proceedings of the Conference on User Modeling, Adaption and Personalization*. 99–110.

Golbandi, N., Koren, Y., and Lempel, R. 2010. On bootstrapping recommender systems. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 1805–1808.

Hall, M. 1999. Correlation-based feature selection for machine learning. Ph.D. thesis, The University of Waikato.

Hammond, K. 1986. CHEF: A model of case-based planning. In *Proceedings of the 5th National Conference on Artificial Intelligence*. Vol. 1.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. 22*, 1, 5–53.

Hinrichs, T. 1989. Strategies for adaptation and recovery in a design problem solver. In *Proceedings of the Workshop on Case-Based Reasoning*.

Jameson, A. 2011. Tutorial / What every IUI researcher should know about human choice and decision making. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, P. Pu, M. J. Pazzani, E. André, and D. Riecken Eds., ACM, 461–462.

Jameson, A. 2012. Choices and decisions of computer users. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications* 3rd Ed., CRC Press, Boca Raton, FL.

Kaptein, M., De Ruyter, B., Markopoulos, P., and Aarts, E. 2012. Adaptive persuasive systems: A study of tailored persuasive text messages to reduce snacking. *ACM Trans. Interact. Intell. Syst. 2*, 2, 10.

Kittur, A., Chi, E., and Suh, B. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems*. ACM, 453–456.

Kuflik, T., Wecker, A. J., Cena, F., and Gena, C. 2012. Evaluating rating scales personality. In *Proceedings of the 20th International Conference on User Modelling, Adaptation and Personalization*. J. Masthoff, B. Mobasher, M. C. Desmarais, and R. Nkambou Eds., Lecture Notes in Computer Science, vol. 7379, Springer, 310–315.

Lops, P., Gemmis, M., and Semeraro, G. 2011. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor Eds., Springer, 73–105.

Noakes, M. and Clifton, P. 2005. *The CSIRO Total Wellbeing Diet Book*. Penguin Group, Australia.

Noakes, M. and Clifton, P. 2006. *The CSIRO Total Wellbeing Diet Book 2*. Penguin Group, Australia.

Paolacci, G., Chandler, J., and Ipeirotis, P. 2010. Running experiments on Amazon Mechanical Turk. *Judgment Decision Mak. 5*, 5, 411–419.

Pazzani, M. and Billsus, D. 2007. Content-based recommendation systems. In *The Adaptive Web*, 325–341.

Quinlan, J. 1992. Learning with continuous classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*. 343–348.

Rashid, A., Albert, I., Cosley, D., Lam, S., McNee, S., Konstan, J., and Riedl, J. 2002. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*. ACM, 127–134.

Rashid, A., Karypis, G., and Riedl, J. 2008. Learning preferences of new users in recommender systems: An information theoretic approach. *ACM SIGKDD Newslett. 10*, 2, 90–100.

Rubens, N., Kaplan, D., and Sugiyama, M. 2011. Active learning in recommender systems. In *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor Eds., Springer, 735–767.

Svensson, M., Höök, K., Laaksolahti, J., and Waern, A. 2001. Social navigation of food recipes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, 341–348.

Tintarev, N. and Masthoff, J. 2011. Designing and evaluating explanations for recommender systems. In *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor Eds., Springer, 479–510.

van Pinxteren, Y., Geleijnse, G., and Kamsteeg, P. 2011. Deriving a recipe similarity measure for recommending healthful meals. In *Proceedings of the International Conference on Intelligent User Interfaces*. 105–114.

Wang, Y. and Witten, I. 1996. Induction of model trees for predicting continuous classes. In *Poster Papers of the 9th European Conference on Machine Learning*.

Zhang, Q., Hu, R., MacNamee, B., and Delany, S. J. 2008. Back to the future: Knowledge light case base cookery. Tech. rep., Dublin Institute of Technology.

Received March 2012; revised October 2012, March 2013; accepted April 2013