RecSys'14 Joint Workshop on Interfaces and
Human Decision Making for Recommender Systems (IntRS'14)

Proceedings of the

# Joint Workshop on Interfaces and Human Decision Making in Recommender Systems

October 6, 2014

In conjunction with the
**8th ACM Conference on Recommender Systems**
Foster City, Silicon Valley, USA

Edited by
Nava Tintarev, John O'Donovan, Peter Brusilovsky,
Alexander Felfernig, Giovanni Semeraro, Pasquale Lops

# Preface

As interactive intelligent systems, recommender systems are developed to suggest items that match users' preferences. Since the emergence of recommender systems, a large majority of research has focused on objective accuracy criteria and less attention has been paid to how users interact with the system and the efficacy of interface designs from users' perspectives. The field has reached a point where it is ready to look beyond algorithms, into users' interactions, decision making processes and overall experience.

Accordingly, the goals of the workshop are to explore the human aspects of recommender systems, with a particular focus on the impact of interfaces and interaction design on decision-making and user experiences with recommender systems, and to explore methodologies to evaluate these human aspects of the recommendation process that go beyond traditional automated approaches.

The aim is to bring together researchers and practitioners around the topics of designing and evaluating novel intelligent interfaces for recommender systems in order to:
(1) share research and techniques, including new design technologies and evaluation methodologies (2) identify next key challenges in the area, and (3) identify emerging topics.

The workshop covers three interrelated themes: a) user interfaces (e.g. visual interfaces, explanations), b) interaction, user modeling and decision-making (e.g. decision theories, argumentation, detection and avoidance of biases), and c) evaluation (e.g. case studies and empirical evaluations).

This workshop aims at creating an interdisciplinary community with a focus on the interface design issues for recommender systems and promoting collaboration opportunities between researchers and practitioners.

The workshop consists of a mix of eight presentations of papers in which results of ongoing research as reported in these proceedings are presented and one invited talk by Julita Vassileva presenting "Visualization and User Control of Recommender Systems". The workshop is closed by a final discussion session.

Nava Tintarev, John O'Donovan, Peter Brusilovsky, Alexander Felfernig, Giovanni Semeraro and Pasquale Lops

*September 2014*

# Organizing Committee

## Workshop Co-Chairs

Nava Tintarev, University of Aberdeen, UK
John O'Donovan, University of California, Santa Barbara
Peter Brusilovsky, University of Pittsburgh
Alexander Felfernig, Graz University of Technology, Austria
Giovanni Semeraro, University of Bari "Aldo Moro", Italy
Pasquale Lops, University of Bari "Aldo Moro", Italy

## Program Committee

Anthony Jameson, DFKI, Germany
Robin Burke, DePaul University, USA
Shlomo Berkovsky, NICTA, Australia
Li Chen, Hong Kong Baptist University
Jaegul Choo, Georgia Tech, USA
Jill Freyne, CSIRO, Australia
Gerhard Friedrich, Alpen-Adria-Universitaet Klagenfurt
Franca Garzotto, Politecnico di Milano, Italy
Marco de Gemmis, Dipartimento di Informatica, University of Bari
Cleotilde Gonzalez, Carnegie Mellon University, USA
Sergiu Gordea, AIT, Austria
Tobias Hollerer, University of California, Santa Barbara, USA
Dietmar Jannach, TU Dortmund, Germany
Henry Lieberman, MIT Media Lab, USA
Bart Knijnenburg, University of California, Irvine, USA
Joseph Konstan, University of Minnesota, USA
Gerald Ninaus, Graz University of Technology, Austria
Denis Parra, PUC, Chile
Francesco Ricci, Free University of Bozen-Bolzano, Italy
Olga Santos, aDeNu, Spain
Christin Seifert, Universitet Passau, Germany
Martijn Willemsen, Eindhoven University of Technology, Netherlands
Julita Vassileva, University of Saskatchewan
Jesse Vig, Palo Alto Research Center
Markus Zanker, Alpen-Adria-Universitaet Klagenfurt

# Table of Contents

## Invited presentation

**Visualization and User Control of Recommender Systems**
*Julita Vassileva* 1

## Accepted papers

**De-Biasing User Preference Ratings in Recommender Systems**
*Gediminas Adomavicius, Jesse Bockstedt, Shawn Curley, Jingjing Zhang* 2

**Investigation of User Rating Behavior Depending on Interaction Methods on Smartphones**
*Shabnam Najafian, Wolfgang Wörndl, Béatrice Lamche* 10

**Interactive Explanations in Mobile Shopping Recommender Systems**
*Béatrice Lamche, Ugur Adigüzel, Wolfgang Wörndl* 14

**If You Liked Herlocker et al.'s Explanations Paper, then You Might Like This Paper Too**
*Derek Bridge, Kevin Dunleavy* 22

**Choicla: Intelligent Decision Support for Groups of Users in the Context of Personnel Decisions**
*Martin Stettinger, Alexander Felfernig* 28

**An Empirical Study on the Persuasiveness of Fact-based Explanations for Recommender Systems**
*Markus Zanker, Martin Schoberegger* 33

**The Effect of Different Set-based Visualizations on User Exploration of Recommendations**
*Katrien Verbert, Denis Parra, Peter Brusilovsky* 37

**A Visualization Interface for Twitter Timeline Activity**
*Wesley Waldner, Julita Vassileva* 45

# Visualization and User Control of Recommender Systems

Julita Vassileva
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada
julita.vassileva@usask.ca

## Abstract

The talk will give an overview of some of the existing approaches for visualizing recommendation mechanisms and eventually allowing users to control them.

Starting with work from the area of open/scrutable learner models in the area of intelligent tutoring systems, through approaches for explaining recommendations to approaches visualizing aspects of collaborative, hybrid and social recommenders, as well as the filter bubble, the talk will be based both on the speakers' own work in this area and on a review of other work and will touch on some philosophical issues about how to evaluate recommendations.

# De-Biasing User Preference Ratings in Recommender Systems

Gediminas Adomavicius
University of Minnesota
Minneapolis, MN
gedas@umn.edu

Jesse Bockstedt
University of Arizona
Tucson, AZ
bockstedt@email.arizona.edu

Shawn Curley
University of Minnesota
Minneapolis, MN
curley@umn.edu

Jingjing Zhang
Indiana University
Bloomington, IN
jjzhang@indiana.edu

## ABSTRACT

Prior research has shown that online recommendations have significant influence on users' preference ratings and economic behavior. Specifically, the self-reported preference rating (for a specific consumed item) that is submitted by a user to a recommender system can be affected (i.e., distorted) by the previously observed system's recommendation. As a result, anchoring (or anchoring-like) biases reflected in user ratings not only provide a distorted view of user preferences but also contaminate inputs of recommender systems, leading to decreased quality of future recommendations. This research explores two approaches to removing anchoring biases from self-reported consumer ratings. The first proposed approach is based on a computational post-hoc de-biasing algorithm that systematically adjusts the user-submitted ratings that are known to be biased. The second approach is a user-interface-driven solution that tries to minimize anchoring biases at rating collection time. Our empirical investigation explicitly demonstrates the impact of biased vs. unbiased ratings on recommender systems' predictive performance. It also indicates that the post-hoc algorithmic de-biasing approach is very problematic, most likely due to the fact that the anchoring effects can manifest themselves very differently for different users and items. This further emphasizes the importance of proactively avoiding anchoring biases at the time of rating collection. Further, through laboratory experiments, we demonstrate that certain interface designs of recommender systems are more advantageous than others in effectively reducing anchoring biases.

## Keywords

Recommender systems, anchoring effects, rating de-biasing

## 1. INTRODUCTION

Recommender systems are prevalent decision aids in the electronic marketplace, and online recommendations significantly impact the decision-making process of many consumers. Recent studies show that online recommendations can manipulate not only consumers' preference ratings but also their willingness to pay for products [1,2]. For example, using multiple experiments with TV shows, jokes and songs, prior studies found evidence that a recommendation provided by an online system serves as an anchor when consumers form their preference for products, even at the time of consumption [1]. Furthermore, using the system-predicted ratings as a starting point and biasing them (by perturbing them up or down) to varying degrees, this anchoring effect was observed to be continuous, with the magnitude proportional to the size of the perturbation of the recommendation in both positive and negative directions – about 0.35-star effect for each 1-star perturbation on average across all users and items [1]. Additionally, research found that recommendations displayed to participants significantly pulled their willingness to pay for items in the direction of the recommendation, even when controlling for participants' preferences and demographics [2].

Based on these previous studies, we know that users' preference ratings can be significantly distorted by the system-predicted ratings that are displayed to users. Such distorted preference ratings are subsequently submitted as users' feedback to recommender systems, which can potentially lead to a biased view of consumer preferences and several potential problems [1,5]: (i) biases can contaminate the recommender system's inputs, weakening the system's ability to provide high-quality recommendations in subsequent iterations; (ii) biases can artificially pull consumers' preferences towards displayed system recommendations, providing a distorted view of the system's performance; (iii) biases can lead to a distorted view of items from the users' perspectives. Thus, when using recommender systems, anchoring biases can be harmful to system's use and value, and the removal of anchoring biases from consumer ratings constitutes an important and highly practical research problem.

In this research, we focus on the problem of "de-biasing" self-reported consumer preference ratings for consumed items. We first empirically demonstrate that the use of unbiased preference ratings as inputs indeed leads to higher predictive accuracy of recommendation algorithms than the use of biased preference ratings. We then propose and investigate two possible approaches to tackle the rating de-biasing problem:

1) Post-hoc rating adjustment (reactive approach): a computational approach that attempts to adjust the user-submitted ratings by taking into account the system recommendation observed by the user.
2) Bias-aware interface design for rating collection (proactive approach): a design-based approach that employs a user interface for rating collection by presenting recommendations in a way that eliminates or reduces anchoring effects.

## 2. BACKGROUND

Prior literature has investigated how the cues provided by recommender systems influence online consumer behavior. For example, Cosley et al. (2003) found that users showed high test-retest consistency when being asked to re-rate a movie with no prediction provided [5]. However, when users were asked to re-rate a movie while being shown a "predicted" rating that was altered upward or downward from their original rating by a single fixed amount of one rating point (i.e., providing a high or low anchor), users tended to give higher or lower ratings, respectively, as compared to a control group receiving accurate original ratings. This showed that anchoring could affect users' ratings based on preference recall, for movies seen in the past and now being evaluated.

Adomavicius et al. (2013) looked at a similar effect in an even more controlled setting, in which the consumer preference ratings for items were elicited at the time of item consumption [1]. Even without a delay between consumption and elicited preference, anchoring effects were observed. The displayed predicted ratings, when perturbed to be higher or lower, affected the submitted consumer ratings to move in the same direction.

Prior research also found that recommendations not only significantly affect consumers' preference ratings but also their economic behavior [2]. Researchers present the results of two controlled experiments in the context of purchasing digital songs. The studies found strong evidence that randomly assigned song recommendations affected participants' willingness to pay, even when controlling for participants' preferences and demographics. Similar effects on willingness to pay were also observed when participants viewed actual system-generated recommendations that were intentionally perturbed up or down (introducing recommendation error).

The anchoring biases occurring due to system-generated recommendations can potentially lead to several issues. From the consumers' perspective, anchoring biases can distort (or manipulate) consumers' preferences and economic behavior, and therefore lead to suboptimal product choices and distorted preference ratings. From the retailer's perspective (e.g., Amazon, eBay), anchoring biases may allow third-party agents to manipulate the recommender system (e.g., by strategically adding malicious ratings) so that it operates in their favor. This would reduce consumers' trust in the recommender system and harm the success of the system in the long term. From the system designers' perspective, the distorted user preference ratings that are subsequently submitted as consumers' feedback to recommender systems can contaminate the inputs of the recommender system, reducing its effectiveness. Therefore, removing the bias of recommendations represents an important research question. In the following sections, we empirically study two possible approaches for tackling the rating de-biasing problem.

# 3. APPROACH I: POST-HOC RATING ADJUSTMENT

## 3.1 Rating Adjustment Algorithm

The underlying intuition of post-hoc rating adjustment is to "reverse-engineer" consumers' true non-biased ratings from the user-submitted ratings and the displayed system recommendations (that were observed by the users). For this, we use the information established by previous research that, in aggregate, the anchoring effect of online recommendations is linear and proportional to the size of the recommendation perturbation [1]. As depicted in Fig 1, the deviation of the submitted rating from the user's unbiased rating (i.e., *Dev*) should be proportional to the deviation of the system's displayed prediction from the user's unbiased rating (i.e., *α* × *Dev*). Given the user's submitted rating, the displayed system prediction, and the expected anchoring effect size, we develop a computational rule to systematically reverse-engineer user's unbiased ratings.

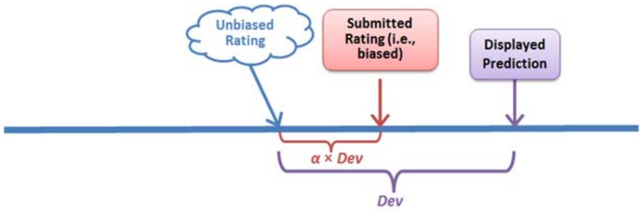


**Fig 1. Post-Hoc Rating Adjustment Illustration**

Mathematically, let $\alpha$ be the expected slope (i.e., proportionality coefficient) of the bias relative to the size of rating perturbation, $R_{ui}^{Shown}$ be the value of the system's predicted rating on item $i$ that was shown to user $u$, and $R_{ui}^{Submitted}$ be the user's submitted rating after seeing the system's prediction. We estimate the unbiased rating of user $u$ for item $i$, i.e., $R_{ui}^{UnbiasedRating}$ using the formula below:

$$R_{ui}^{UnbiasedRating} = (R_{ui}^{Submitted} - \alpha \times R_{ui}^{Shown})/(1-\alpha).$$

In this post-hoc adjustment approach, the value of $\alpha$ is determined by the observed slope of the bias and can range between 0 (inclusive) and 1 (exclusive). Varying the size of $\alpha$ within [0, 1) changes the degree of rating adjustment, i.e., a larger value of $\alpha$ leads to a larger adjustment to the submitted rating, while $\alpha = 0$ means no adjustment is made. In our experiments, the slope $\alpha$ can be either a global constant that applies to all users and items, or user-specific values determined by an individual user's tendency of anchoring on the system's recommendations.

## 3.2 Computational Experiments

### *3.2.1 Joke Rating Dataset*

Our experiments use a Joke rating dataset collected in laboratory settings by a prior study on anchoring effects of recommender systems [1]. The dataset includes ratings provided by 61 users on 100 jokes. At the beginning of the study, participants first evaluated 50 jokes without seeing a system's recommendations. These initial ratings reflect user's unbiased preferences and were used as a basis for computing the system's predictions. Next, the participants received 40 jokes with a predicted rating displayed. Among them, thirty of these predicted ratings were perturbed to various degrees and ten were not perturbed. These 40 jokes were randomly intermixed.

Prior research has observed continuous and linear anchoring effects on this joke rating dataset. On average, the anchoring slope across all users and items is $\alpha = 0.35$, and is significantly positive. Individual linear regression models were also obtained at an individual-user level. These user-specific regression slopes are predominately positive, suggesting that significant anchoring bias was observed for most participants.

For the post-hoc de-biasing experiments, we partition the joke ratings for each user into two subsets. The first subset contains the initial 50 ratings provided by each user before seeing any system recommendations (i.e., unbiased), and the second subset contains the subsequent 40 user ratings submitted after user received system's recommendations with various levels of perturbations (i.e., biased ratings). Next, on the 40 biased ratings, we apply the post-hoc rating adjustment rule to remove possible anchoring biases to recover users' unbiased ratings.

To evaluate the benefits of post-hoc rating adjustment, we compute predictive accuracy (measured as Root Mean Squared Error, i.e., RMSE) of standard recommendation algorithms using the adjusted ratings (i.e., de-biased) as training data and the initial ratings (i.e., unbiased) as testing data. We then compare this accuracy performance with that of using actual submitted ratings (i.e., biased) as training data and the same initial ratings as testing data. If rating de-biasing is successful, the prediction accuracy on "de-biased" ratings should be better than accuracy on "biased" ratings. We explore the post-hoc rating adjustment under a variety of settings, as described below.

### *3.2.2 Experiments*

Our first experiment investigated the accuracy performance on unbiased, biased, and de-biased ratings adjusted based on various rules and statistically compared their differences. First, we randomly divided the 50 initial (unbiased) ratings provided by each user into two equal subsets with 25 ratings per user (aggregated across all users) in each subset. We used one subset as the training data to build the model and evaluated the model's predictive accuracy on the other subset (i.e., the testing set). Because both training and testing data are comprised of unbiased ratings submitted by users without seeing any system prediction, the accuracy performance computed based on these initial ratings would provide us the upper bound of accuracy performance for each recommendation algorithm.

We then selected 25 random ratings from the set of 40 biased submissions for each user and used them as inputs to re-build the recommendation model. The model's predictive accuracy was evaluated on the same exact testing set (i.e., 25 unbiased ratings from each user). Next we adjusted these 25 biased ratings using either the suggested global slope of $\alpha$ = 0.35 or user-specific adjustment slopes. When a global adjustment is used, the ratings submitted by all users are adjusted using the same global slope $\alpha$. In contrast, when a user-specific adjustment is used, we first estimate the regression slope $\alpha_u$ for each user $u$ based on the user's experimental data. If the estimated slope $\alpha_u$ is significant (i.e., $p$ <= 0.05), we use $\alpha_u$ to adjust the ratings provided by the given user. Each user hence has a unique adjustment slope. Finally, we computed the predictive accuracy using these 25 de-biased ratings as training data. The predictive accuracy of rating samples was computed for several well-known recommendation algorithms, including a simple global baseline heuristic (i.e., Baseline) [3], the matrix factorization approach (i.e., SVD) [8], and user- and item-based collaborative filtering algorithms (i.e., CF_User and CF_Item) [7,10].

In our experiment we repeated the above steps 30 times and extracted different random samples each time. We report the average accuracy performances based on unbiased, biased, and de-biased ratings in Table 1. The training data resulting in best performance for each recommendation method is indicated in boldface.

**Table 1. Mean predictive accuracy performance (measured in RMSE) based on different training ratings**

| Method | Initial (Unbiased) Ratings | Biased Ratings | De-Biased (Global Slope 0.35) | De-Biased (User-Specific Slopes) |
|---|---|---|---|---|
| SVD | **0.9572** | 0.9663 | 0.9955 | 0.9945 |
| CF_Item | **0.9749** | 0.9968 | 1.0450 | 1.0421 |
| CF_User | **0.9810** | 1.0025 | 1.0568 | 1.0536 |
| Baseline | **0.9521** | 0.9707 | 1.0048 | 1.0046 |

As seen in Table 1, the initial (unbiased) ratings provide the best accuracy performance for all recommendation algorithms, clearly demonstrating the advantage of unbiased ratings over biased ratings on recommender systems' predictive performance. Most of the accuracy comparisons in the table are statistically significant ($p$ < 0.05). The only two exceptions are the contrasts between de-biased ratings based on global and user-specific slopes for Baseline and SVD. The results suggest that the use of unbiased preference ratings as inputs indeed leads to significantly higher predictive accuracy of recommendation algorithms than the use of biased preference ratings. In addition, the de-biased ratings (adjusted based on either global or user-specific slopes) did not provide accuracy benefits. Adjusted ratings based on user-specific slopes lead to slightly better accuracy than ratings adjusted based on the global slope of $\alpha$ = 0.35. However, neither of the two post-hoc de-biasing adjustments was helpful in improving accuracy. These patterns are consistent across various popular recommendation algorithms described in Table 1.

In the second experiment, we explored different de-biasing slope values for user ratings and computed predictive accuracy on the entire rating dataset (as opposed to randomly chosen rating samples as in first experiment). Specifically, we took all 40 biased ratings submitted by users after seeing the system's predictions and adjusted these ratings using the post-hoc de-biasing rule. All of these 40 "de-biased" ratings were then used as training data to compute predictions using standard recommendation algorithms, and the predictive accuracy was evaluated on the initial 50 unbiased ratings. We varied the de-biasing slopes and explored both global and user-specific adjustments.

Fig 2 summarizes the predictive accuracy performance on ratings de-biased based on different adjustment slope parameters. When the slope value is equal to zero, it means no adjustment was made, i.e., the user's actual submitted ratings (biased) were used as training data for the recommendation algorithms. The vertical black line on the left side corresponds to the accuracy performance of various algorithms with these actual-submitted ratings (i.e., biased) as training data. In addition to exploring different global adjustment slopes, we also experimented with user-specific adjustments as indicated by the vertical black line on the right side.

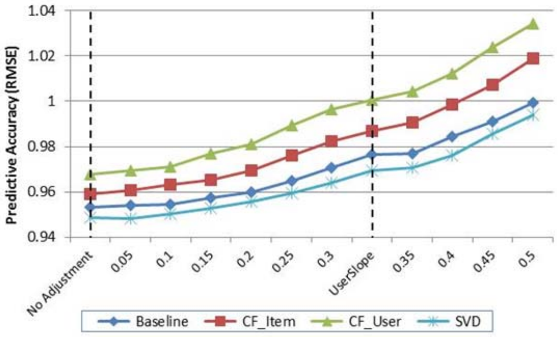


**Fig 2. Predictive accuracy of de-biased ratings, with varying adjustment slopes.**

Based on our experimental results, using users' actual submitted ratings (i.e., no adjustment) provided better accuracy performance than using de-biased ratings adjusted to any degree. As we increase the size of the global adjustment slope, the predictive accuracy performance estimated on test ratings decreases monotonically. Additionally, although the resulting accuracy of a user-specific adjustment is slightly better than that of the global slope of $\alpha$ = 0.35 suggested in prior research, the user-specific adjustment still did not yield better accuracy than no adjustment or small global adjustments. Overall, our experiment was unable to achieve any predictive accuracy improvements by de-biasing consumer ratings with either a global de-biasing rule based on a single slope parameter or the individual user-level rules based on user-specific slope parameters. We also conducted additional experiments with a variety of settings of post-hoc rating adjustment. For example, we introduced a tolerance threshold and only adjusted a submitted rating when it differs from the system's predicted rating by more than a certain amount (e.g., 0.5 stars). We also rounded de-biased ratings to various rating scales (e.g., to half stars, or to the first decimal place). We further experimented with adjusting only the positively biased ratings or only the negatively biased ratings to compare accuracy improvements. In addition, we empirically explored post-hoc rating de-biasing with a real-world movie rating dataset provided by Netflix [4].

However, based on our empirical explorations with these various post-hoc de-biasing methods, we have not been able to achieve any recommendation accuracy improvements by de-biasing consumer ratings with a global rule based on a single slope parameter (as demonstrated by Fig 2, we also explored other possible de-biasing slope values in addition to the empirically observed 0.35 value) or with a user-specific slope-based de-biasing rule. This indicates that, once the biased ratings are submitted, "reverse-engineering" is a difficult task. More specifically, while previous research was able to demonstrate that, *in aggregate*, there exist clear, measurable anchoring effects, it is highly likely that each *individual* anchoring effect (i.e., for a specific user/item rating) could be highly irregular – the biases could be user-dependent, item-dependent, context-dependent, and may have various types of other interaction effects. In fact, previous research provides some evidence to support such irregularity and situation-dependency. For example, prior studies observed symmetric (i.e., both positive and negative, equally pronounced) anchoring biases when they were aggregated across many items and asymmetric anchoring biases when they were tested on one specific item [1].

Therefore, an alternative approach to rating de-biasing would be to eliminate anchoring biases at rating-collection time through a carefully designed user interface. We discuss experiments with various interfaces in the next section.

## 4. APPROACH II: BIAS-AWARE INTERFACE DESIGN

The bias-aware interface design approach focuses on proactively preventing anchoring biases from occurring rather than trying to eliminate them after they have already occurred. We use a laboratory experiment to investigate various rating representation forms that may reduce anchoring effects at the rating collection stage. Besides the recommendation display, all other elements of the user interface were controlled to be equivalent across all experimental conditions. Our experiments explored *seven* different recommendation displays. Among them, four display designs were based on two main factors: (i) information representation (numeric vs. graphical ratings); and (ii) vagueness of recommendation (precise vs. vague rating values). Another two displays simulate popular star-rating representations used in many real-world recommender systems: stars-only and star along with a numeric rating. The seventh interface we explored was a binary design where only "thumbs up (down)" are displayed for high (low) predictions. Table 2 summarizes the seven rating representation options (i.e., Binary, Graphic-Precise, Graphic-Vague, Numeric-Precise, Numeric-Vague, Star-Numeric, and Star-Only).

### 4.1 Experiment Procedure

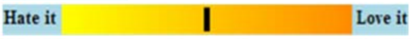
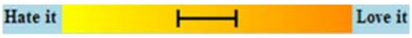
A database of 100 jokes was used for the study, with the order of the jokes randomized across participants. The jokes and the rating data for training the recommendation algorithm were taken from the Jester Online Joke Recommender System repository, a database of jokes and preference data maintained by the Univ. of California, Berkeley (http://eigentaste.berkeley.edu/dataset) [9]. The well-known item-based collaborative filtering technique was used to implement a recommender system that estimates users' preference ratings for the jokes [11]. The study was conducted at a behavioral research lab at a large North American university, and participants were recruited from the university's research participant pool. In total 287 people completed the study for a fixed participation fee.

Upon logging in, participants were randomly assigned to one of the seven treatment groups. Subjects in different treatment groups saw different displays of predicted rating. Examples of the display and number of participants in each treatment group are provided in Table 2.

The experimental procedure consisted of three tasks, all of which were performed using a web-based application on personal computers with dividers, providing privacy between participants.

**Task 1.** In the first task, each participant was asked to provide his/her preference ratings for 50 jokes randomly selected from the pool of 100 jokes. Ratings were provided using a scale from one to five stars with half-star increments, having the following verbal labels: * = "Hate it", ** = "Don't like it", *** = "Like it", **** = "Really like it", and ***** = "Love it". For each joke, we also asked participants to indicate whether they have heard the joke before. The objective of this joke-rating task was to capture joke preferences from the participants. Based on ratings provided in this task, predictions for the remaining unrated 50 jokes were computed.

**Table 2. Example displays of system predicted ratings**

| Group | N | Example Display of Predicted Rating |
|---|---|---|
| Binary | 40 | 👍 Thumb Up or 👎 Thumb Down |
| Graphic Precise | 40 |  |
| Graphic-Vague | 40 |  |
| Numeric-Precise | 40 | 3.0 (out of 5) |
| Numeric-Vague | 39 | between 2.6 and 3.4 (out of 5) |
| Star-Numeric | 45 | ★★★☆☆ 3.0 (out of 5) |
| Star-Only | 43 | ★★★☆☆ |

**Task 2.** In the second task, from the remaining unrated 50 jokes, participants were presented with 25 jokes (using 5 recommendation conditions with 5 jokes each) along with a rating recommendation for each joke and 5 jokes without a recommendation (as a control condition). The recommendation conditions are summarized below:

- *High-Artificial*: randomly generated high recommendation between 3.5 and 4.5 stars (drawn from a uniform distribution)
- *Low-Artificial*: randomly generated low recommendation between 1.5 and 2.5 stars (drawn from a uniform distribution)
- *High-Perturbed*: algorithmic predictions were perturbed upward by 1 star
- *Low-Perturbed*: algorithmic predictions were perturbed downward by 1 star
- *Accurate*: actual algorithmic predictions (i.e., not perturbed)
- *Control*: no recommendation to act as a control

We first selected 5 jokes for the High-Perturbed condition and 5 jokes for the Low-Perturbed condition. These 10 jokes were chosen pseudo-randomly to assure that the manipulated ratings would fit into the 5-point rating scale. Among the remaining jokes we randomly selected 15 jokes and assigned them to three groups: 5 to Accurate, 5 to High-Artificial and 5 to Low-Artificial. 5 more jokes were added as a control with no predicted system rating provided. The 25 jokes with recommendations were randomly ordered and presented on five consecutive webpages (with 5 displayed on each page). The 5 control jokes were presented on the subsequent webpage. Participants were asked to provide their preference ratings for all these 30 jokes on the same 5-star rating scale.

**Task 3.** As the third task, participants completed a short survey that collected demographic and other individual information for use in the analyses.

### 4.2 Analysis and Results

The Perturbed vs. Artificial within-subjects manipulation described above represents two different approaches to the study of recommendation system bias. The Artificial recommendations provide a view of bias that controls for the value ranges shown, manipulating some to be high and some low, while not accounting for individual differences in preferences in providing the recommendations. The Perturbed recommendations control for such possible preference differences, allowing a view of recommendation error effects. We analyze the results from each of these approaches separately. First, we test different rating presentations with *artificially* (i.e. randomly) generated recommendations (i.e., not based on users' preferences).

### *4.2.1 Artificial Recommendations*

Fig 3 presents a plot of the aggregate means of user-submitted ratings for each of the treatment groups when high and low artificial recommendations were provided. As can be seen in the figure, low artificial recommendations pull down user's preference ratings relative to the control, and the high artificial recommendations tend to increase user's preference ratings. As an initial analysis, for each rating display we performed pairwise *t*-tests to compare user submitted ratings after receiving high and low artificial recommendations. The *t*-test results are presented in Table 3.

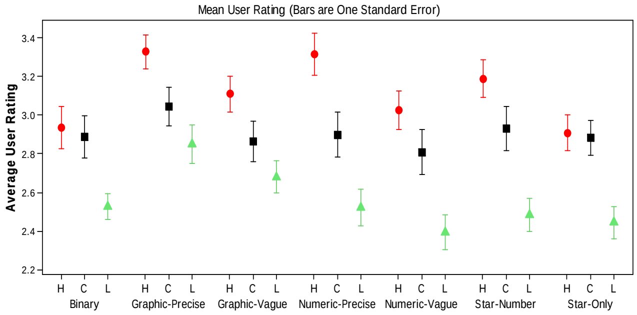


**Fig 3. Mean and standard deviation of user submitted ratings after receiving high artificial (High: red dot), low artificial (Low: green triangle), or no recommendations (Control: black square).**

**Table 3. Pair-wise comparisons of mean user rating difference for each rating display option using t-tests.**

| Rating Display | High − Low | High − Control | Low − Control |
|---|---|---|---|
| Binary | $0.408^{***}$ | 0.045 | $-0.363^{***}$ |
| Graphic-Precise | $0.478^{***}$ | $0.283^{**}$ | $-0.195^{*}$ |
| Graphic-Vague | $0.428^{***}$ | $0.245^{**}$ | $-0.183^{*}$ |
| Numeric-Precise | $0.793^{***}$ | $0.415^{***}$ | $-0.378^{***}$ |
| Numeric-Vague | $0.628^{***}$ | $0.215^{*}$ | $-0.413^{***}$ |
| Star-Numeric | $0.702^{***}$ | $0.258^{***}$ | $-0.444^{***}$ |
| Star-Only | $0.463^{***}$ | 0.026 | $-0.437^{***}$ |

* p < 0.05, ** p < 0.01, ** p < 0.001

All comparisons between High and Low conditions are significant across the seven rating representations (one-tailed *p*-value < 0.001 for all High vs. Low tests), showing a clear, positive effect of randomly-generated recommendations on consumers' preference ratings. All effect sizes are large (Cohen's *d* values range between 0.71 and 1.23). The control condition demonstrated intermediate preference ratings, showing a statistically significant difference from the both High and Low conditions for the majority of the rating display options. This analysis demonstrates that the anchoring bias of artificial recommendations exists in *all* rating displays examined in our experiment. In other words, we found that none of the seven rating display options could completely remove the anchoring biases generated by recommendations.

We further compare the *anchoring bias size* of different rating display options. We computed rating differences between High and Low conditions and performed one-way ANOVA to test the overall group difference. Our results suggest significant difference in effect sizes among different rating representations ($F(6, 280) = 2.24$, $p < 0.05$). Since the overall effect was significant, we next performed regression analysis to explore the difference in anchoring bias between different rating display options, while controlling for participant-level factors.

In our regression analysis, we created a panel from the data. The repeated-measures design of the experiment, wherein each participant was exposed to both high and low artificial recommendations in a random fashion, allows us to model the aggregate relationship between shown ratings and user's submitted ratings while controlling for individual participant differences. The standard OLS model using robust standard errors, clustered by participant, and using participant-level controls represents our model for the analysis.

$$UserRating_{ij} = b_0 + b_1(Group_i) + b_2(High_{ij}) + b_3(Group_i \times High_{ij}) + b_4(ShownRatingNoise_{ij}) + b_5(PredictedRating_{ij}) + b_6(Controls) + u_i + \varepsilon_{ij}$$

In the regression equation shown above, $UserRating_{ij}$ is the submitted rating for participant $i$ on joke $j$, $Group_i$ is the rating display option shown to participant $i$, $High_{ij}$ indicates whether the shown rating for participant $i$ on joke $j$ is a high or low artificial recommendations, $ShownRatingNoise_{ij}$ is a derived variable that captures the deviation between shown rating for participant $i$ on joke $j$ and the expected rating value in the corresponding condition. Specifically, it is computed by either subtracting 4.0 from the shown rating if it is in the high artificial condition or by subtracting 2.0 from the shown rating if it is in the low artificial condition. $PredictedRating_{ij}$ is the predicted recommendation star rating for participant $i$ on joke $j$, and $Controls$ is a vector of joke and consumer-related variables for participant $i$. The controls included in the model were the joke's funniness (average joke rating in the Jester dataset, continuous between 0 and 5), participant gender (binary), age (integer), whether they are native speakers of English (yes/no binary), whether they thought recommendations in the study were accurate (interval five point scale), whether they thought the recommendations were useful (interval five point scale), and their self-reported numeracy levels reflecting participants' beliefs about their mathematical skills as a perceived cognitive ability using a scale of four items developed and validated by prior research [6] (continuous between 4 and 24). The latter information was collected in order to check for possible relationships between individual's subjective numeracy capabilities and individual's susceptibility to anchoring biases due to numeric vs. graphical rating displays. As the study utilized a repeated-measures design with a balanced number of observations on each participant, to control for participant-level heterogeneity the composite error term ($u_i$ + $\varepsilon_{ij}$) includes the individual participant effect $u_i$ and the standard disturbance term $\varepsilon_{ij}$.

The Numeric-Precise rating display condition was chosen to be the baseline rating representation to compare with the other six options. We chose Numeric-Precise for two reasons. First it is a popular rating display used in many real-world recommender systems of large e-commerce websites such as Amazon, eBay and Netflix. Second, the Numeric-Precise rating display option was used by previous experiments in literature [1] and was found to lead to substantial anchoring biases in consumers' preference ratings. Therefore in our analysis we compare Numeric-Precise with other alternative rating display options to examine whether other rating representations can reduce the observed biases.

We ran three regression models with high artificial only, low artificial only, and both high and low artificial recommendations.

Note when only high or low recommendations were included for analysis, the model omitted the High variable and its related interaction terms. Table 4 presents the estimated coefficients and standard errors for the three regression models. All models utilized robust standard error estimates. The regression analysis controls for both participant and joke level factors as well as the participant's predicted preferences for the product being recommended.

**Table 4. Regression analysis on artificial recommendations (baseline: Numeric-Precise; dependent variable: UserRating)**

| | **Model 1**<br>High Only | **Model 2**<br>Low Only | **Model 3**<br>High&Low |
|---|---|---|---|
| Anchoring (High=1) | | | 0.794$^{***}$ |
| ShownRatingNoise | 0.350$^{***}$ | 0.249$^{**}$ | 0.289$^{***}$ |
| PredictedRating | 0.319$^{***}$ | 0.291$^{***}$ | 0.289$^{***}$ |
| ***Group*** | | | |
| Binary | -0.372$^{***}$ | 0.045 | 0.050 |
| Graphic-Precise | -0.045 | 0.314$^{**}$ | 0.301$^{**}$ |
| Graphic-Vague | -0.207$^{*}$ | 0.176 | 0.165 |
| Numeric-Vague | -0.238$^{**}$ | -0.073 | -0.073 |
| Star-Numeric | -0.149 | -0.007 | -0.015 |
| Star-Only | -0.392$^{***}$ | -0.020 | -0.036 |
| ***Interactions*** | | | |
| Binary×Anchoring | | | -0.427$^{***}$ |
| Graphic-Precise×Anchoring | | | -0.331$^{*}$ |
| Graphic-Vague×Anchoring | | | -0.365$^{*}$ |
| Numeric-Vague×Anchoring | | | -0.169 |
| Star-Numeric×Anchoring | | | -0.127 |
| Star-Only×Anchoring | | | -0.345$^{**}$ |
| ***Controls*** | | | |
| jokeFunniness | 0.618$^{***}$ | 0.539$^{***}$ | 0.587$^{***}$ |
| age | 0.005 | 0.000 | 0.003 |
| male | 0.114$^{*}$ | 0.009 | 0.063 |
| native | -0.127$^{*}$ | -0.002 | -0.067 |
| PredictionAccurate | 0.116$^{***}$ | 0.005 | 0.062$^{**}$ |
| PredictionUseful | 0.082$^{***}$ | -0.019 | 0.033 |
| Numeracy | 0.013 | 0.002 | 0.008 |
| Constant | -2.219$^{***}$ | -0.592 | -0.845$^{***}$ |
| $R^2$ within-subject | 0.0514 | 0.0397 | 0.1485 |
| $R^2$ between-subject | 0.5735 | 0.3548 | 0.5561 |
| $R^2$ overall | 0.2648 | 0.1388 | 0.2450 |
| $\chi^2$ | 476.82$^{***}$ | 155.74$^{***}$ | 768.28$^{***}$ |

$^{*}$ p < 0.05, $^{**}$ p < 0.01, $^{**}$ p < 0.001

Our analysis found randomly-generated recommendations displayed in Numeric-Precise format can substantially affect consumers' preference ratings, as indicated by significant coefficients for Anchoring and ShownRatingNoise in all three models. More importantly, we found significant negative interaction effects between multiple rating display options and anchoring (Model 3). The results clearly indicate that there are significant differences in anchoring biases between Numeric-Precise and other rating display options. Specifically, we observed that groups including Binary, Graphic-Precise, Graphic-Value, and Star-Only, when compared to Numeric-Precise, can generate much lower biases in consumers' preference ratings. All the corresponding interaction terms have negative coefficients with *p*-values smaller than 0.05. On the other hand, the interaction terms for Numeric-Vague and Star-Numeric were not significant, suggesting that these two display options lead to similar levels of anchoring biases as Numeric-Precise.

Overall, the Model 3 results suggest that, among all seven experimental rating display conditions, when randomly-assigned recommendations are presented in any non-numeric format (including Binary, Graphic-Precise, Graphic-Vague, Star-Only), they will generate much smaller anchoring biases compared to the same recommendations displayed in numeric formats such as Numeric-Precise, Numeric-Vague and Star-Numeric. In other words, the information representation of recommendations (e.g., numeric vs. non-numeric) largely determines the size of bias in consumers' preferences. Introducing vagueness to recommendations did not seem to reduce the anchoring bias when compared to the Numeric-Precise baseline (i.e., interaction between Numeric-Vague and anchoring is insignificant).

In a follow-up regression analysis (Table 5), we focused on four rating displays (i.e., Numeric-Precise, Numeric-Vague, Graphic-Precise, and Graphic-Vague) and similarly found the interaction between information presentation and anchoring (i.e., Numeric × Anchoring) was significant while the interaction between vagueness and anchoring (i.e., Precise × Anchoring) was not significant. This further confirms that the anchoring bias can be reduced by presenting recommendations in graphical forms rather than numeric forms. Anchoring bias, however, cannot be reduced by presenting the recommendations as vague rating ranges (as opposed to precise values).

**Table 5. Regression analysis on artificial recommendations, for Numeric/Graphic and Precise/Vague rating displays (dependent variable: UserRating)**

| | **Coefficient** |
|---|---|
| Anchoring (High=1) | 0.4027$^{***}$ |
| ShownRatingNoise | 0.2024$^{**}$ |
| PredictedRating | 0.2457$^{***}$ |
| Representation (Numeric=1) | -0.2667$^{**}$ |
| Vagueness (Precise=1) | 0.1100 |
| Numeric×Precise | 0.0046 |
| Numeric×Anchoring | 0.2562$^{**}$ |
| Precise×Anchoring | 0.1037 |
| ***Controls*** | |
| jokeFunniness | 0.7051$^{***}$ |
| age | 0.0017 |
| male | 0.0788 |
| native | -0.1013 |
| PredictionAccurate | 0.0687 |
| PredictionUseful | 0.0296 |
| Numeracy | 0.0146 |
| Intercept | -1.0531$^{**}$ |
| $R^2$ | 0.2500 |
| $\chi^2$ | 420.37$^{***}$ |

$^{*}$ p < 0.05, $^{**}$ p < 0.01, $^{**}$ p < 0.001

In addition, Model 1 focuses on high artificial recommendations (Table 4) and demonstrates significantly smaller anchoring biases for Binary, Graphic-Vague, Numeric-Vague and Star-Only displays, when compared to the Numeric-Precise display as the baseline. Model 2 focuses on low artificial recommendations and suggests that Graphic-Precise displays generated smaller biases compared to the baseline when recommendations were low. Therefore, another finding from Models 1 and 2 is that the "bias-reducing" effects of many rating display options can be highly asymmetric and depend on contextual factors such as the actual value of the recommendation.

Among the secondary factors, predicted consumer preferences, joke funniness, and perceived accuracy of recommendations all had consistently significant effects across all models. Therefore, controlling for these factors in the regression model was warranted.

### *4.2.2 Perturbed Recommendations*

As an extension to a more realistic setting and as a robustness check, we next examine whether anchoring biases generated by *perturbations* in real recommendations from an actual recommender system can be eliminated by certain rating display options. Recall that participants received recommendations that were perturbed either upward (High-Perturbed) or downward (Low- Perturbed) by 1 star from the actual predicted ratings. As a control, each participant also received recommendations without perturbations (Accurate). Consumers' submitted ratings for the jokes were adjusted for the predicted ratings in order to obtain a response variable on a comparable scale across subjects. Thus, the main response variable is the rating drift, which we define as:

$$RatingDrift = UserRating - PredictedRating$$

Fig 4 is a plot of the aggregate means of rating drift for each treatment group when recommendations were perturbed to be higher or lower or received no perturbation. As can be seen, the negative perturbations (Low, green triangle) lead to negative rating drifts and positive perturbations (High, red dot) lead to positive drifts in user ratings, while the accurate recommendations with no perturbation (Accurate, black square) lead to drifts around zero. For each rating display, we performed pairwise *t*-tests to compare user-submitted ratings after receiving high and low artificial recommendations. The *t*-test results are presented in Table 6.

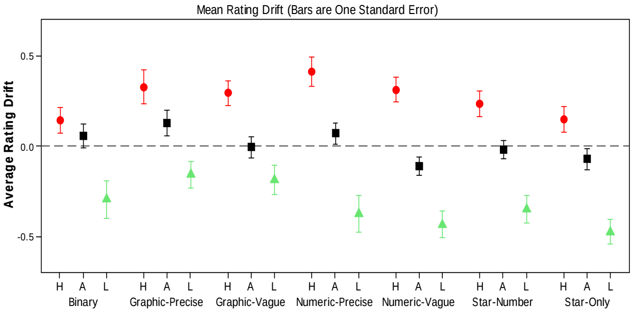


**Fig 4. Mean and standard deviation of user rating drift after receiving high perturbed (High: red dot), low perturbed (Low: green triangle), and non-perturbed recommendations (Accurate: black square).**

**Table 6. Pairwise comparisons of mean rating drift difference for each rating display option using t-tests.**

| Rating Display | High − Low | High − Accurate | Low − Accurate |
|---|---|---|---|
| Binary | $0.446^{***}$ | 0.104 | $-0.318^{**}$ |
| Graphic-Precise | $0.492^{***}$ | $0.292^{**}$ | $-0.187^{*}$ |
| Graphic-Vague | $0.482^{***}$ | $0.286^{**}$ | $-0.196^{*}$ |
| Numeric-Precise | $0.799^{***}$ | $0.491^{***}$ | $-0.297^{**}$ |
| Numeric-Vague | $0.770^{***}$ | $0.315^{**}$ | $-0.420^{***}$ |
| Star-Numeric | $0.599^{***}$ | $0.196^{**}$ | $-0.391^{***}$ |
| Star-Only | $0.671^{***}$ | $0.140^{*}$ | $-0.474^{***}$ |

* p < 0.05, ** p < 0.01, ** p < 0.001

All mean rating drift comparisons between High and Low perturbed conditions are significant for all rating display options (one-tailed *p*-value < 0.001 for all High vs. Low tests), showing a clear and positive anchoring bias of system recommendations on consumers' rating drift. Such anchoring biases exist in both High and Low perturbed conditions for the majority of the rating display options. The results clearly demonstrate that the anchoring effect of perturbed recommendations still exist in *all* rating display options investigated in our experiment. Hence, similar to the artificial groups, we found that none of the seven rating display options could completely remove the anchoring biases generated by perturbed real recommendations.

We next performed regression analysis to compare the size of anchoring bias across different rating display options, while controlling for participant-level factors. In our regression analysis, we created a panel from the data as each participant was exposed to both high and low perturbed recommendations in a random fashion. The standard OLS model using robust standard errors, clustered by participant, and participant-level controls represents our model for the analysis.

$$RatingDrift_{ij} = b_0 + b_1(Group_i) + b_2(High_{ij}) + b_3(Group_i \times High_{ij}) + b_4(PredictedRating_{ij}) + b_5(Controls) + u_i + \varepsilon_{ij}$$

In the above regression model, $RatingDrift_{ij}$ is the difference between submitted rating and predicted rating for participant *i* on joke *j*, $Group_i$ is the rating display option shown to participant *i*, $High_{ij}$ indicates whether the recommendation for participant *i* on joke *j* was perturbed upward or downward. *Controls* is the same vector of joke and consumer-related variables that was used in the previous regression analysis for artificial recommendations.

**Table 7. Regression analysis on perturbed recommendations (baseline: Numeric-Precise; dependent variable: RatingDrift)**

| | **Perturbed Recommendations**<br>High & Low |
|---|---|
| Anchoring (High = 1) | $0.777 (0.119)^{***}$ |
| PredictedRating | -0.128 (0.068) |
| ***Group*** | |
| Binary | 0.081 (0.143) |
| Graphic-Precise | 0.198 (0.126) |
| Graphic-Vague | 0.159 (0.131) |
| Numeric-Vague | -0.087 (0.126) |
| Star-Numeric | 0.023 (0.129) |
| Star-Only | -0.12 (0.126) |
| ***Interactions*** | |
| Binary×Anchoring | $-0.361 (0.169)^{*}$ |
| Graphic-Precise×Anchoring | -0.284 (0.168) |
| Graphic-Vague×Anchoring | $-0.302 (0.152)^{*}$ |
| Numeric-Vague×Anchoring | -0.042 (0.153) |
| Star-Numeric×Anchoring | -0.187 (0.157) |
| Star-Only×Anchoring | -0.139 (0.154) |
| ***Controls*** | |
| jokeFunniness | $0.236 (0.095)^{**}$ |
| age | 0.002 (0.005) |
| male | 0.016 (0.042) |
| native | -0.003 (0.052) |
| PredictionAccurate | 0.032 (0.03) |
| PredictionUseful | 0.011 (0.024) |
| Numeracy | 0.011 (0.007) |
| Constant | -1.241 (0.405) |
| $\boldsymbol{R^2}$ within-subject | 0.1493 |
| $\boldsymbol{R^2}$ between-subject | 0.0122 |
| $\boldsymbol{R^2}$ overall | 0.1214 |
| $\chi^2$ | $265.95^{***}$ |

* p < 0.05, ** p < 0.01, ** p < 0.001

The regression model used ordinary least squares (OLS) estimation and a random effect to control for participant-level heterogeneity. The Numeric-Precise rating display condition was again chosen to be the baseline rating representation to compare with the other six options. Table 7 summarizes the regression analysis of perturbed recommendations.

Consistent with what we found in the artificial conditions, interaction terms between anchoring and some non-numeric displays including Binary and Graphic-Vague were significantly negative. Thus, when recommendations were displayed in Binary and Graphic-Vague formats, they generated much smaller rating drifts from consumer's actual preference, when compared to the baseline Numeric-Precise display.

Similar to Table 5, we also performed a 2×2 analysis on the two main dimensions: representation (numeric vs. graphic) and vagueness (precise vs. vague) of the displayed recommendations. Our results in Table 8 confirm that presenting recommendations in numeric format can lead to much larger ratings shifts in consumer's preference ratings than presenting the same recommendations in graphical format. The vagueness of recommendation value, however, does not have significant influence on size of anchoring bias.

**Table 8. Regression analysis on perturbed recommendations, for Numeric/Graphic and Precise/Vague rating displays (dependent variable: RatingDrift)**

| | **Coefficient** |
|---|---|
| Anchoring (High=1) | $0.4680^{***}$ |
| PredictedRating | $-0.1969^{*}$ |
| Representation (Numeric=1) | $-0.2558^{**}$ |
| Vagueness (Precise=1) | 0.0415 |
| Numeric×Precise | 0.0843 |
| Numeric×Anchoring | $0.2648^{*}$ |
| Precise×Anchoring | 0.0304 |
| ***Controls*** | |
| jokeFunniness | 0.4008 |
| age | $0.0097^{**}$ |
| male | 0.0975 |
| native | -0.0381 |
| PredictionAccurate | 0.0779 |
| PredictionUseful | -0.0378 |
| Numeracy | 0.0228 |
| Intercept | $-1.8631^{*}$ |
| $R^2$ | $0.1497^{**}$ |
| $\chi^2$ | $420.37^{***}$ |

* p < 0.05, ** p < 0.01, ** p < 0.001

Overall, we observed that the real recommendations presented graphically can significantly lead to lower anchoring biases than real recommendations displayed in numeric forms (either as a precise number or as a numeric range). In addition, displaying real recommendations in binary format leads to much lower anchoring biases compared to recommendations in numeric forms (both numeric-precise and numeric-vague). Further, displaying real recommendations as a vague numeric range could not significantly reduce anchoring biases when compared to the benchmark approach of showing a precise value.

### *4.2.3 Discussion*

Using several regression analyses and controlling for various participant-level factors, we found that none of the seven rating display options completely removed the anchoring biases generated by recommendations. However, we observed that some rating representations were more advantageous than others. For example, we find that graphical recommendations can lead to significantly lower anchoring biases than equivalent numeric forms (either as a precise number or a numeric range). In addition, displaying recommendations in binary format leads to lower anchoring biases compared to recommendations in numeric forms.

## 5. CONCLUSIONS

This paper focuses on the problem of "de-biasing" users' submitted preference ratings and proposes two possible approaches to remove anchoring biases from self-reported ratings.

The first proposed approach uses post-hoc adjustment rules to systematically sanitize user-submitted ratings that are known to be biased. We ran experiments under a variety of settings and explored both global adjustment rules and user-specific adjustment rules. Our investigation explicitly demonstrates the advantage of unbiased ratings over biased ratings on recommender systems' predictive performance. We also empirically show that post-hoc de-biasing of consumer preference ratings is a difficult task. Removing biases from submitted ratings using a global rule or user-specific rule is problematic, most likely due to the fact that the anchoring effects can manifest themselves very differently for different users and items. This further emphasizes the need to investigate more sophisticated post-hoc de-biasing techniques and, even more importantly, the need to proactively prevent anchoring biases in recommender systems during rating collection.

Therefore, the second proposed approach is a user-interface-based solution that tries to minimize anchoring biases at rating collection time. We provide several ideas for recommender systems interface design and demonstrate that using alternative representations can reduce the anchoring biases in consumer preference ratings. Using a laboratory experiment, we were not able to completely avoid anchoring biases with any of the variety of carefully designed user interfaces tested. However, we demonstrate that some interfaces are more advantageous for minimizing anchoring biases. For example, using graphic, binary, and star-only rating displays can help reduce anchoring biases when compared to using the popular numerical forms.

In future research, another possible de-biasing approach might be through consumer education, i.e., to make consumers more cognizant of the potential decision-making biases introduced through online recommendations. This constitutes an interesting direction for future explorations.

## 6. REFERENCES

[1] Adomavicius, G., Bockstedt, J., Curley, S., and Zhang, J. 2013. "Do Recommender Systems Manipulate Consumer Preferences? A Study of Anchoring Effects," *Information Systems Research*, 24(4).

[2] Adomavicius, G., Bockstedt, J., Curley, S., and Zhang, J. 2012. "Effects of Online Recommendations on Consumers' Willingness to Pay," *Conference on Information Systems and Technology*. Phoenix, AZ.

[3] Bell, R.M., and Koren, Y. 2007. "Improved Neighborhood-Based Collaborative Filtering," *KDDCup'07*, San Jose, CA, USA, 7-14.

[4] Bennet, J., and Lanning, S. 2007. "The Netflix Prize," *KDD Cup and Workshop*, www.netflixprize.com.

[5] Cosley, D., Lam, S., Albert, I., Konstan, J.A., and Riedl, J. 2003. "Is Seeing Believing? How Recommender Interfaces Affect Users' Opinions," *CHI 2003 Conference*, Fort Lauderdale FL.

[6] Fagerlin, A., Zikmund-Fisher, B.J., Ubel, P.A., Jankovic, A., Derry, H.A., and Smith, D.M. 2007. "Measuring Numeracy without a Math Test: Development of the Subjective Numeracy Scale," *Medical Decision Making*, 27, 672-680.

[7] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. 1997. "Grouplens: Applying Collaborative Filtering to Usenet News," *Comm. the ACM*, 40, 77-87.

[8] Koren, Y., Bell, R., and Volinsky, C. 2009. "Matrix Factorization Techniques for Recommender Systems," *IEEE CS*, 42, 30-37.

[9] Lemire, D. 2005. "Scale and Translation Invariant Collaborative Filtering Systems," *Information Retrieval*, 8(1), 129-150.

[10] Sarwar, B., Karypis, G., Konstan, J.A., and Riedl, J. 2001. "Item-Based Collaborative Filtering Recommendation Algorithms," *Int'l WWW Conference*, Hong Kong, 285 - 295.

[11] Sarwar, B., Karypis, G., Konstan, J.A., and Riedl, J. 2001. "Item-Based Collaborative Filtering Recommendation Algorithms," *the 10th International WWW Conference*, Hong Kong, 285 - 295.

# Investigation of User Rating Behavior Depending on Interaction Methods on Smartphones

Shabnam Najafian
TU München
Boltzmannstr. 3
85748 Garching
Germany
s.najafian@tum.de

Wolfgang Wörndl
TU München
Boltzmannstr. 3
85748 Garching
Germany
woerndl@in.tum.de

Beatrice Lamche
TU München
Boltzmannstr. 3
85748 Garching
Germany
lamche@in.tum.de

## ABSTRACT

Recommender systems are commonly based on user ratings to generate tailored suggestions to users. Instabilities and inconsistencies in these ratings cause noise, reduce the quality of recommendations and decrease the users' trust in the system. Detecting and addressing these instabilities in ratings is therefore very important. In this work, we investigate the influence of interaction methods on the users' rating behavior as one possible source of noise in ratings. The scenario is a movie recommender for smartphones. We considered three different input methods and also took possible distractions in the mobile scenario into account. In a conducted user study, participants rated movies using these different interaction methods while either sitting or walking. Results show that the interaction method influences the users' ratings. Thus, these effects contribute to rating noise and ultimately affect recommendation results.

## Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces - Input devices and strategies, Interaction styles

## General Terms

Design, Experimentation, Human Factors.

## Keywords

user interfaces, recommender systems, rating behavior, user study, gestural interaction, mobile applications.

## 1. INTRODUCTION

In an age where information overload is becoming greater, generating accurate recommendations plays an increasingly important role in our everyday life. On the other hand, smartphones equipped with some set of embedded sensors provide an important platform to access data. Moreover, limitations in the user interface and the absence of suitable interaction methods makes it more and more difficult for mobile users to filter necessary information. Personalization and customization of the generated data helps deal with this information overload. Recommendation techniques are a subarea of intelligent personalizing and are seeking to obtain the users' preferences to allow personalized recommendations of tailored items. Recommender systems apply various recommendation techniques such as collaborative filtering, content-based, hybrid or context-aware recommendations, but all depend on acquiring accurate preferences (e.g. ratings) from users.

Preference acquisition is addressed via either explicit (user states his/her preferences), or implicit (system observes and analyzes the user's behavior) methods [3]. Because of the ambiguous nature of the implicit approach, explicit techniques are often employed to gather more reliable ratings from users to capture the users' preferences. Existing research usually assume stable ratings, i.e. the assumptions is that an available rating exactly reflect the user's opinion about an item. However, explicitely entered ratings may contain some level of noise. If this is the case, the system can not generate accurate recommendations. A lot of reasearch has been invested to increase the accuracy of recommendation algorithms, but relatively little to investigate the rating process.

This work explores one probable source of error in the rating process on smartphones which has not been considered much yet: the influence of input methods on the resulting ratings. Our specific scenario is a recommender system on a mobile device (smartphone). Mobile applications offer different input options for interaction including touchscreen and free-form gestures [7]. Touchscreen gestures allow users to tap on the screen, either using on-screen buttons or other interface elements, e.g. sliders. Free-form gestures do not require the user to actively touch the screen but to move the devices to initiate functions. In our previous work, we investigated which interaction methods are preferrable from a user's perspective for certain recommender system tasks [8].

The aim of this user study was to show that participants rate items differently depending on the applied input method. Errors that may occur due to re-rating were also taken into account to reduce other noises. We considered two situations in our study: the user were either sitting and concentrated on the task, or walking around and thus possibly distracted by the environment. We also measured the

*IntRS 2014*, October 6, 2014, Silicon Valley, CA, USA.

ease of use and effectiveness of our implementation based on an online survey.

The rest of the paper is organized as follows. We first outline related work. Next, we present our employed interaction methods and their implementation. In Section 4, we explain the setup and the results of our user study. Finally, we conclude the paper with a summary and a brief outlook.

## 2. RELATED WORK

Analyzing and characterizing noise in user rating of recommender systems in order to improve the quality of recommendations and therefore user acceptance is still an open research problem. Jawaheer et al. [3] recently surveyed methods to model and acquire user prefereces for recommender systems, distinguishing between explicit and implicit methods. They also mention that user ratings inherently have noise and cited some earlier studies. One earlier example is the study by Cosley et al. [2]. They investigated the influence of showing rating predictions when asking users to re-rate items. They found out that users applied their original rating more often when shown the predictions.

Amatriain et al. [1] attempted to quantify the noise due to inconsistencies of users in giving their feedback. They examined 100 movies from the Netflix Prize database in 3 trials of the same task: rating 100 movies via a web interface at different points in time. RMSE values were measured in the range of 0.557 and 0.8156 and four factors influencing user inconsistencies: 1) Extreme rating are more consistent were inferred, 2) Users are more consistent when movies with similar ratings are grouped together, 3) The learning effect on the setting improves the user's assessment, 4) The faster act of clicking on user's part does not yield more inconsistencies.

Nguyen et al. [5] performed a re-rate experiment consisting of 386 users and 38586 ratings in MovieLens. They developed four interfaces: one with minimalistic support that serves as the baseline, one that shows tags, one that provides exemplars, and another that combines the previous two features, to address two possible source of errors within the rating method. The first assumption is that users may not clearly recall items. Secondly, users may struggle to consistently map their internal preferences to the rating scale. The results showed that although providing rating support helps users rate more consistently, participants liked baseline interfaces because they perceived the interfaces to be more easy to use. Nevertheless, among interfaces providing rating support, the proposed one that provides exemplars appears to have the lowest RMSE, the lowest minimum RMSE, and the least amount of natural noise.

Our own previous work [8] aimed at mapping common recommender system methods - such as rating an item - to reasonable gesture and motion interaction patterns. We provided a minimum of two different input methods for each application function (e.g. rating an item). Thus, we were able to compare user interface options. We conducted a user study to find out which interaction patterns are preferred by users when given the choice. Our study showed that users preferred less complicated, easier to handle gestures over more complex ones.

Most of the existing studies do not take the mobile scenario into account, i.e. were not focussed on the interaction on mobile devices. When interacting with mobile devices, users may not be concentrated while being on the move or being distracted by the environment. Negulescu et al. [4] examined motion gestures in two specific distracted scenarios: in a walking scenario and in an eyes-free seated scenario. They showed that, despite somewhat lower throughput, it is beneficial to make use of motion gestures as a modality for distracted input on smartphones. Saffer [7] called these motion gestures free-form gestural interfaces which do not require the user to touch or handle them directly. Using these techniques the user input can be driven by the interaction with the space and can overcome some of the limitations of more classical interactions (e.g. via keyboards) on mobile devices [6].

In constrast to the existing work, we investigate the effect of user interaction methods on rating behavior on mobile devices (smartphones). We apply different input methods and interaction gestures in our interface to explore which ones decrease noise in the rating process. In the corresponding user study, we investigate the possible source of noise in rating results provoked by different input methods in the rating process. This study provides and analyzes the impacts of different interaction modalities on smartphones in the user giving feedback proceeding in details with the aim of overcoming rating result noise and enhancing recommender system quality.

## 3. INPUT METHODS IN THE TEST APPLICATION

To address this research question, we extend our previous work of a mobile recommendation application [8] . The scenario is a movie search and recommendation application that is similar the Internet Movie Database (IMDb) mobile application[1].

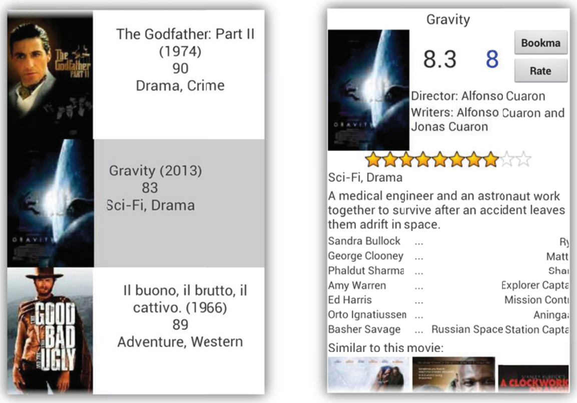
On the main screen, users can browse through the items to select a movie from the list (see Figure 1 (a)). Once they find a movie they are interested in, a single tap on that entry opens a new screen containing a more detailed description of the movie (Figure 1 (b)). Users can rate movies on a score from 1 (worst) to 10 (best) stars. To perfom the rating, they can choose one of the following three input methods:

1. On-screen button: users can rate a movie by selecting the "rate" on-screen button. The actual rating is performed by a simple tap on the 1 to 10 scale of stars (Figure 1 (b)).

2. Touch-screen gesture (*One-Finger_Hold_Pinch*) [8]: This rating uses a two-finger gesture. One finger is kept on the screen, while the second finger moves on the screen to increase or decrease or the rating stars respectively.

3. Free-form gesture (*Tilt*): Tilting means shifting the smartphone horizontally which is determined using it's gyroscope sensor. Shifting to the right increases the rating and shifting to the left decreases it. This rating is performed and saved without a single touch.

[1] see http://www.imdb.com/apps/?ref_=nb_app



The Godfather: Part II
(1974)
90
Drama, Crime

Gravity (2013)
83
Sci-Fi, Drama

Il buono, il brutto, il
cattivo. (1966)
89
Adventure, Western

(a)

Gravity

8.3 8

Director: Alfonso Cuaron
Writers: Alfonso Cuaron and
Jonas Cuaron

Sci-Fi, Drama

A medical engineer and an astronaut work together to survive after an accident leaves them adrift in space.

| | | |
|---|---|---|
| Sandra Bullock | ... | Ry |
| George Clooney | ... | Matt |
| Phaldut Sharma | ... | Sha |
| Amy Warren | ... | Explorer Capta |
| Ed Harris | ... | Mission Conti |
| Orto Ignatiussen | ... | Aninga |
| Basher Savage | ... | Russian Space Station Capta |

Similar to this movie:

(b)

Figure 1: (a) List of movies. (b) Details of movie screen.

# 4. USER STUDY

## 4.1 Gesture Investigation

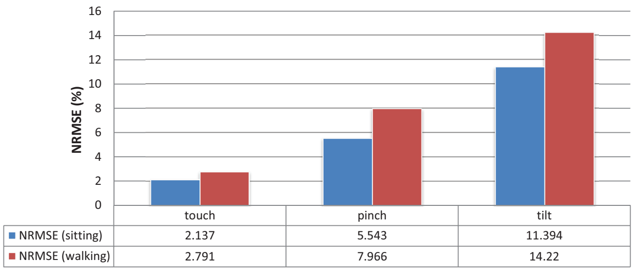
We conducted a user study to examine how a user's rating is influenced by the chosen input method. Another objective of this study was to evaluate the intuitiveness and efficiency of mapping input methods to some common recommender systems' functions in particular in a mobile scenario with a low attention span.

## 4.2 Procedure

At the beginning of each session, the task was explained to the users and the participants were asked to choose and rate 16 movies. The movies and corresponding ratings were recorded manually, not in the mobile application. Then, we handed the smartphones to the subjects and the users were asked to re-evaluate their intended rating for the same movies using the explained three input methods: on-screen button, touch-screen gesture (*One-Finger_Hold_Pinch*) and free-form gesture (*Tilt*) in two different scenarios. Participants had to rate four movies using each of the three different input methods, and then could freely choose a preferred method to rate another four items. Afterwards, the errors of users' in applying ratings were calculated based on their initial ratings.

The study investigated two scenarios. The first scenario was conducted while the user is sitting and thus can concentrate on the task. In the second scenario, the user is walking and thus not fully concentrated. We name these two scenarios *concentrated case* and *non-concentrated case*. Thus, each scenario consists of 16 ratings the subjects have to perform. Each rating process only takes a few seconds.

After having finished the experiment, the respondents were asked to fill out an online questionnaire. The questionnaire contained three main categories: prior knowledge, concentrated case (sitting scenario), non-concentrated case (walking scenario). For each part, we inquired the intuitiveness and user preference and also asked for the users' opinion on how much they thought the different interaction methods would affect their rating result. At the end, the interviewer asked the participants for suggestions of other gestures suiting the rating function better. The results of the evaluation are presented in the following section.



| | touch | pinch | tilt |
|---|---|---|---|
| ■ NRMSE (sitting) | 2.137 | 5.543 | 11.394 |
| ■ NRMSE (walking) | 2.791 | 7.966 | 14.22 |

Figure 2: NRMSE% for the three different interaction methods.

## 4.3 Participants and Apparatus

20 persons participated in the study, mostly students and researchers of the Munich University of Technology. The experiment was performed using a Samsung Galaxy S III mini smartphone running Android 4.1.

# 5. RESULTS

## 5.1 Evaluation Methodology

We evaluate the error for every interaction method for rating by calculating the root mean squared error (RMSE) (formula 1). In formula (1), n equals to the number of rated movies, $\hat{y}_t$ denotes the user's intended rating, which was elicited before the beginning of the test application was started as mentioned in 4.2. $y_t$ is equivalent to the user's rating which was obtained from the test application log.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (\hat{y}_t - y_t)^2}{n}} \tag{1}$$

Figure 2 shows the normalized root mean squared error percentage (NRMSE%) which were derived from the following formula:

$$NRMSE\% = \frac{RMSE}{9} \times 100 \tag{2}$$

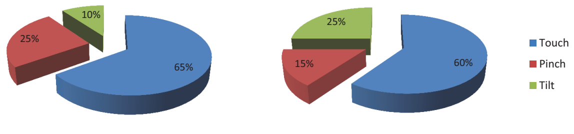
In this equation, 9 represents the maximum error since the values of ratings are in the range of [1;10].

## 5.2 Evaluation results

Figure 2 shows that the performance within the concentrated scenario is more precise than in the non-concentrated scenario, regardless which interaction method has been used. This was expected of course.

Among the different input methods, the on-screen button has the lowest error (with less than 3% very close to the intended rating), the touch-screen gesture has a medium accuracy, and the *Tilt* gesture has the highest NRMSE (more than 10%). Thus, the input method has a considerable effect on the resulting rating. In addition to that, the noise was lower towards the extreme ends of the rating scale.

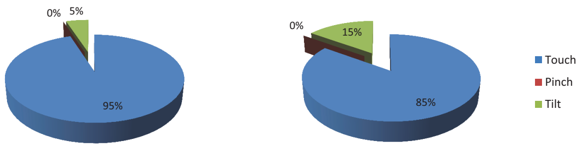
The low score for *Tilt* might be caused by our implementation of the gesture and better calibration might change the result. However, less than ideal implementations of interaction methods may be present in many mobile applications.



(a) sitting scenario

(b) walking scenario

**Figure 3: User preferred gesture.**



(a) sitting scenario

(b) walking scenario

**Figure 4: Intuitivity of different gestures.**

At the end of each session, the participants were asked to rate four movies using the preferred interaction method which was logged afterwards. The goal of this part was to determine which input method is preferred depending on the specific scenario (sitting or walking). Figure 3 illustrates the results. Our subjects preferred the on-screen button as input method in both scenarios. However, *Tilt* and *One-Finger_Hold_Pinch* were assessed differently depending on the scenario. Participants preferred *Tilt* in the non-concentrated (walking) scenario over *One-Finger_Hold_Pinch*, but vice versa in the concentrated (sitting) case.

We also asked the participants how intuitive they found the three input methods for rating on a scale from 1 to 5 with 5 being "very intuitive". Figure 4 illustrates which methods were rated as more intuitive by the participants. The results show that the *on-screen button* was rated as most intuitive in both scenarios, while *Tilt* being the second highest but still with minor percantage in the walking scenario. This may be due to the fact that the on-screen buttons are commonly used in mobile applications and most people are used to it.

In our survey, we defined an intuitive gesture as "being easy to learn and a pleasure to use". There is a difference in what the users found intuitive and what they actually preferred. Our participants found the common and simple on-screen button as most intuitive but 35% preferred the other options in the sitting scenario and 40% in the walking scenario, respectively.

## 6. CONCLUSION

Customer trust is the critical success factor for recommender systems. Since recommender systems frequently depend on the users' ratings, there is a need to reduce the users' rating errors in order to improve the reliablility of recommendations. In this study, a new source of errors in the rating process on mobile phones was investigated. We showed that rating results differ depending on the interaction method. Thus they distort the actual rating of the user, which can be improved by using more intuitive and easy to perform gestures. In our study, the results of the on-screen button appear to be more precise and reliable being near to the user's stated actual opinion.

We also demonstrate that free-form gestures such as *Tilt* are somewhat more desired in non-concentrated scenarios. When the environment is distracting, free-form gestures are more embraced by users even though, as a nature of non-concentrated situation, the results contain some noise. Due to the mobile phone's character, users are willing to be able to exploit their smartphones in situations which need less attention to perform an action, such as rating. To satisfy this requirement, a free-form gesture is applied in order to facilitate actions on mobile phones.

Regarding future work, introducing and studying more free-form gestures is desirable for recommender systems especially in non-concentrated scenarios. Moreover, people may get more and more used to performing free-form gestures. Since the detailed implementation and calibration of free-form gestures may have effect, an optimized *Tilt* implementation may reduce the error for this input method, in comparision to the result in our study. Investigating voice input would also be an interesting research topic as they do not require much effort and attention.

## 7. REFERENCES

[1] Amatriain, X., Pujol, J.M., and Oliver, N. 2009. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *Proc. of the 17th International Conference on User Modeling, Adaptation, and Personalization* (UMAP '09), 247–258. Springer.

[2] Cosley, D., Lam, S.K., Albert, I., Konstan, J.A., and Riedl, J. 2003. Is seeing believing?: how recommender system interfaces affect users' opinions. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '03). 585–592, ACM.

[3] Jawaheer, G., Weller, P., and Kostkova, P. 2014. Modeling user preferences in recommender systems: a classification framework for explicit and implicit user feedback In *Proc. of the 17th International Conference on User Modeling, Adaptation, and Personalization* (UMAP '09), 247–258. Springer.

[4] Negulescu, M., Ruiz, J., Li, Y., and Lank, E. 2012. Tap, swipe, or move: attentional demands for distracted smartphone input. In *Proc. of the Int. Working Conference on Advanced Visual Interfaces*, AVI '12, Capri Island, Italy, 173–180, ACM.

[5] Nguyen, T.T., Kluver, D., Wang, T.Y., Hui, P.M., Ekstrand, M.D., Willemsen, M.C., and Riedl, J. 2013. Rating support interfaces to improve user experience and recommender accuracy. In *Proc. of the 7th ACM Conference on Recommender Systems*, Hong Kong, China, ACM.

[6] Ricci, F. 2010. Mobile recommender systems. *J. of IT & Tourism*, 12, 3 (April 2010), 205–231.

[7] Saffer, D. 2008. *Designing Gestural Interfaces.* O'Reilly, Sebastopol.

[8] Woerndl, W., Weicker, J., Lamche, B. 2013. Selecting gestural user interaction patterns for recommender applications on smartphones. In *Proc. Decisions @ RecSys workshop*, 7th ACM Conference on Recommender Systems, Hong Kong, China, ACM.

# Interactive Explanations in Mobile Shopping Recommender Systems

Béatrice Lamche
TU München
Boltzmannstr. 3
85748 Garching, Germany
lamche@in.tum.de

Uğur Adıgüzel
TU München
Boltzmannstr. 3
85748 Garching, Germany
adiguzel@in.tum.de

Wolfgang Wörndl
TU München
Boltzmannstr. 3
85748 Garching, Germany
woerndl@in.tum.de

## ABSTRACT

This work presents a concept featuring interactive explanations for mobile shopping recommender systems in the domain of fashion. It combines previous research in explanations in recommender systems and critiquing systems. It is tailored to a modern smartphone platform, exploits the benefits of the mobile environment and incorporates a touch-based interface for convenient user input. Explanations have the potential to be more conversational when the user can change the system behavior by interacting with them. However, in traditional recommender systems, explanations are used for one-way communication only. We therefore design a system, which generates personalized interactive explanations using the current state of the user's inferred preferences and the mobile context. An Android application was developed and evaluated by following the proposed concept. The application proved itself to outperform the previous version without interactive and personalized explanations in terms of transparency, scrutability, perceived efficiency and user acceptance.

## Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Interaction styles, User-centered design*

## General Terms

Design, Experimentation, Human Factors.

## Keywords

mobile recommender systems, explanations, user interaction, Active Learning, content-based, scrutability

## 1. INTRODUCTION

In today's world, we are constantly dealing with complex information spaces where we are often having trouble to either find what we want or make decisions. Mobile recommender systems are addressing this problem in a mobile environment by providing their users with potentially useful suggestions that can support their decisions to find what they are looking for or discover new interesting things. Explanations of recommendations help users to make better decisions in contrast to recommendations without explanations while also increasing the transparency between the system and the user [8]. However, recommender systems employing explanations so far did not leverage their interactivity aspect. Touch based interfaces in smartphones reduce user effort while giving input. This can empower the interactivity for explanations. There are two main goals of this work. One is to study whether a mobile recommender model with interactive explanations leads to more user control and transparency in critique-based mobile recommender systems. Second is to develop a strategy to generate interactive explanations in a content-based recommender system. A mobile shopping recommender system is chosen as application scenario. The rest of the paper is organized as follows. We first start off with some definitions relevant for explanations in recommender systems and summarize related work. The next section explains the reasoning behind and the path towards a final mobile application, detailing the vision guiding the process. The user study evaluating the developed system is discussed in section 4. We close by suggesting opportunities for future research.

## 2. BACKGROUND & RELATED WORK

An important aspect of explanations is the benefit they can bring to a system. Tintarev et al. define the following seven goals for explanations in recommender systems [8]: 1. *Transparency* to help users understand how the recommendations are generated and how the system works. 2. *Scrutability* to help users correct wrong assumptions made by the system. 3. *Trust* to increase users' confidence in the system. 4. *Persuasiveness* to convince users to try or buy items and enhance user acceptance of the system. 5. *Effectiveness* to help users make better decisions. 6. *Efficiency* to help users decide faster, which recommended item is the best for them and 7. *Satisfaction* to increase the user's satisfaction with the system. However, meeting all these criteria is unlikely, some of these aims are even contradicting such as persuasiveness and effectiveness. Thus, choosing which criteria to improve is a trade-off. Explanations might also differ by the degree of personalization. While non-personalized explanations use general information to indicate the relevance of a recommendation, personalized explanations clarify how

*IntRS 2014*, October 6, 2014, Silicon Valley, CA, USA.

a user might relate to a recommended item [8].

Due to the benefits of explanations in mobile recommender systems, a lot of research has been conducted in this context. Since our work focuses on explanations aiming at improving transparency and scrutability in a recommender system, we investigated previous research in these two areas.

The work of Vig et al. [9] separates justification from transparency. While transparency should give an honest statement of how the recommendation set is generated and how the system works in general, justification can be refrained from the recommendation algorithm and explain why a recommendation was selected. Vig et al. developed a web-based **Tagsplanations** system where the recommendation is justified using relevance of tags. Their approach, as the authors noted, lacked the ability to let users override their inferred tag preferences.

Cramer et al. [3] applied transparent explanations in the web-based **CHIP** (Cultural Heritage Information Personalization) system that recommends artworks based on the individual user's ratings of artworks. The main goal of the work was to make the criteria more transparent the system uses to recommend artworks. It did so by showing the users the criteria on which the system based its recommendation. The authors argue that transparency increased the acceptance of the system.

An interesting approach to increase scrutability has been taken by Czarkowski [4]. The author developed **SASY**, a web-based holiday recommender system which has scrutinization tools that aim not only to enable users to understand what is going on in the system, but also to let them take control over recommendations by enabling them to modify data that is stored about them.

**TasteWeights** is a web-based social recommender system developed by Knijnenburg et al. [5] aiming at increasing inspectability and control. The system provides inspectability by displaying a graph of the user's items, friends and recommendations. The system allows control over recommendations by allowing users to adjust the weights of the items and friends they have. The authors evaluated the system with 267 participants. Their results showed that users appreciated the inspectability and control over recommendations. The control given via weighting of items and friends made the system more understandable. Finally, the authors concluded that such interactive control results in scrutability.

Wasinger et al. [10] apply scrutinization in a mobile restaurant recommender system named **Menu Mentor**. In this system, users can see the personalized score of a recommended restaurant and the details of how the system computed that score. However, users can change the recommendation behavior only by critiquing presented items via meal star ratings, no granular control over meal content is provided. A conducted user study showed that participants perceived enhanced personal control over given recommendations.

In summary, although previous research focused on increasing either scrutability or transparency in recommender systems, no research was conducted on how interactive explanations can increase transparency as well as scrutability in mobile recommender systems.

# 3. DESIGNING THE PROTOTYPE

Our system aims at offering shoppers a way to find nearby shopping locations with interesting clothing items while also supporting them in decision making by providing interactive explanations. Mobile recommender systems use a lot of situational information to generate recommendations, so it might not always be clear to the user how the recommendations are generated. Introducing transparency can help solving this problem. However, mobile devices require even more considerations in the design and development (e.g. due to the small display size). Thus, these should also be taken into account when generating transparent explanations. Moreover, the explanation framework should generate textual explanations that make it clear to the user how her preferences are modeled. In order not to bore the user, explanations must be concise and include variations in wording. Furthermore, introducing transparency alone might not be enough because users often want to feel in control of the recommendation process. The explanation goal scrutability addresses this issue by letting users correct system mistakes. There have been several approaches to incorporate scrutable explanations to traditional web-based recommender systems. However, more investigation is required in the area of mobile recommender systems. First of all, the system should highlight the areas of textual explanations that can be interacted with. Second, the system should allow the user to easily make changes and get new recommendations. While transparent and scrutable explanations are the main focus of this work, there are also some side goals, such as satisfaction and efficiency.

## 3.1 The Baseline

*Shopr*, a previously developed mobile recommender system serves as the baseline in our user study [6]. The system uses a conversation-based Active Learning strategy that involves users in ongoing sessions of recommendations by getting feedback on one of the items in each session. Thus, the system learns the user's preferences in the current context. An important point is that the system initially recommends very diverse items without asking its users to input their initial preferences. After a recommendation set is presented, the user is expected to give feedback on one of the items in the form of *like* or *dislike* over item features (e.g. price of the item or color) and can state which features she in particular *likes* or *dislikes*. In case the user submitted a positive feedback, using the *refine* algorithm shows more similar items. Otherwise, the system concludes a negative progress has been made and *refocuses* on another item region and shows more diverse items. The algorithm keeps the previously critiqued item in the new recommendation set in order to allow the user to further critique it for better recommendations. The explanation strategy used in this system is very simple. An explanation text is put on top of all items, which tries to convey the current profile of the user's preferences. It allows the user to observe the effect of her critiques and to compare the current profile against the actually displayed items. An example for such an explanation text is *"avoid grey, only female, preferably shirt/dress"*.

## 3.2 How Explicit Feedback Affects Weights

The modeling of the user's preferences is an important part of the proposed explanation generation strategy and feedback model and is adapted from the approach of *Shopr* [6], described in the *Baseline* section. It is modeled as a search query $q$ with weights for values of features (e.g. *red* is a possible value of the feature *color*). For each feature, there is a weight vector that allows the prioritization of one feature value over another. A query $q$ for a user looking for only red dresses from open shops in 2000m reach would look like this (we here assume that each item has only the two features 'color' and 'type'):

$$\begin{aligned} & q = ((distance \leq 2000m) \wedge (time\_open = \\ & now + 30min)), \{color_{red,blue,green}(1.0, 0, 0), \\ & type_{blouse,dress,trousers}(0, 1.0, 0)\} \end{aligned} \quad (1)$$

Our system uses two types of user feedback. One of them is by critiquing the recommended items on their features (which was already provided in the baseline system, see *section 3.1*). The other is by correcting mistakes regarding the user's preferences via explicit preference statement. Explanations are designed to be interactive, so that the user can state her actual preference over feature values after tapping on the explanation. If the user states interests on some feature values, a new value vector will be initialized for the query with all interested values being assigned equal weight summing to 1.0 and the rest having 0.0 weight. That means that the system will focus on the stated feature values, whereas the other values will be avoided. For example if a user interacts with the explanation associated with the query presented in *equation 1* and states that she is actually only interested in blue and green, then the resulting new weight vector would look like the following (assuming that we only distinguish between three colors) which will influence the search query and thus the new recommendations:

$$\begin{aligned} & feedback_{positive}(blue, green) : \\ & color_{red,blue,green}(0.0, 0.5, 0.5) \end{aligned} \quad (2)$$

## 3.3 Generating Interactive Explanations

The main vision behind interactive explanations is to use them not only as a booster for transparency and understandability of the recommendation process but also as an enabler for user control. In order to explain the current state of the user model (which stores the user's preferences) and the reasoning behind recommendations, two types of explanations are defined: *recommendation-* and *preference* explanations.

### 3.3.1 Interactive Recommendation Explanations

*Recommendation explanations* are interactive textual explanations. Their first aim is to justify why an item in the recommendation set is relevant for the user. Second, they let the user make direct changes to her inferred preferences. The generation is based on the set of recommended items, the user model and the mobile context.

*Argument Assessment.*

Argument assessment is used to determine the quality of every possible argument about an item. The argument assessment method is based on the method described in [1] . It uses Multi-Criteria Decision Making Methods (MCDM) to assess items $I$ on multiple decision dimensions $D$ (e.g. features that an item can have) by means of utility functions. Dimensions in the context of this recommender system are features and contexts. The method described in [1] uses four scores, which lay a good foundation for the method in this work. However, their calculations have to be adapted to the underlying recommendation infrastructure to produce meaningful explanations.

**Local score** $LS_{I,D}$ measures the performance of a dimension without taking into account how much the user values that dimension. Our system uses feature value weight vectors to represent both item features and features in a query, which represents the current preferences of the user. Local score of a feature is the scalar product of the weight vector (for that feature) in the query with respective weight vector in the item's representation. It is formalized as below, where $w_{I,D}$ represents the feature value weight vector for item dimension $D$ and $w_{Q,D}$ represents the feature value weight vector for query dimension $D$ and $n$ stands for the number of feature values for that dimension:

$$LS_{I,D} = \sum_{i=0}^{n-1} w_{I,D}(i).w_{Q,D}(i) \quad (3)$$

**Explanation score** $ES_{I,D}$ describes the explaining performance of a dimension. The weight for each dimension is calculated dynamically by using a function that decreases the effects of the number of feature values in each dimension. It is formalized as follows, where $length_{w_D}$ denotes the number of feature values in a specific dimension $D$ and $length_{total\_attribute\_values}$ the total number of feature values for all dimensions. Using the square root produced good results since it limits the effect of number feature values on the calculation of weights.

$$w_D = \sqrt{\frac{length_{w_D}}{length_{total\_attribute\_values}}} \quad (4)$$

With the following dynamically calculated weight for a dimension, explanation score of the dimension can be calculated by multiplying it with the local score of that dimension:

$$ES_{I,D} = LS_{I,D}.w_D \quad (5)$$

**Information score** $IS_D$ measures the amount of information provided by a dimension. The calculation of information score suggested by [1] is preserved as it already lays a good foundation to reason whether explaining an item from a given dimension provides a good value. So, it can be defined as follows where $R$ denotes the range of explanation scores for that dimension for all recommended items and $I$ denotes the information that dimension provides for an item:

$$IS_D = \frac{R + I}{2} \quad (6)$$

Range $R$ is calculated as the difference between the maximum and minimum explanation score for the given dimension for all recommended items, namely $R = max(ES_{I,D}) - min(ES_{I,D})$. Information $I$, however, is calculated quite differently from the strategy proposed by [1]. In their system, a dimension provides less and less information as the number of items to be explained from the same dimension increases. This does not apply to the context of the clothing recommender developed for this work. An item could still provide good information if not there are not so many items that can be explained from the same feature value. For instance, it is still informative to explain an item from the color blue; although another item is also explained by the same dimension (color) but from a different value, let's say green. Therefore, $I$ is calculated as a function of the size of recommendation set ($n$) and number of items in the set that has the same value for a dimension ($h$): $I = \frac{n-h}{n-1}$.

**Global score** $GS_I$ measures the overall quality of an item in all dimensions. It is the mean of explanation scores of all of its dimensions. The following formula demonstrates how it is formalized, where $n$ denotes the total number of all dimensions and $ES_{I,D_i}$ the explanation score of an item on $i_{th}$ dimension.

$$GS_I = \frac{\sum_{i=0}^{n-1} ES_{I,D_i}}{n} \tag{7}$$

The above-defined methods for calculating explanation and information scores are only valid for item features. Explanations should also include relevant context arguments. In order to support that, every context instance that is captured and used by the system in the computation of the recommendation set should also be assessed. The explanation score of a context dimension is calculated using domain knowledge. The most important values for the context gets the highest explanation score and it becomes lower and lower as the relevance of the value of the context decreases. For example, for location context, the explanation score is inversely proportional to the distance between the current location of the user and the shop where the explained item is sold. Explanation score gets higher as the distance gets lower. Information score is calculated with the same formula defined earlier for features $IS_D = \frac{R+I}{2}$, but Information $I$ slightly changes. As proposed earlier, it is calculated using the formula $I = \frac{n-h}{n-1}$, but in this case $h$ stands for the number of items with similar explanation score.

### *Argument Types.*

In order to generate explanations with convincing arguments, different argument aspects are defined by following the guidelines for evaluative arguments described in [2]. Moreover, the types of arguments described in [1] are taken as a basis. First of all, arguments can be either *positive* or *negative*. While positive arguments are used to convince the user to the relevance of recommendations, negative arguments are computed so that the system can give an honest statement about the quality of the recommended item. The second aspect of arguments is the type of dimension they explain, *feature* or *context*. Lastly, they can be *primary* or *supporting* arguments. Primary arguments alone are used to generate concise explanations. Combination of primary and supporting arguments are used to generate detailed explanations. We distinguish between five argument types: *Strong primary feature arguments*, *Weak primary feature arguments*, *Supporting feature arguments*, *Context arguments* and *Negative arguments*.

### *Explanation Process.*

The explanation process is based on the approach described in [1] but it is adapted to use the previously defined argument types. Different from the system in [1], explanations are designed to contain multiple positive arguments on features. Negative arguments are generated but only displayed when necessary by using a ramping strategy. *Figure 1* shows the process to select arguments. It follows the framework for explanation generation described in [2] as the process is divided into the selection and organization of explanation content and the transformation in a human readable form.

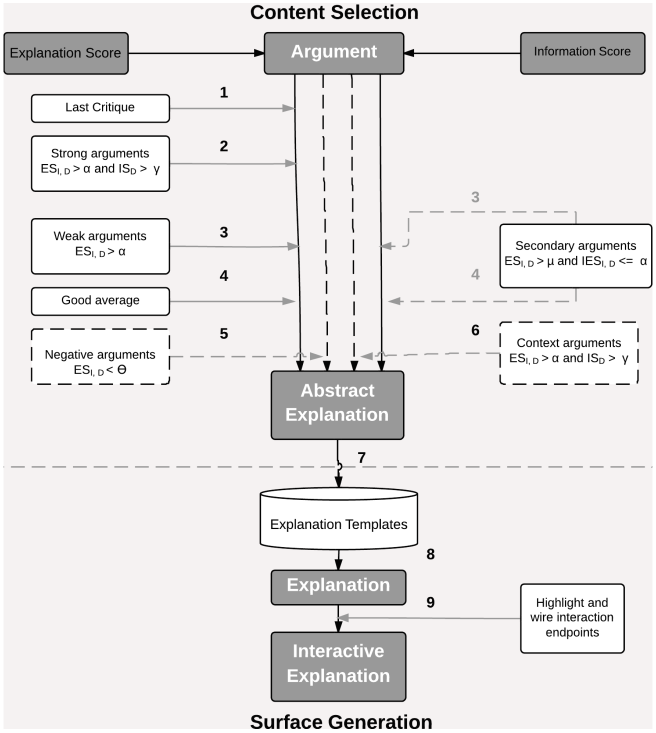


Figure 1: Generation of explanations.

**Content Selection.** The argumentation strategy selects arguments for every item $I$ separately. One or more primary arguments are selected first to help the user to instantly recognize why the item is relevant. There are four alternative ways to select the primary arguments (alternatives 1 to 4 in *figure 1*). First alternative is that the item is in the recommendation set because it was the last critique and it was carried (1). Another is that the system has enough *strong arguments* to explain an item (2). If there are not any strong arguments, the strategy checks if there are any *weak arguments* (3). In case there are one or more weak arguments, the system also adds supporting arguments to make the explanation more convincing. Finally, if there are no weak arguments too, then the item is checked if it is a good average by comparing its global score $GS_I$ to threshold $\beta$ (4). If so, similar to alternative (3), supporting arguments are also added to increase the competence of the explanation. Otherwise the strategy supposes that the recommended item is serendipitous and added to the set to explore the user's preferences. With one or more primary arguments, the system checks if there are any negative arguments and context arguments to add (5 and 6).

**Surface Generation.** The result of the content selection is an *abstract explanation*, which needs to be resolved to something the user understands. This is done in the surface generation phase. Various explanation sentence templates are decorated with either feature values or context values (7 and 8). Explanation templates are sentences with placeholders for feature and context values stored in XML format. The previously determined primary argument type

Table 1: Text templates for recommendation explanations.

| Text template | Example phrase |
|---|---|
| Strong argument | *"Mainly because you currently like X."* |
| Weak argument | *"Partially as you are currently interested in X."* |
| Supporting argument | *"Also, slightly because of your current interest in X."* |
| Location context | *"And it is just Y meters away from you."* |
| Average item | *"An average item but might be interesting for you."* |
| Last critique | *"Kept so that you can keep track of your critiques."* |
| Serendipity | *"This might help us discovering your preferences."* or *"A serendipitous item that you perhaps like."* |
| Negative argument | *"However, it has the following feature(s) you don't like: X, Y [...]."* |

is used to determine which type of explanation template to use. Feature values in the generated textual output are then highlighted and their interaction endpoints are defined (9). The resulting output is a textual explanation, highlighted in the parts where feature values are mentioned. They are interactive such that, after the user taps on the highlighted areas, she can specify what she exactly wants.

### *3.3.2 Interactive Preference Explanations*

*Preference explanations* have got two main goals. First, they aim to let the user inspect the current state of the system's understanding of the user's preferences. Second, they intend to let the user make direct changes to the preference. Two main types of preferences explanations are defined, *interactive textual explanations* and *interactive visual explanations.*

#### *Generating Textual Preference Explanations.*

The only input to textual preference explanation generation algorithm is the user model. For each dimension $D$ the algorithm can generate interactive explanations. Dimensions are features that an item can have. The algorithm distinguishes between four feature value weight vectors, indicating different user preferences: First, the user is indifferent to any feature value. Second, the user is only interested in a set of feature values. Third, the user is avoiding a set of feature values. And fourth, the user prefers a set of feature values over others.

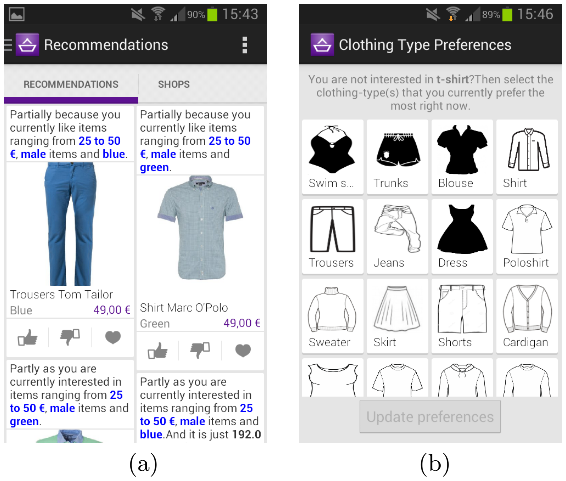
#### *Generating Visual Preference Explanations.*

Visual preference explanations are generated also by using the user model, more specifically by making use of the array of feature value weight vectors, which represents the user's current preferences. For each feature, there is already a feature value weight vector, which indicates the priorities of the user among feature values. All those weights are between 0.0 and 1.0 summing up to 1.0. They could be scaled to a percentage to generate charts showing the distribution of percentage of interests for feature values.

In order to generate charts, it is also required to determine with which color and description a feature value will be represented in a chart. In order to support that, a feature value

Table 2: Text templates for preference explanations.

| Text template | Example phrase |
|---|---|
| Only some values | *"You are currently interested only in X, Y [...]."* The word "only" in the text is emphasized in bold. |
| Avoiding some values | *"You are currently avoiding X, Y [...]."* The word "avoiding" is emphasized in bold. |
| Preferably some values | *"It seems, you currently prefer X, Y [...]."* |
| Indifferent to feature | *"You are currently indifferent to X feature".* |



Partially because you currently like items ranging from **25 to 50 €**, **male** items and **blue**.

Trousers Tom Tailor
Blue 49,00 €

Partly as you are currently interested in items ranging from **25 to 50 €**, **male** items and **green**.

Partially because you currently like items ranging from **25 to 50 €**, **male** items and **green**.

Shirt Marc O'Polo
Green 49,00 €

Partly as you are currently interested in items ranging from **25 to 50 €**, **male** items and **blue**.And it is just **192.0**

You are not interested in **t-shirt**?Then select the clothing-type(s) that you currently prefer the most right now.

Figure 2: Recommendation list (a) and explicit preference feedback screen (b).

appearing in the chart is modeled with its weights (scaled to a percentage), color and description in the user interface. *Figure 5* illustrates this chart representation.

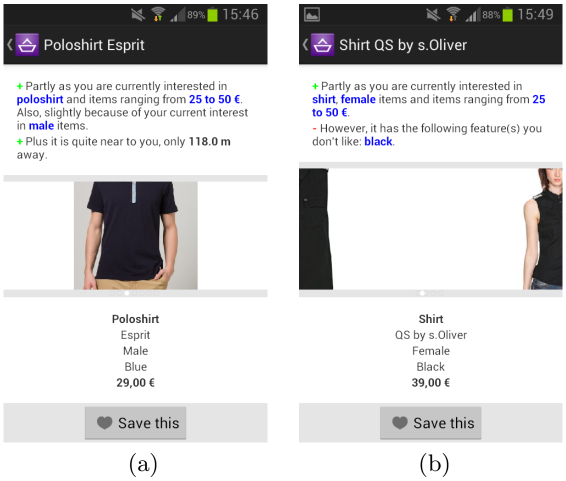
### *3.3.3 Using Text Templates Supporting Variation*

XML templates are used to generate explanation sentences for the different user preference types in English language. Those templates contain placeholders for feature and context values which are replaced during the explanation generation process. For *recommendation explanations*, there are a few sentence variations for almost every type of arguments. See *table 1* for examples of the different text templates for recommendation explanations. These templates can be used in combination with each other. For example, supporting arguments can support a weak argument. In such cases, argument sentences are connected using conjunctions.

Similar mechanism is also used for the *preference explanations.* However, to keep it simple, variation is not provided, as the number of features to explain is already limited. See *table 2* for selected examples of several text templates for preference explanations.

## 3.4 Interaction and Interface Design

The first issue was to clarify how to integrate the interaction process with textual explanations. It was envisioned to give the user the opportunity to tap on the highlighted



+ Partly as you are currently interested in **poloshirt** and items ranging from **25 to 50 €**. Also, slightly because of your current interest in **male** items.

+ Plus it is quite near to you, only **118.0 m** away.

**Poloshirt**
Esprit
Male
Blue
**29,00 €**

(a)

+ Partly as you are currently interested in **shirt**, **female** items and items ranging from **25 to 50 €**.

- However, it has the following feature(s) you don't like: **black**.

**Shirt**
QS by s.Oliver
Female
Black
**39,00 €**

(b)

Figure 3: Detailed information screens of items.

areas of the explanation text to state her actual preferences on a feature. This leads to a two-step process. First, the user sees an item with an explanation including highlighted words (highlighted words are always associated with a feature, see *figure 2a*) and taps on one of them (e.g. in *figure 2b*, "t-shirt" was tapped). Then the system directs the user to the screen where the user can make changes. In this second step, she specifies which feature values she is currently interested in. Lastly, the system updates the list of recommendations which completes a recommendation cycle. Note that the critiquing process and associated screens from the project *Shopr*, which is taken as a basis (see *section 3.1*) are kept in the developed system. Eventually, the interaction is a hybrid of critiquing and explicitly stating current preferences. On top of each explicit feedback screen, a text description of what is expected from the user is given.

Due to the applied ramping strategy mentioned in *section 3.3.1*, all extra arguments in explanations that are not important were not shown as explanations in the list of recommendations but in the screen where item details are presented. Tapping on an item picture accesses that screen. Here, the user can also browse through several pictures of an item by swiping the current picture from right to left (see *figure 3b*). In order to make it obvious for the user, the sentences with positive arguments always start with a green "+" sign. Negative arguments sentences, on the other hand, always start with a red "-" sign (see *figure 3*).

The next issue was to implement preference explanations, what we call *Mindmap feature*. Mindmap feature is the way that system explains its mental map about the preferences of the user. The overview screen for mindmap was designed to quickly show the system's assumptions about the user's current preferences. To keep it simple but yet usable, only textual explanations are used for each feature (see *figure 4b*). In order to make it easy for the user to notice what is important, the feature values used in the explanation text are highlighted. Moreover, every element representing a feature is made interactive. This lets the user access the explicit feedback screen to provide her actual preferences.

The user should also be able to get more detailed visual information for all the features. In order to achieve that, a different "drill down" screen for all screens was developed as part of the mindmap feature. *Figure 5* shows the mindmap detail screens for the clothing color feature. The user's preferences on feature values are represented as a chart. Every feature value is displayed as a different color in the charts. One of the most important features is that the highlighted parts of the explanation texts and the charts are interactive as well which lets the user access the explicit feedback screen to provide her actual preferences.

The full source code and resources for the Android app and the algorithm are available online[1].

# 4. USER STUDY

The main three goals of the evaluation are: First, to find out whether transparency and user control can be improved by feature-based personalized explanations supported by scrutable interfaces in recommender systems. Second, to find out whether side goals such as higher satisfaction are achieved and lastly to see whether other important system goals such as efficiency are not damaged.

## 4.1 Setup

The *test hardware* is a 4.3 inch 480 x 800 resolution Android smartphone (Samsung Galaxy S2) running the *Jelly Bean* version of the Android operating system (4.1.2).

Two *variants* of the system are put to the test. In order to refrain from the effects of different recommender algorithms, both variants use the same recommendation algorithm which uses diversity-based Active Learning [6]. Moreover, critiquing and item details interfaces are exactly the same. The difference lies in the explanations: The *EXP* variant refers to the proposed system, described in the previous section. In order to test the value of the developed explanations and scrutinization tools, a baseline (*BASE* variant) to compare against is needed (see *subsection 3.1*). The study is designed as within-subject to keep the number of testers at a reasonable size. Thus one group of people tests both variants. Which system is tested first was flipped in between subjects so that a bias because of learning effects could be reduced.

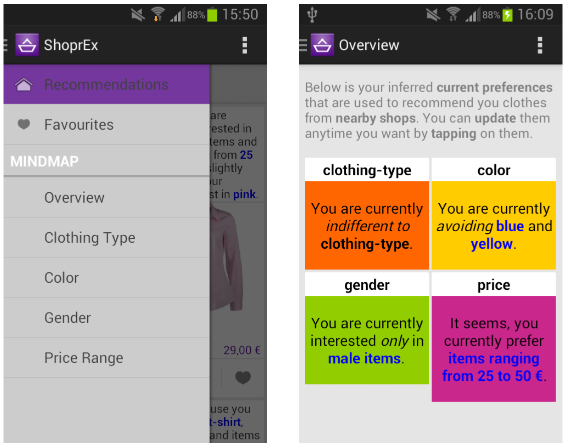
In order to create a realistic setup, it is necessary to generate a *data set* that represents real-world items. For that purpose, we developed a data set creation tool as an open-source project[2]. The tool crawls clothing items from a well-known online clothing retailer website. To keep the amount of work reasonable, items were associated with an id, one of 19 types of clothing, one of 18 colors, one of 5 brands, the price (in Euro), the gender (male, female or unisex) and a list of image links for the item. The resulting set is 2318 items large, with 1141 for the male and 1177 for the female gender.

For the study, *participants* of various age, educational background and current profession were looked for. Overall 30 people participated, whereas 33% of users were female and 67% were male.

The actual *testing procedure* used in the evaluation was structured as follows: We first asked the participants to provide background information about themselves, such as demographic information and their knowledge about mobile systems and recommender systems. Next, the idea of the

[1] `https://github.com/adiguzel/Shopr`

[2] `https://github.com/adiguzel/pickpocket`

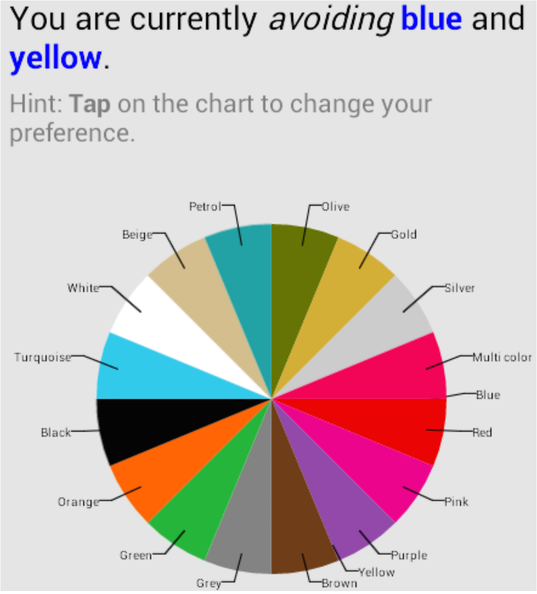


Below is your inferred **current preferences** that are used to recommend you clothes from **nearby shops**. You can **update** them anytime you want by **tapping** on them.

| clothing-type | color |
|---|---|
| You are currently *indifferent to* **clothing-type**. | You are currently *avoiding* **blue** and **yellow**. |
| **gender** | **price** |
| You are currently interested *only* in **male items**. | It seems, you currently prefer **items ranging from 25 to 50 €**. |

(a) (b)

Figure 4: Navigation Drawer (a) and Overview (b).



You are currently *avoiding* **blue** and **yellow**.

Hint: **Tap** on the chart to change your preference.

Figure 5: Mindmap detail screens for color.

system was introduced and the purpose of the user study was made clear. We chose a realistic scenario instead of asking users to find an item they could like:

***Task:*** *Imagine you want to buy yourself new clothes for an event in a summer evening. You believe that following type of clothes would be appropriate for this event: shirt, t-shirt, polo shirt, dress, blouse or top. As per color you consider shades of blue, green, white, black and red. You have a budget of up to € 100. You use the Shopr app to look for a product you might want to purchase.*

After introducing them to the task, users were given hands on time to familiarize themselves with the user interface and grasp how the app works. After selecting and confirming the choice for a product, the task was completed. Then testers were asked to rate statements about transparency, user control, efficiency and satisfaction based on their experience with the system on a five-point Likert scale (from 1, strongly disagree to 5, strongly agree) and offer any general feedback and observations. After having tested both variants, participants stated which variant they preferred and why that was the case.

## 4.2 Results

The testing framework applied in the user study is a subset of the aspects that are relevant for critiquing recommenders and explanations in critiquing recommenders. It follows the user-centric approach presented in [7]. The measured data is divided into four areas: transparency, user control, efficiency and satisfaction.

The means of the measured values for the most important metrics of the two systems, BASE denoting the variant using only simple non-interactive explanations, EXP the version with interactive explanations, are shown in *table 3*. Next to the mean the standard deviation is shown, the last column denoting the p-value of a one-tail paired t-test with 29 degrees of freedom (30 participants - 1).

In order to measure actual understanding after using a variant, users were asked to describe how the underlying recommendation system of that variant works. In general, almost all of the participants could explain for both recommenders that the systems builds a model of the user's preferences in each cycle and uses it to generate recommendations that can be interesting for the user.

On average, when asked if a tester understands the system's reasoning behind its recommendations, EXP performs better than BASE (mean average of 4.63 compared to 4.3 out of a 1-5 Likert scale). Further analysis suggests that the variant with interactive explanations (EXP) is perceived significantly more transparent than the variant with baseline explanations (one-tail t-test, p<0.05 with p=0.018).

Users were asked about the ease of telling the system what they want in order to measure the overall user control they perceived. Average rating of participants was better with EXP (4.33 versus 3.23). In a further analysis, EXP seemed significantly better in terms of perceived overall control than BASE (one-tail t-test, p<0.05 with p=0.0003).

When asked about the ease of correcting system mistakes, EXP performs a lot better than BASE (mean average of 4.36 compared to 3 out of a 1-5 Likert scale). Further analysis reveals that EXP is significantly better in terms of perceived scrutability than BASE (one-tail t-test, p<0.05 with p=0.6.08E-06).

Participants completed their task in average one cycle less using EXP than BASE (6.5 with EXP, 7.46 with BASE). However, one-tail t-test shows that EXP is not significantly better than BASE (p>0.05 with p=0.14).

The next part of measuring objective effort is done via tracking the time it took for each participant from seeing the initial set of recommendations until the target item was selected. On average BASE seems to be better with a mean session length of 160 seconds against 165 seconds. However, it was found not to be significantly more time efficient (one-tail t-test, p>0.05 with p=0.39). One reason for this could be that although EXP gives its users tools to update preferences over several features quickly, it has more detailed explanations. Thus, users spent more time with reading.

Users were asked about the ease of finding information and the effort required to use the system in order to get an idea about the system's efficiency. The participants' av-

Table 3: The means of some important measured values comparing both variations of the system.

| | BASE mean | stdev | EXP mean | stdev | p value |
|---|---|---|---|---|---|
| Perceived transparency | 4.3 | 0.70 | 4.63 | 0.49 | **0.018** |
| Perceived overall control | 3.23 | 1.04 | 4.33 | 0.71 | **0.0003** |
| Scrutability | 3 | 1.31 | 4.36 | 0.85 | **6.08E-06** |
| Cycles | 7.46 | 3.64 | 6.5 | 3.28 | 0.14 |
| Time consumption | 160 s | 74 | 165 s | 83 | 0.39 |
| Perceived efficiency | 3.43 | 1.13 | 4.33 | 0.75 | **0.0003** |
| Satisfaction | 3.76 | 0.85 | 4.43 | 0.56 | **0.0004** |

erage rating was better with EXP with 4.33 against 3.43 with BASE. Further analysis revealed that users perceived EXP significantly more efficient than BASE (one-tail t-test, $p<0.05$ with $p=0.0003$).

When inquired how satisfied participants were with the system overall, EXP performs better with 4.43 against 3.76. One-tail t-test suggests that this is a significant result ($p<0.05$ with $p=0.0004$).

Finally, participants were asked to pick a favorite from the two evaluated variants. 90% preferred the variant with interactive explanations (EXP) over the variant with simple non-interactive explanations (BASE), mostly because of the increased perception of control over recommendations.

## 5. CONCLUSION AND FUTURE WORK

This work investigated the development and impact of a concept featuring interactive explanations for Active Learning critique-based mobile recommender systems in the fashion domain. The developed concept proposes the generation of explanations to make the system more transparent while also using them as an enabler for user control in the recommendation process. Furthermore, the concept defines the user feedback as a hybrid of critiquing and explicit statements of current interests. A method is developed to generate explanations based on a content-based recommendation approach. The explanations are always made interactive to give the user a chance to correct possible system mistakes. In order to measure the applicability of the concept, a mobile Android app using the proposed concept and the explanation generation algorithm was developed. Several aspects regarding display and interaction design of explanations in mobile recommender systems are discussed and solutions to the problems faced during the development process are summarized. The prototype was evaluated in a study with 30 real users. The proposed concept performed significantly better compared to the approach with non-interactive simple explanations in terms of our main goals to increase transparency and scrutability and side goals to increasing perceived efficiency and satisfaction. Overall, the developed interactive explanations approach demonstrated the user appreciation of transparency and control over the recommendation process in a conversation-based Active Learning mobile recommender system tailored to a modern smartphone platform. Some changes, such as increasing the number of recommendations, skipping to the next list of recommendations without critiquing and having more item attributes for critiquing, could make the application even more appealing.

Future development may also include the creation of more complex recommendation scenarios to test the capability of the proposed concept even further. One can add more item features to critique and also take the user's mobile context (e.g. mood and seasonal conditions) into account during the recommendation process. Furthermore, future research might study the generation of interactive explanations for systems with rather complex recommendation algorithms. Interactive explanations might make adjustable parts of the algorithm transparent and allow the user to change them.

## 6. REFERENCES

[1] R. Bader, W. Woerndl, A. Karitnig, and G. Leitner. Designing an explanation interface for proactive recommendations in automotive scenarios. In *Proceedings of the 19th International Conference on Advances in User Modeling*, UMAP'11, pages 92–104, Berlin, Heidelberg, 2012. Springer-Verlag.

[2] G. Carenini and J. D. Moore. Generating and evaluating evaluative arguments. *Artif. Intell.*, 170(11):925–952, Aug. 2006.

[3] H. Cramer, V. Evers, S. Ramlal, M. Someren, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 18(5):455–496, Nov. 2008.

[4] M. Czarkowski. *A Scrutable Adaptive Hypertext*. PhD thesis, University of Sydney, 2006.

[5] B. P. Knijnenburg, S. Bostandjiev, J. O'Donovan, and A. Kobsa. Inspectability and control in social recommenders. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 43–50, New York, NY, USA, 2012. ACM.

[6] B. Lamche, U. Trottman, and W. Wörndl. Active learning strategies for exploratory mobile recommender systems. In *Proceedings of CaRR workshop, 36th European Conference on Information Retrieval*, Amsterdam, Netherlands, Apr 2014.

[7] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 157–164, New York, NY, USA, 2011. ACM.

[8] N. Tintarev and J. Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):399–439, Oct. 2012.

[9] J. Vig, S. Sen, and J. Riedl. Tagsplanations: Explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 47–56, New York, NY, USA, 2009. ACM.

[10] R. Wasinger, J. Wallbank, L. Pizzato, J. Kay, B. Kummerfeld, M. Böhmer, and A. Krüger. Scrutable user models and personalised item recommendation in mobile lifestyle applications. In *User Modeling, Adaptation, and Personalization*, volume 7899 of *Lecture Notes in Computer Science*, pages 77–88. Springer Berlin Heidelberg, 2013.

# If you liked Herlocker et al.'s explanations paper, then you might like this paper too

Derek Bridge
Insight Centre for Data Analytics
University College Cork, Ireland
derek.bridge@insight-centre.org

Kevin Dunleavy
School of Computer Science and IT
University College Cork, Ireland
kevdunleavy@gmail.com

## ABSTRACT

We present *explanation rules*, which provide explanations of user-based collaborative recommendations but in a form that is familiar from item-based collaborative recommendations; for example, "People who liked *Toy Story* also like *Finding Nemo*". We present an algorithm for computing explanation rules. We report the results of a web-based user trial that gives a preliminary evaluation of the perceived effectiveness of explanation rules. In particular, we find that nearly 50% of participants found this style of explanation to be helpful, and nearly 80% of participants who expressed a preference found explanation rules to be more helpful than similar rules that were closely-related but partly-random.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

## General Terms

Algorithms, Experimentation, Human Factors

## Keywords

Recommender Systems, Explanations

## 1. INTRODUCTION

An *explanation* of a recommendation is any content, additional to the recommendation itself, that is presented to the user with one or more of the following goals: to reveal how the system works (transparency), to reveal the data it has used (scrutability), to increase confidence in the system (trust), to convince the user to accept the recommendation (persuasion), to help the user make a good decision (effectiveness), to help the user make a decision more quickly (efficiency), or to increase enjoyment in use of the system (satisfaction) [11, 14]. The focus in this paper is effectiveness: explanations that help users to decide which item to consume.

*IntRS 2014*, October 6, 2014, Silicon Valley, CA, USA.

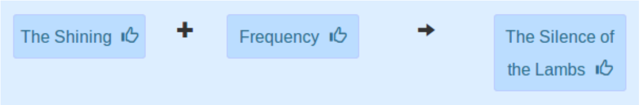


**Figure 1: An explanation rule**

The problem that we examine in this paper is how to produce effective explanations of user-based collaborative recommendations. It is relatively easy to explain the recommendations of *content-based recommenders*, e.g. by displaying meta-descriptions (such as features or tags) that the active user's profile and the recommended item have in common [10, 13]. *Item-based collaborative recommendations* are also amenable to explanation, e.g. by displaying items in the user's profile that are similar to the recommended item [8, 6]. *User-based collaborative recommendations*, on the other hand, are harder to explain. Displaying the identities of the active user's neighbours is unlikely to be effective, since the user will in general not know the neighbours; displaying their profiles is unlikely to be effective, since even the parts of their profiles they have in common with the active user will be too large to be readily comprehended.

It is possible to explain a recommendation using data other than that which the recommender used to generate the recommendation [2]. For example, a system could explain a user-based collaborative recommendation using the kind of data that a content-based recommender uses (features and tags), e.g. [9]. In our work, however, we try to preserve a greater degree of fidelity between the explanation and the operation of the recommender. Specifically, we generate the explanation from co-rated items on which the active user and her nearest-neighbour agree.

We propose an algorithm for making item-based explanations, also referred to as influence-style explanations [1]; for example, "People who liked *Toy Story* also like *Finding Nemo*". This style of explanation is familiar to users of amazon.com [6], for example. These are the kind of explanation most commonly produced by item-based collaborative recommenders. But we will show how to produce them in the case of user-based collaborative recommenders. The algorithm is adapted from one recently proposed to explain case-based classifiers [7]. It produces explanations in the form of *explanation rules*. The antecedent of an explanation rule characterizes a subset of the active user's tastes that are predictive of the recommended item, which appears in the consequent of the rule; see the example in Figure 1.

|  | *Alien* | *Brazil* | *Crash* | *Dumbo* | *E.T.* | *Fargo* |
|---|---|---|---|---|---|---|
| Ann | 2 | 4 | 1 | 2 |  | 4 |
| Bob | 5 | 4 |  | 1 |  | 5 |

**Table 1: A ratings matrix**

## 2. EXPLANATION ALGORITHM

We use a conventional user-based collaborative recommender of the kind described in [4]. Like theirs, our recommender finds the active user's 50 nearest neighbours using significance-weighted Pearson correlation; for each item that the neighbours have rated but the active user has not, it predicts a rating as the similarity-weighted average of deviations of neighbours' ratings from their means; it recommends the items with the highest predicted ratings.

Before presenting the explanation algorithm, we define some terms:

**Explanation partner:** The *explanation partner* is the member of the set of nearest neighbours who is most similar to the active user and who likes the recommended item. Often this will be the user who is most similar to the active user — but not always. In some cases, the most similar user may not have liked the recommended item: the recommendation may be due to the votes of other neighbours. In these cases, one of these other neighbours will be the explanation partner. It may appear that *recommendations* exploit the opinions of a set of neighbours (for accuracy), but *explanations* exploit the opinions of just one of these neighbours, the explanation partner. But this is not completely true. As we will explain below, the items included in the explanation are always members of the explanation partner's profile, but they are also validated by looking at the opinions of *all* other users (see the notions of coverage and accuracy below).

**Candidate explanation conditions:** Let $u$ be the active user and $v$ be the explanation partner; let $j$ be a co-rated item; and let $r_{uj}$ and $r_{vj}$ be their ratings for $j$. We define *candidate explanation conditions* as co-rated items $j$ on which the two users agree.

In the case of numeric ratings, we do not insist on rating equality for there to be agreement. Rather, we define agreement in terms of liking, indifference and disliking. For a 5-point rating scale, the candidate explanation conditions would be defined as follows:

$$\begin{aligned}&\text{candidates}(u, v) = \\ &\quad \{\text{likes}(j) : r_{uj} > 3 \wedge r_{vj} > 3\} \cup \\ &\quad \{\text{indiff}(j) : r_{uj} = 3 \wedge r_{vj} = 3\} \cup \\ &\quad \{\text{dislikes}(j) : r_{uj} < 3 \wedge r_{vj} < 3\}\end{aligned}$$

For example, the candidate explanation conditions for users Ann and Bob in Table 1 are

$$\{\text{likes}(\textit{Brazil}), \text{dislikes}(\textit{Dumbo}), \text{likes}(\textit{Fargo})\}$$

*Alien* does not appear in a candidate condition because Ann's and Bob's ratings for it disagree; *Crash* and *E.T.* do not appear in candidate conditions because neither of them is co-rated by Ann and Bob.

```
Input: user profiles $U$, recommended item $i$, active
       user $u$, explanation partner $v$
Output: an explanation rule for $i$
$R \leftarrow$ if _ then $i$;
$Cs \leftarrow$ candidates$(u, v)$;
while accuracy$(R) < 100 \wedge Cs \neq \{\ \}$ do
    $Rs \leftarrow$ the set of all new rules formed by adding
    singly each candidate condition in $Cs$ to the
    antecedent of $R$;
    $R^* \leftarrow$ most accurate rule in $Rs$, using rule coverage
    to break ties between equally accurate rules;
    if accuracy$(R^*) \leq$ accuracy$(R)$ then
        return $R$;
    $R \leftarrow R^*$;
    Remove from $Cs$ the candidate condition that was
    used to create $R$;
return $R$;
```

**Algorithm 1:** Creating an explanation rule

**Rule coverage:** A rule *covers* a user if and only if the rule antecedent is satisfied by the user's profile. For example, the rule in Figure 1 covers any user $u$ whose profile contains ratings $r_{u,\textit{TheShining}} > 3$ and $r_{u,\textit{Frequency}} > 3$, irrespective of what else it contains. Rule *coverage* is then the percentage of users that the rule covers.

**Rule accuracy:** A rule is *accurate* for a user if and only if the rule covers the user and the rule consequent is also satisfied by the user's profile. For example, the rule in Figure 1 is accurate for any user $u$ whose profile additionally contains $r_{u,\textit{TheSilenceoftheLambs}} > 3$. Rule *accuracy* is then the percentage of covered users other than the active user for whom the rule is accurate.

The algorithm for building an explanation rule works incrementally and in a greedy fashion; see Algorithm 1 for pseudocode. Initially, the rule has an empty antecedent, and a consequent that contains the recommended item $i$, written as 'if _ then $i$' in Algorithm 1. On each iteration, the antecedent is refined by conjoining one of the candidate explanation conditions, specifically the one that leads to the most accurate new rule, resolving ties in favour of coverage. This continues until either the rule is 100% accurate or no candidate explanation conditions remain.

## 3. EXPERIMENTS

We tested three hypotheses, the first using an offline experiment, the other two using a web-based user trial.

### 3.1 Practicability of explanation rules

The number of candidate explanation conditions can be quite large. If explanation rules are to be practicable, then the number of conditions that the algorithm includes in the antecedent of each explanation rule needs to be quite small.

**Hypothesis 1:** that explanation rules will be short enough to be practicable.

We ran the user-based collaborative recommender that we described at the start of the previous section on the MovieLens 100k dataset, and obtained its top recommendation for each user in the dataset. We then ran the explanation algorithm to produce an explanation rule that would explain

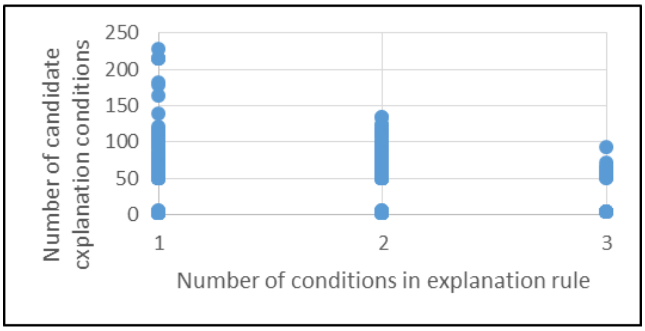


**Figure 2: Rule length**

the recommended item to that user. In Figure 2, we plot the number of candidate explanation conditions (vertical axis) against the number of these conditions that the algorithm includes in the rule (horizontal axis).

From the Figure, we see that the longest rules contained only three items in their antecedents. Not only that, but actually only 4% of the rules had three items in their antecedents; the other 96% were split nearly evenly between those having one and those having two items. We also see that the more candidates there are, the shorter the explanation rule tends to be. We have not investigated the exact reasons for this.

We repeated this experiment using a dataset with unary ratings to see what difference this might make. We took a LastFM dataset that contains artist play counts for 360 thousand users and 190 thousand artists.[1]. We converted play counts to unary ratings, i.e. recording 1 if and only if a user has played something by an artist. The results were very similar to those in Figure 2 (which is why we do not show them here), again with no rule having more than three items in its antecedent.

These are encouraging results for the practicability of explanation rules.

## 3.2 Effectiveness of this style of explanation

We designed a web-based user trial, partly inspired by the experiment reported in [5], drawing data from the MovieLens 1M dataset. Trial participants visited a web site where they progressed through a series of web pages, answering just three questions. An initial page established a context, essentially identical to the one in [5]:

> Imagine you want to go to the cinema but only if there is a movie worth seeing. You use an online movie recommender to help you decide. The movie recommender recommends one movie and provides an explanation.

First, we sought to elicit the perceived effectiveness of this *style* of explanation with the following hypothesis:

**Hypothesis 2:** that users would find explanation rules to be an effective style of explanation.

We showed participants an explanation rule for a recommendation and we asked them to rate its helpfulness on a 5-point scale. Specifically, we asked "Would this style of explanation help you make a decision?" with options Very unhelpful, Unhelpful, Neutral, Helpful, and Very helpful. Our wording differs from that used by [5]. They asked how likely the user would be to go and see the movie, with answers on a 7-point scale. Our wording focuses on explanation effectiveness (helpfulness in making a decision), whereas theirs focuses on persuasiveness.[2]

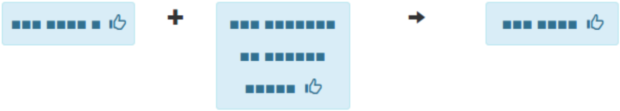


**Figure 3: A redacted explanation rule**

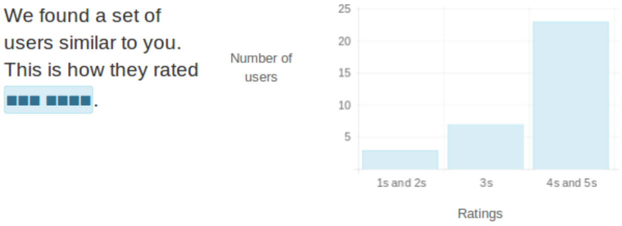

We found a set of users similar to you. This is how they rated [redacted].

**Figure 4: A redacted explanation in the style of [5]**

To encourage participants to focus on explanation *style*, we followed [5] in redacting the identity of the recommended movie. A participant's feedback is then not a function of the quality of the recommendation itself. For the same reasons, we obscured the identities of the movies in the antecedent of the explanation rule; see the example in Figure 3.

To obtain a 'yardstick', we also showed participants another explanation and asked them whether it too was helpful. For this purpose, we used the most persuasive explanation style from [5]. This explanation takes the form of a histogram that summarizes the opinions of the nearest neighbours. Figure 4 contains an example of this style of explanation (again with the recommended item redacted).

In the experiment, the software randomly decides the order in which it shows the two explanation styles. Approximately 50% of participants see and rate the explanation rule before seeing and rating the histogram, and the remainder see and rate them in the opposite order.

Prior to asking them to rate either style of explanation, users saw a web page that told them that we had obscured the movie titles, and we showed them an explicit example of a redacted movie title. We conducted a pilot run of the experiment with a handful of users before launching the real experiment. Participants in the pilot run did not report and difficulty in understanding the redacted movie titles or the redacted explanation rules.

We had 264 participants who completed all parts of the experiment. We did not collect demographic data about the participants but, since they were reached through our own contact lists, the majority will be undergraduate and postgraduate students in Irish universities.

Figure 5 shows how the participants rated explanation rules for helpfulness. Encouragingly, nearly 50% of participants found explanation rules to be a helpful or very helpful style of explanation (100 and 16 participants out of the 264,

[1] `mtg.upf.edu/node/1671`

[2] This is an observation made by Joseph A. Konstan in lecture 4-4 of the Coursera course *Introduction to Recommender Systems*, `www.coursera.org`.

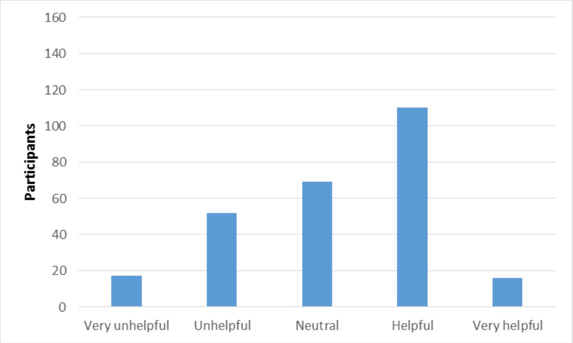


**Figure 5: Helpfulness of redacted explanation rules**

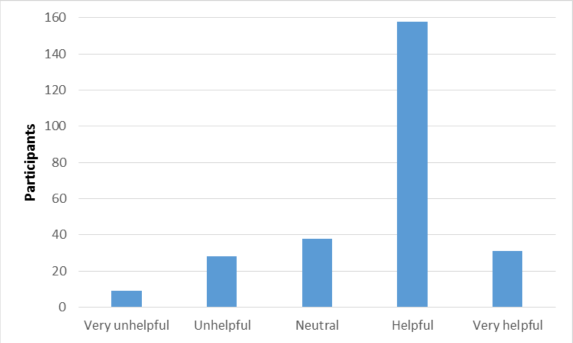


**Figure 6: Helpfulness of redacted histograms**

resp.); but about a quarter of participants found them neutral (69 participants), and a quarter found them unhelpful or very unhelpful (52 and 17, resp.). Figure 6 shows the same for the other style of explanation. Just over 70% of participants found this style of explanation to be helpful or very helpful (158 and 31 participants, resp.).

Note that we did not ask participants to *compare* the two styles of explanation. They are not in competition. It is conceivable that a real recommender would use both, either side-by-side or showing one of the two explanations by default and only showing the other to users who click through to a more detailed explanation page.

Furthermore, as the reader can judge by comparing Figures 3 and 4, any direct comparison of the results is unfair to the explanation rules since they have two levels of redaction (the recommended movie and the antecedents in the rules) whereas the histogram has just one (the recommended movie). As far as we can tell, there is no explanation style in [5] that would give comparable levels of redaction for a fair experiment.

For some readers, this may raise the question of why we showed participants the redacted histograms at all. The reason is to give a 'yardstick'. If we simply reported that nearly 50% of participants found explanation rules to be helpful or very helpful, readers would not know whether this was a good outcome or not.

From the results, we cannot confidently conclude that the hypothesis holds: results are not in the same ball-park as the 'yardstick'.[3] But we can conclude that explanation rules are a promising style of explanation: many users perceive them to be a helpful style of explanation, and they are therefore deserving of further study in a more realistic setting.

We note as a final comment in this subsection that the experiment reported in [1], which uses a very different methodology and no redaction of movie titles, found item-based explanations (there referred to as influence style explanations) to be better than neighbourhood style explanations.

## 3.3 Effectiveness of the selection mechanism

Next, we sought to elicit the perceived effectiveness of our algorithm's way of building explanation rules:

**Hypothesis 3:** that users would find the algorithm's selection of conditions in the antecedents of the rules (based on accuracy and coverage) to be better than random.

In the same web-based user trial, we showed the participants two rules side-by-side (the ordering again being determined at random). One rule was constructed by Algorithm 1. The other rule was constructed so as to have the same number of conditions in its antecedent, but these were selected at random from among the candidate explanation conditions. Note they are not wholly random: they are still candidate explanation conditions (hence they are co-rated items on which the user and explanation partner agree) but they are not selected using accuracy and coverage.

We asked participants to compare the two rules. They selected one of four options: the first rule was more helpful than the second; the second was more helpful than the first; the two rules were equally helpful; and they were unable to tell which was the more helpful ("don't know").

There was no redaction in this part of the experiment. It was important that participants judged whether the movie preferences described in the antecedents of the rules did support the recommended movie. Prior to asking users to rate the two explanation rules, users saw a web page that told them: that they would see a recommendation; that they should pretend that the recommended movie was one that they would like; that they would see two explanations; that movie titles would no longer be obscured; and that they should compare the two explanations for helpfulness. There are, of course, the risks that measuring effectiveness before consumption like this may result in judgements that overlap with persuasiveness, and that measuring perceived effectiveness is not as reliable as measuring something more objective [12].

Figure 7 shows the outcomes of this part of the experiment. We see that 32% found the explanation rule to be more helpful (85 participants) and only 10% (27 participants) found the partly-random rules to be more helpful. This means that, of those who expressed a preference (85 plus 27 participants), 76% preferred the explanation rules and only 24% preferred the partly-random rules. Furthermore, a two-tailed z-test shows the difference to be significant at the 0.01 level. This suggests that the algorithm does select candidate explanation conditions in a meaningful way.

However, 36% of participants found the rules to be equally helpful and 22% could not make a decision (95 and 57 participants resp.). This means, for example, that (again using

[3] For readers who insist on a comparison: using Very Unhelpful = 1, Unhelpful = 2, etc., the mean rating for the redacted explanation rules is 3.21 (st.dev. 1.03), the mean rating for the redacted histograms is 3.66 (st.dev. 0.94); and, using Welch's t-test, we reject at the 0.01 level the null hypothesis that there is no difference in the means.

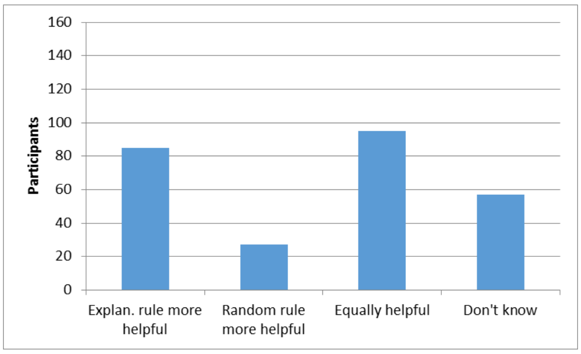


**Figure 7: Helpfulness of explanation rules compared with partly-random rules**

| Explanation rule | | Partly-random rule | |
|---|---|---|---|
| Accuracy | Coverage | Accuracy | Coverage |
| 91% | 2% | 56% | 15% |
| 83% | 1% | 68% | 4% |
| 76% | 11% | 42% | 33% |
| 25% | 3% | 20% | 13% |

**Table 2: Accuracy and coverage of pairs of rules**

a two-tailed z-test), there is no significant difference between the proportion who found explanation rules to be more helpful and the proportion who found the two rules to be equally helpful.

There are at least two reasons for this. The first is that the participant is required to put herself 'in the shoes' of another user. The recommendation and the rules are computed for a user in the MovieLens dataset, not for the person who is completing the experiment, who must pretend that she likes the recommendation. The person who completes the experiment may not know much, if anything, about the movies mentioned in the rules. This may be why the "don't know" option was selected so often.[4] The alternative was to require participants in the experiment to register with the recommender and to rate enough movies that it would be able to make genuine recommendations and build realistic explanation rules. We felt that this placed too great a burden on the participants, and would likely result in an experiment skewed towards users with relatively few ratings.

The second reason is that the partly-random rules are still quite good rules: they are considerably more meaningful than wholly-random rules. As Table 2 shows, one of the partly-random rules used in the experiment is nearly as accurate as its corresponding explanation rule. The partly-random rules also have high coverage because randomly selected movies are often popular movies. In our pilot run of the experiment, we had tried wholly-random rules, but they were so egregiously worse than their corresponding explanation rules that we felt that using them would prejudice the results of the real experiment. Ironically, the partly-random rules that we use instead perhaps include too many movies that are reasonable substitutes for the ones in their corresponding explanation rules, thus giving us much more equivocal results.

## 4. CONCLUSIONS

We have presented an algorithm for building explanation rules, which are item-based explanations for user-based collaborative recommendations. We ran an offline experiment and web-based user trial to test three hypotheses. We conclude that explanation rules are a practicable form of explanation: on two datasets no rule antecedent ever contained more than three conditions. We conclude that explanation rules offer a promising style of explanation: nearly 50% of participants found them to be helpful or very helpful, but the amount of redaction used in the experiment makes it hard to make firm conclusions about their effectiveness. Finally, we conclude that users do find the algorithm's selection of conditions for the rule antecedent to be better than random: just under 80% of participants who expressed a preference preferred the explanation rule to a partly-random variant. But results here are also partly confounded by the conditions of the experiment, where a participant has to put herself 'in the shoes' of another user.

Given the caveats about the limitations of the experiments, our main conclusion is that explanation rules are promising enough that we should evaluate them further, perhaps in a comparative experiment such as the one reported in [3] or in A/B experiments in a real recommender.

## 5. ACKNOWLEDGMENTS

This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289. We are grateful to Barry Smyth for discussions about the design of our experiment.

## 6. REFERENCES

[1] M. Bilgic and R. Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Procs. of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces*, 2005.

[2] G. Friedrich and M. Zanker. A taxonomy for generating explanations in recommender systems. *AI Magazine*, 32(3):90–98, 2011.

[3] F. Gedikli, D. Jannach, and M. Ge. How should I explain? A comparison of different explanation types for recommender systems. *Int. J. Hum.-Comput. Stud.*, 72(4):367–382, 2014.

[4] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In F. Gey et al., editors, *Procs. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237. ACM Press, 1999.

[5] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In W. Kellogg and S. Whittaker, editors, *Procs. of the ACM Conference on Computer Supported Cooperative Work*, pages 241–250. ACM Press, 2000.

[4] An on-screen note told the participant that she was able to click on any title to get some information about the movie. If she did, we fetched and displayed IMDb genres and a one-line synopsis for the movie. But we did not record how many users exploited this feature.

[6] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[7] D. McSherry. A lazy learning approach to explaining case-based reasoning solutions. In B. Díaz-Agudo and I. Watson, editors, *Procs. of the 20th International Conference on Case-Based Reasoning*, LNCS 7466, pages 241–254. Springer, 2012.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Procs. of the 10th International Conference on World Wide Web*, pages 285–295, 2001.

[9] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. MoviExplain: A recommender system with explanations. In *Procs. of the Third ACM Conference on Recommender Systems*, pages 317–320, 2009.

[10] N. Tintarev. Explanations of recommendations. In *Procs. of the First ACM Conference on Recommender Systems*, pages 203–206, 2007.

[11] N. Tintarev and J. Masthoff. Designing and evaluating explanations for recommender systems. In F. Ricci et al., editors, *Recommender Systems Handbook*, pages 479–510. Springer, 2011.

[12] N. Tintarev and J. Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4–5):399–439, 2012.

[13] J. Vig, S. Sen, and J. Riedl. Tagsplanations: Explaining recommendations using tags. In *Procs. of the 14th International Conference on Intelligent User Interfaces*, pages 47–56, 2009.

[14] M. Zanker. The influence of knowledgeable explanations on users' perception of a recommender system. In *Procs. of the Sixth ACM Conference on Recommender Systems*, pages 269–272, 2012.

# Choicla: Intelligent Decision Support for Groups of Users in the Context of Personnel Decisions

Martin Stettinger
Institute for Software Technology
Inffeldgasse 16b
A-8010, Graz, Austria
mstettinger@ist.tugraz.at

Alexander Felfernig
Institute for Software Technology
Inffeldgasse 16b
A-8010, Graz, Austria
afelfern@ist.tugraz.at

## ABSTRACT

Group recommendation technologies have been successfully applied in domains such as interactive television, music, and tourist destinations. Existing technologies are focusing on specific domains and do not offer the possibility of supporting different kinds of decision scenarios. The *Choicla* group decision support environment advances the state of the art by supporting decision scenarios in a domain-independent fashion. In this paper we present an overview of the *Choicla* environment and exemplify it's application in the context of personnel decisions.

## Categories and Subject Descriptors

D.2 [**Software and its engineering**]: Software creation and management; H.5 [**Information Interfaces and Presentation**]: Modelling Environments

## General Terms

Algorithms; Human Factors; Experimentation

## Keywords

Recommender Systems, Group Recommendation, Group Decision Making, Personnel Decisions

## 1. INTRODUCTION

Decisions in everyday life often come up in groups, for example, a decision about the destination for the next holidays or a decision about which restaurant to choose for a dinner. Knowledge about the preferences of other users in early phases of a decision process can lead to sub-optimal decision outcomes [12]. Missing explanations can lead to a lower level of trust in recommendations [2]. So-called anchoring effects [6] are responsible for decisions which are biased by the voting of the first preference-articulating person. If single persons have to take a decision in place of persons who are not available for a meeting, the outcome of the decision can also be negatively influenced. Decision processes are often not open in the sense that it is impossible to easily integrate new decision alternatives or change the individual preferences within the scope of a decision process - both aspects can lead to low-quality decision outcomes (see [13]). In many cases, the criteria for the decision remain unclear since there is no explanation of the outcome of ”the final decision”. All these mentioned threats can negatively influence the quality of group decisions.

One major goal of the *Choicla* environment is to facilitate group decision making and improve the overall quality of decision outcomes. The idea of this environment is to support definitions of different types of decision tasks in a domain-independent fashion while taking into account the above mentioned risk factors. In order to achieve this goal, *Choicla* builds upon different group recommendation algorithms [11] which are used for determining alternative solutions for the participants of a group decision process.

One example of the application of *Choicla* is to support groups of users in context of personnel decisions with the aim of achieving a more structured, fair, and transparent way of job interviews as well as to find the most suitable candidate for the job advertisement. Other typical scenarios for the application of *Choicla* technologies are the decision about which restaurant to select for a dinner or - in a scientific community - a decision regarding the selection of the destination of next year's conference.

The remainder of this paper is organized as follows. In Section 2 we provide insights to (1) the *Choicla* modelling process where participants can design decision tasks from scratch and (2) the intelligent management of already created decision apps. In the Section 3 we give an overview of the personnel decision scenario. We then discuss related & future work (Section 4) and thereafter conclude the paper (Section 5).

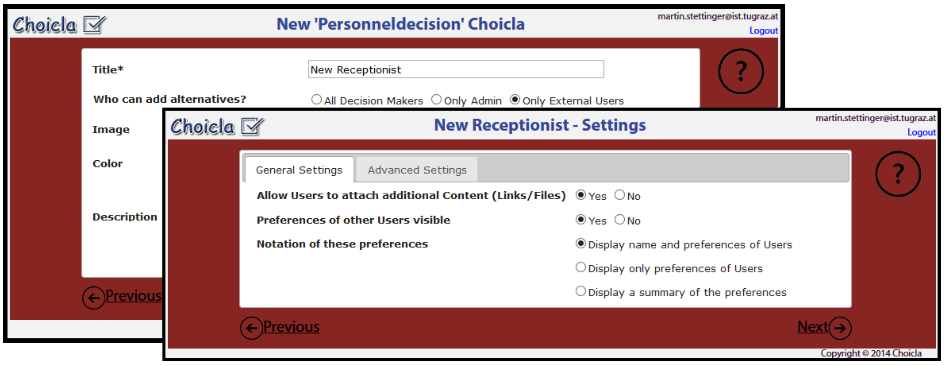
## 2. CHOICLA DECISION SUPPORT

Because decision scenarios differ from each other in their process design, a variety of parameters is needed to specify all relevant properties of a decision task. We will now discuss basic features (parameters) which can be configured (modelled) by the creator of a decision task. In this context we refer to the example features depicted in Figure 1.

### 2.1 Design of Decision Apps

Because decision scenarios differ from each other, some decision scenarios rely on a preselected decision heuristic that

*IntRS 2014*, October 6, 2014, Silicon Valley, CA, USA.



New 'Personneldecision' Choicla

Title* New Receptionist

Who can add alternatives? ○ All Decision Makers ○ Only Admin ● Only External Users

Image

Color

Description

New Receptionist - Settings

Allow Users to attach additional Content (Links/Files) ● Yes ○ No

Preferences of other Users visible ● Yes ○ No

Notation of these preferences ● Display name and preferences of Users ○ Display only preferences of Users ○ Display a summary of the preferences

**Figure 1: *Choicla*: definition of a decision task. Basic settings & further configurable features.**

defines the criteria for taking the decision, for example, a group decides to use *majority voting* for deciding about the next restaurant or cinema visit. The design of decision tasks (the underlying process) can be interpreted as a configuration problem (see [17]). The achieved flexibility of making the process design of a decision task configurable is needed due to the heterogeneity of decision problems. This way the *Choicla* components are organized as a kind of a software product line that is open in terms of the implementation (generation) of problem-specific decision applications.

**Explanations.** Explanations can have an important role in decision tasks since they are able to increase the trust of users in the outcome of a decision process [2]. When designing a decision task in *Choicla*, explanations can be selected as a feature of the decision process. If this feature is selected, the administrator of a decision task has to enter some explanatory text, if not, the entering of such a text remains just an option.

**Administration of Decision Alternatives.** The administration of decision alternatives within the scope of a decision task can be supported in different ways. *First*, only the initiator of a decision task is allowed to add alternatives – this could be desired if a person is interested in knowing the opinions of his/her friends about a concrete set of alternatives (e.g., alternative candidates for the next family car). Another related scenario are so-called ”Micro-Polls” where the initiator is only interested in knowing the preference distribution of a larger group of users. *Second*, in some scenarios it is important that all decision makers can add alternatives during the decision task by themselves – a common example of such a scenario is the group-based decision regarding a holiday destination or a hotel [7]. In such a context, each participant should be allowed to add relevant alternatives. The support of group-based personnel decisions can be seen as an example scenario of the *third* case (only external users can add alternatives) – in this context it should be possible that candidates apply for a certain job position (the application itself is interpreted as the addition of a new alternative to the decision task). The selection of the next conference location where proposers can submit their material is another example.

**Preference Visibility.** The scope ”private” allows only invited users to participate, i.e., the decision task is only accessible for invited users and not accessible for other users. If the scope is ”public”, the decision task is accessible for all users – this is typically the case in the context of so-called Micro-Polls. The decision quality can be influenced if the individual preferences of the other participants are visible during the decision process (see [3] and [7]). There exist decision scenarios where all participants profit from the knowledge of who entered which rating. If, for example, the decision task is to find a date for a business meeting it is essential to find a date where all managers can attend the meeting and therefore it is important to know the individual preferences of the participants. On the other hand there are decision scenarios where full preference visibility can lead to disadvantages for some participants but some kind of transparency of the individual preferences is helpful to achieve a reasonable decision. In such cases a summary of all given preferences is a feasible way to support decision makers (participants). A summary prevents the participants from statistical inferences to the individual preferences but still can help participants who are unsure about how to rate.

**Recommendation Support.** In the context of group decision tasks, an essential aspect is the aggregation function (recommendation heuristic). In a group decision process aggregation functions can help to foster consensus. User studies show that these functions also help to increase the degree of the perceived decision quality (see, for example [3]). Individual user preferences can be aggregated in many different ways and there exists no default heuristic which fits for every decision scenario. To provide a support for groups of users in different decision scenarios, the selection of recommendation heuristics is a key feature which has to be configured by the initiator of a decision task. Due to space limitations we only describe selected aggregation heuristics below. Masthoff [11] gives an overview of basic aggregation heuristics such as *Majority Vote (MAJ)*, *Average Vote (AVV)*, *Least Misery (LMIS)*, and *Most Pleasure (MPLS)* which are also available in the *Choicla* environment.
*Group Distance (GD)* (see Formula 1) returns the value $d$ as group recommendation which causes the lowest overall change of the individual user preferences where $eval(u, s)$ denotes the rating for a solution $s$ defined by user $u$.

$$GD(s) = minarg_{(d \in \{1..5\})}(\sum_{u \in Users} |eval(u, s) - d|) \quad (1)$$

*Ensemble Voting* can be seen as an example of a meta-aggregation function included in *Choicla*. Ensemble Voting (see Formula 2) determines the majority of the results of the individual voting strategies $H$ = {MAJ, AVV, LMIS, MPLS, GD} where $eval(h, s)$ denotes the result of an individual voting strategy for a solution $s$.

$$ENS(s) = maxarg_{(d\in\{1..5\})}(\#(\bigcup_{h\in H} eval(h, s) = d)) \qquad (2)$$

## 2.2 Choicla Decision Apps

After the design process has been finished, the creator of the decision task as well as all invited participants (after accepting the invitation) see a corresponding decision app directly on the personal home screen (see Figure 2).

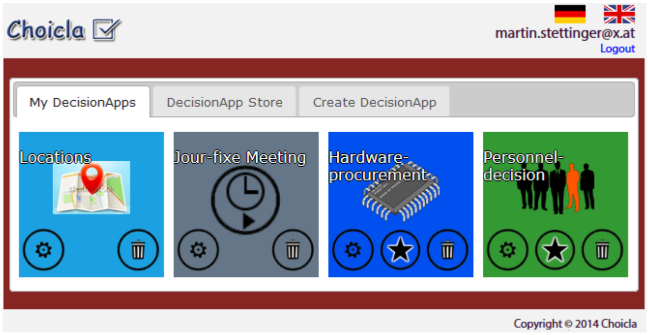


**Figure 2: *Choicla*: Home screen of a registered user. The symbols within the tiles trigger actions which can be performed in the current state of the decision app. Possible actions are (from left to right): configuration, evaluation (only possible if the decision app is publicly available over the store), and delete.**

The tab *DecisionApp Store* contains publicly available decision apps which can be searched and installed on the personal *Home Screen*. This method prevents a creation from scratch every time for frequent decision tasks such as, for example, scheduling decision tasks. In such a case the decision process can be triggered right after the download of a decision app. This reuse technique has the potential to reduce the entry barrier for using *Choicla* and keep the interaction simple – especially for people who want to start a decision process quickly. The tab *Create DecisionApp* allows a user to design a completely new decision app from scratch.

Due to the fact that many decision tasks occur regularly – for example, a group of friends go for dinner once a month – a concept is needed to manage a potentially large number of decision tasks. To keep the potentially large number of decision tasks manageable, every decision app consists of a variable number of instances. A concrete instance of a decision app can be accessed within the corresponding decision app - all instances of a concrete decision app will be loaded when the decision app is opened. The created instance of the example depicted in Figure 1 is accessible in the "Personnel-decision" app (see Figure 2). This mechanism offers the possibility of an exact documentation of all past decisions and is also a basis for supporting recurring decision tasks.

# 3. CHOICLA PERSONNEL DECISIONS

## 3.1 Users View

Personnel decisions are often influenced by various factors. Such factors are, for example, if a candidate has physical handicaps, in most cases no concrete structure is followed during the job interview and the evaluation often gets subjective. In such a case the assessment criteria of the candidates change and no "fair" and objective decision can be made. Another important factor is that in most cases personnel decisions come up in groups of users which means that often more than one person is affected by the hiring procedure.

To prevent groups from unsystematic reviews, *Choicla* offers a structured and fair way to evaluate candidates of a job position. Figure 3 shows the evaluation of the candidates in context of our working example (new receptionist) for a particular decision maker.

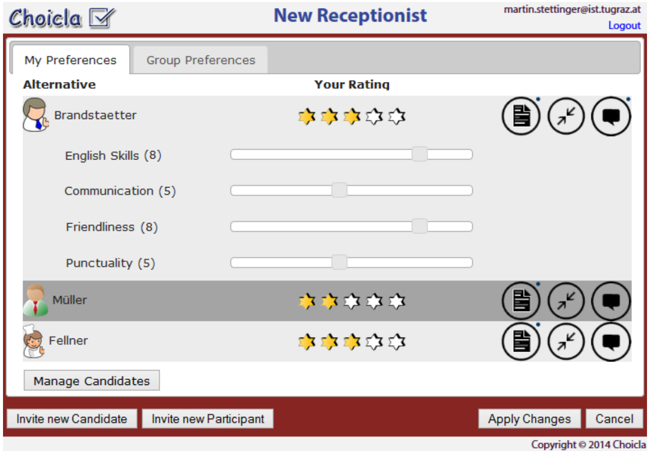


**Figure 3: *Choicla*: example of individual ratings. Each user can take a look at the current recommendation and adapt his/her preferences if needed.**

To keep the screen understandable, only the line with the aggregated information of a candidate is visible - by clicking on this line, several dimensions including their actual ratings show up for the corresponding candidate (only visible for first candidate in Figure 3). In order to avoid misunderstandings in context of evaluation the sliders of the first candidate are automatically displayed if the screen is loaded. Due to the fact that depending on the advertised job position different assessment criteria are needed, the dimensions on which a candidate can be evaluated can be chosen by the creator of a decision task. If we look at the example in Figure 3 we can see that for the "New Receptionist" the dimensions *English skills*, *Communication*, *Friendliness*, and *Punctuality* are chosen.

In situations where there are candidates for whom not all criteria (dimensions) have been evaluated or there exists a discrepancy between individual evaluations, special markers are used to point out open issues. This approach creates need for closure (see, e.g., [15]), i.e., users are additionally motivated to make the candidate evaluations complete and consistent.

If a candidate should be excluded from the application procedure in early phases (e.g., some criteria are not met), this can be achieved by using the "Manage Candidates" button (a new menu shows up). The early exclusion of an unsuitable candidate supports more clarity since only the "relevant" candidates are displayed.

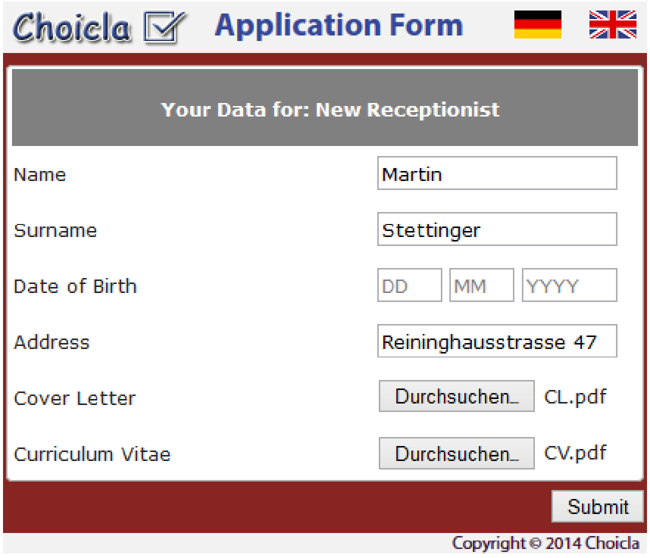
The tab *Group Preference* presents the current group recommendation, after a predefined number (the threshold) of participants articulated their preferences. This threshold prevents from statistical inferences to the individual preferences of other participants (only in combination with a "private" decision scope - see Section 2). The group recommendation in context of personnel decisions is based on the MAUT-principle (multi-attribute-utility-theory [1]). A group recommendation based on the MAUT-principle (see Formula 3) returns the average value of all individual MAUT values of all participants as group recommendation for one candidate (solution $s$). A group member's individual MAUT value represents the weighted average of all personal ratings of the dimensions of an alternative. This means that the attribute values are subjective and the weights are fixed which is different in a typical MAUT scenario.

$$MAUT(s) = \sum_{u \in Users} \frac{\sum_{d \in s} eval(u, d) * weight(d)}{|dimensions|} \qquad (3)$$

If we look at the individual ratings in Figure 3 we notice the values 8, 5, 8, and 5 for the dimensions. For simplification purposes we assume in our example that all dimensions have the same weight ($w_{d1} = w_{d2} = w_{d3} = w_{d4} = 5$). Due to Formula 3, the individual MAUT value for the actual user of the first alternative is 32.5. To present the evaluation of a solution (candidate) within a five star scale, these values have to be normed.

## 3.2 Candidates View

All previous described options and screens can only be accessed by the decision makers of the decision task itself and can of course not be seen by the applicants of the job position. During the design phase of a decision task the input fields (e.g., name, age, and application text) which are then visible by the applicants during the application process can be defined. Figure 4 shows the view of an applicant in our running example "New Receptionist".



Choicla Application Form

Your Data for: New Receptionist

| | |
|---|---|
| Name | Martin |
| Surname | Stettinger |
| Date of Birth | DD MM YYYY |
| Address | Reininghausstrasse 47 |
| Cover Letter | Durchsuchen... CL.pdf |
| Curriculum Vitae | Durchsuchen... CV.pdf |

Submit

Copyright © 2014 Choicla

**Figure 4: *Choicla*: example of the entering of application data. Each applicant can insert his/her personal data needed for the advertised job position.**

All the added information of the candidates is then prepared and accessible for the decision makers during the assessment phase - see Figure 3. This way of adding solutions to a decision process shifts the burden of entering candidate information by a single person - in most cases a secretary - to the applicants.

## 4. RELATED & FUTURE WORK

There exist a couple of online tools supporting decision scenarios. Rodriguez et al. [16] describes a system called *Smartocracy*. Smartocracy is a decision support tool which supports the definition of tasks in terms of issues or questions and corresponding solutions. The recommendation (solution selection) is based on exploiting information from an underlying social network which is used to rank alternative solutions. *Dotmocracy*[1] includes a method for collecting and visualizing the preferences of a large group of users. It is related to the idea of participatory decision making – it's major outcome is a graph type visualization of the group-immanent preferences. Doodle[2] is an internet calendar tool with the focus on coordinating appointments. VERN [19] is (very similar to doodle) a tool that supports the identification of meeting times. VERN is based on the idea of unconstrained democracy where individuals are enabled to freely propose alternative dates themselves. A major advantage of *Choicla*[3] compared to these tools is that users of *Choicla* are able to customize their decision processes depending on the application domain and can also focus on specific tasks. Furthermore, the mentioned tools provide no concepts which help to improve the overall quality of group decisions, for example, in terms of integrating explanations, recommendations for groups, and consistency management for user preferences.

Recommendation approaches in the line of *Choicla* are also presented in Sangeetha et al. [8] and Malinowski et al. [10]. Sangeetha et al. [8] introduce recommendation approaches that support people-to-people recommendation (detection of latent relationships between similar users) whereas Malinowski et al. [10] discuss approaches (based on fitness measures) that support the pre-selection of candidates for existing teams (groups). In contrast, *Choicla* focuses on supporting a group decision where parameters such as the fit of a candidate with an existing group are represented in terms of MAUT dimensions.

Our *future work* will focus on the analysis of further application domains for the *Choicla* technologies. Our vision is to make the design (implementation) of group decision tasks as simple and straightforward as possible. The resulting decision task should be easy to handle for users and make group decisions in general more efficient. Our focus will also be on the analysis of decision phenomena within the scope of group decision processes. Phenomena such as decoy effects [5], [18] and anchoring effects [6] have been well studied for single-user cases, however, in group-based decision scenarios no studies have been conducted.

Biases can be induced if a system is open in the sense that new decision alternatives can be added during the decision process. However, such a feature is imperative in cases where all possible decision alternatives are not available from the beginning. The group preferences can also be influenced by the order of the incoming individual preferences due to the fact that the participants of a group will perceive already selected alternatives more attractive than new options [14].

[1] dotmocracy.org.

[2] doodle.com.

[3] www.choicla.com.

If consensus out of discussion is reached in early phases, literature shows that this consensus is cognitive resistant to changes. That means that additional information which is added later in a decision process will be adapted to already defined consensus and due to this it is very unlikely that another alternative is chosen [9]. Such a phenomenon can be explained by the *assimilating effect* which is ascribable to the *dissonance theory* [4]. The *assimilating effect* states that individuals are motivated to reduce psychological incongruity or discrepancy that is very likely to arise if new information is added to a present perception [14]. A high group cohesion intensifies this effect, because within such a group the fear of exclusion is higher (see [9]). Future versions of *Choicla* will reduce this effect by providing a special way of preference visibility which, for example, only shows the preferences of other users for those participants who completed their individual ratings of the alternatives. Another research direction in this context is if such mechanisms can increase the willingness of participants to articulate their real preferences. A further issue for future work is to figure out which group recommendations help to achieve consensus more quickly. Finally, we will develop further group recommendation heuristics which help to achieve a high level of fairness (in the long run).

We want to emphasize that one of our major goals is to make the *Choicla* datasets available to the research community in an anonymized fashion for experimentation purposes.

## 5. CONCLUSIONS

In this paper we gave a short introduction to *Choicla* which supports the flexible design and execution of different types of group decision tasks with a focus on personnel decisions. With the help of *Choicla* it is possible to achieve more transparent, fair, and structured personnel decisions. Compared to existing group decision support approaches, *Choicla* provides an end user modelling environment which supports an easy development and execution of group decision tasks. We also discussed further research directions which can help to extend the available functionality of the *Choicla* environment.

## 6. REFERENCES

[1] J. Dyer. Maut - multiattribute utility theory. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research & Management Science*, pages 265–292. Springer New York, 2005.

[2] A. Felfernig, B. Gula, and E. Teppan. Knowledge-based Recommender Technologies for Marketing and Sales. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 21(2):1–22, 2006.

[3] A. Felfernig, C. Zehentner, G. Ninaus, H. Grabner, W. Maalej, D. Pagano, L. Weninger, and F. Reinfrank. Group decision support for requirements negotiation. In L. Ardissono and T. Kuflik, editors, *Advances in User Modeling*, volume 7138 of *Lecture Notes in Computer Science*, pages 105–116. Springer Berlin Heidelberg, 2012.

[4] L. Festinger. *A Theory of Cognitive Dissonance.* Stanford University Press, June 1957.

[5] J. Huber, J. Payne, and C. Puto. Adding Asymmetrically Dominated Alternatives: Violations of Regularity and the Similarity Hypothesis. *The Journal of Consumer Research*, 9(1):90–98, 1982.

[6] K. Jacowitz and D. Kahneman. Measures of Anchoring in Estimation Tasks. *Personality and Social Psychology Bulletin*, 21(1):1161–1166, 1995.

[7] A. Jameson. More than the sum of its members: challenges for group recommender systems. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 48–54, New York, NY, USA, 2004. ACM.

[8] S. Kutty, L. Chen, and R. Nayak. A people-to-people recommendation system using tensor space models. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 187–192, New York, NY, USA, 2012. ACM.

[9] E. Lind, L. Kray, and L. Thompson. Primacy effects in justice judgments: Testing predictions from fairness heuristic theory. *Organizational Behavior and Human Decision Processes*, 85(2):189 – 210, 2001.

[10] J. Malinowski, T. Weitzel, and T. Keim. Decision support for team staffing: An automated relational recommendation approach. *Decision Support Systems*, 45(3):429 – 447, 2008. Special Issue Clusters.

[11] J. Masthoff. Group Recommender Systems: Combining Individual Models. *Recommender Systems Handbook*, pages 677–702, 2011.

[12] A. Mojzisch and S. Schulz-Hardt. Knowing other's preferences degrades the quality of group decisions. *Journal of Personality & Social Psychology*, 98(5):794–808, 2010.

[13] E. Molin, H. Oppewal, and H. Timmermans. Modeling Group Preferences Using a Decompositional Preference Approach. *Group Decision and Negotiation*, 6:339–350, 1997.

[14] M. Neale, L. Ross, and J. Curhan. Dynamic Valuation: Preference Changes in the Context of Face-to-face Negotiation. *Journal of Experimental Social Psychology*, 40(2):142–151, 2004.

[15] G. Ninaus, A. Felfernig, M. Stettinger, S. Reiterer, G. Leitner, L. Weninger, and W. Schanil. Intelligent techniques for software requirements engineering. *European Conference on Artificial Intelligence, Prestigious Applications of Intelligent Systems (PAIS)*, to appear 2014.

[16] M. Rodriguez, D. Steinbock, J. Watkins, C. Gershenson, J. Bollen, V. Grey, and B. deGraf. Smartocracy: Social networks for collective decision making. In *HICSS 2007*, page 90, Waikoloa, Big Island, HI, USA, 2007. IEEE.

[17] M. Stettinger, A. Felfernig, G. Ninaus, M. Jeran, S. Reiterer, and G. Leitner. Configuring decision tasks. *Workshop on Configuration, Novi Sad*, pages 17–21, 2014.

[18] E. Teppan and A. Felfernig. Asymmetric dominance- and compromise effects in the financial services domain. In *Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on*, pages 57–64, July 2009.

[19] S. Yardi, B. Hill, and S. Chan. VERN: Facilitating Democratic group Decision Making Online. In *International ACM SIGGROUP Conference on Supporting Group Work (GROUP 2005)*, pages 116–119, Sanibel Island, Florida, USA, 2005. ACM.

# An empirical study on the persuasiveness of fact-based explanations for recommender systems

Markus Zanker
Alpen-Adria-Universitaet Klagenfurt
9020 Klagenfurt, Austria
mzanker@acm.org

Martin Schoberegger
Alpen-Adria-Universitaet Klagenfurt
9020 Klagenfurt, Austria
m3schobe@edu.uni-klu.ac.at

## ABSTRACT

Recommender Systems (RS) help users to orientate themselves in large product assortments and provide decision support. Explanations help recommender systems to enhance their impact on users by, for instance, justifying made recommendations. Arguments provide reason in a more structured way, by denoting a conclusion that follows from one or more premises. While expert systems' explanation have a long tradition in using argumentative patterns, argumentative explanations for recommendations have not yet been systematically researched. This paper compares therefore the persuasion potential of different explanation styles (sentences, facts or argument style) by comparing the robustness of subjects' preferences when employing an additive utility model from conjoint analysis.

## Keywords

Recommender Systems, Explanation styles, Persuasion potential of explanations

## 1. INTRODUCTION

Recommender Systems (RS) support online customers in their decision making and should help them to avoid poor decisions [4]. Persuasive systems [9] are focusing on changing a user's belief or actions in an intended way. In this context recommender systems need to be also seen as persuasive systems, as their purpose lies in pointing users towards unknown items that presumably match their interest, i.e. making serendipitous propositions. This clearly differentiates a recommendation system (RS) from an information retrieval (IR) system that assumes an objective information need of a user that can satisfied. In general explanations can be seen as an attempt to fit a particular phenomenon into a general pattern in order to increase understanding and remove bewilderment or surprise [5]. In the context of product recommendation scenarios explanations can be seen as additional information about recommendations [2] that serves the purpose of justifying why a specific item is part of a recommendation list and promote objectives such as users' trust in the system and confidence in decision making. In the domain of expert systems explanations have already a long tradition, where formal argumentation traces can serve as explanations that justify the output of a system [8]. According to [5] an argument is (a) a series of sentences, statements, or propositions (b) where some are premises (c) and one is the conclusion (d) where the premises are intended to give a reason for the conclusion. As we believe that research on explanations in general and comparative studies on competing explanation styles are rare (a few pointers to more recent exceptions [7, 6, 3]), we conducted a supervised lab study that had the purpose to research the impact of different explanation styles of knowledgeable explanations [11]. In particular we are interested in effects on the robustness of users' preferences when confronted with additional explanations, i.e. exploring the persuasion potential of explanations. More concretely we compared fact-based explanations, that presented keywords as explanations to users, such as A, B, C, with a basic argument style with A and B as premises and C as a consequent, i.e. A, B therefore C. Furthermore, we compared these fact-based explanations to sentence-based explanations requiring more cognitive effort to understand them. We selected three different item domains that typically trigger high involvement of users, i.e. hiking routes from the tourism and leisure domain (hiking routes), energy plans and mobile phone plans, and controlled for user preferences, item portfolio and the semantics of the explanations themselves. We would like to note that the study was conducted in the scope of the O-STAR project that researches techniques for personalized route planning for hikers in alpine regions. Next we will provide details on our study design and finally discuss results and conclusions.

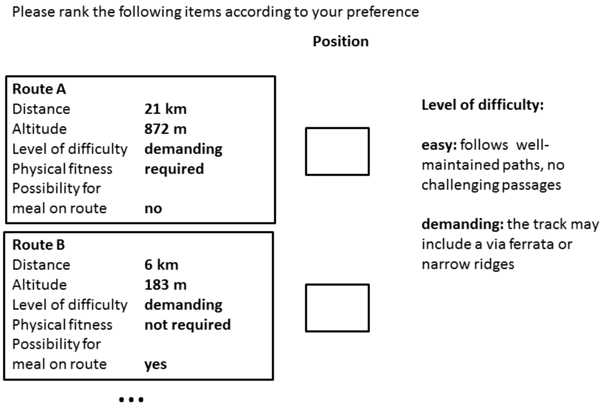
## 2. STUDY DESIGN

We researched the question if the introduction of an argument-based writing style, i.e. use of the keyword *therefore* to denote the conclusion of the preceding premises, has an impact on the robustness of users' preferences in face of additional explanations. As already mentioned we asked users to disclose their preferences for three different item domains (hiking routes, mobile phone plans and energy plans) in a supervised offline questionnaire. Figures 1 and 2 depict two exemplary items from the hiking domain. Subjects were invited to participate in a seminar room, where they had to answer a paper & pencil survey with two parts. The first part included for each of the three domains exactly 6 items, that are described by either 4 or 5 characteristics.

*IntRS 2014*, October 6, 2014, Silicon Valley, CA, USA.

| **Hiking routes** | |
|---|---|
| Distance | in km |
| Altitude | in m |
| Level of difficulty | easy or demanding |
| Physical fitness | (not) required |
| Possibility for meal on route | yes/no |
| **Energy plans** | |
| Renewable energy | 100%/no |
| Pricing | dynamic vs. fixed |
| Fixed contract duration | in months |
| Guaranteed price | yes/no |
| **Mobile phone plans** | |
| Basic fee | in EUR |
| Type of phone | Smartphone vs. simple phone |
| Anytime minutes | amount |
| Fixed contract duration | in months |

**Table 1: Attributes describing item domains**



Please rank the following items according to your preference

**Position**

| **Route A** | |
|---|---|
| Distance | **21 km** |
| Altitude | **872 m** |
| Level of difficulty | **demanding** |
| Physical fitness | **required** |
| Possibility for meal on route | **no** |

□

| **Route B** | |
|---|---|
| Distance | **6 km** |
| Altitude | **183 m** |
| Level of difficulty | **demanding** |
| Physical fitness | **not required** |
| Possibility for meal on route | **yes** |

□

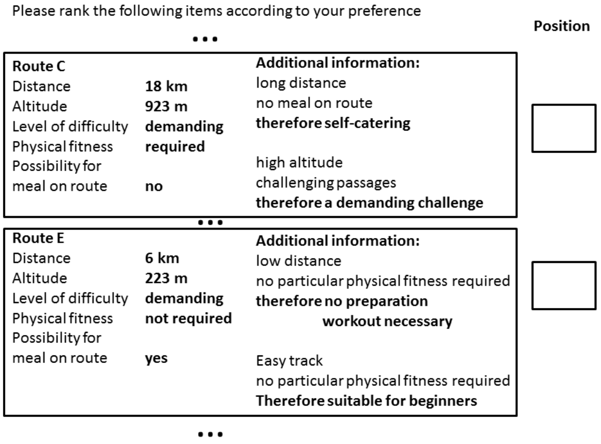
...

**Level of difficulty:**

**easy:** follows well-maintained paths, no challenging passages

**demanding:** the track may include a via ferrata or narrow ridges

**Figure 1: Excerpt from questionnaire - part 1**



Please rank the following items according to your preference

**Position**

...

| **Route C** | |
|---|---|
| Distance | **18 km** |
| Altitude | **923 m** |
| Level of difficulty | **demanding** |
| Physical fitness | **required** |
| Possibility for meal on route | **no** |

**Additional information:**
long distance
no meal on route
**therefore self-catering**

high altitude
challenging passages
**therefore a demanding challenge**

□

...

| **Route E** | |
|---|---|
| Distance | **6 km** |
| Altitude | **223 m** |
| Level of difficulty | **demanding** |
| Physical fitness | **not required** |
| Possibility for meal on route | **yes** |

**Additional information:**
low distance
no particular physical fitness required
**therefore no preparation**
**workout necessary**

Easy track
no particular physical fitness required
**Therefore suitable for beginners**

□

...

**Figure 2: Excerpt from questionnaire - part 2**

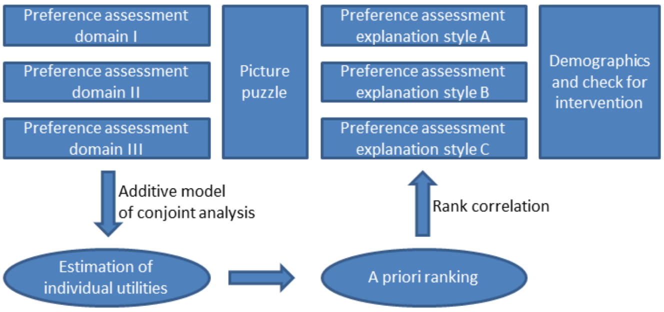

Preference assessment domain I
Preference assessment domain II
Preference assessment domain III
Picture puzzle
Preference assessment explanation style A
Preference assessment explanation style B
Preference assessment explanation style C
Demographics and check for intervention
Additive model of conjoint analysis
Rank correlation
Estimation of individual utilities
A priori ranking

**Figure 3: Big picture of research design**

Table 1 depicts the three item domains and the artificial design space of the item portfolios. To avoid confusion the semantics of the domain attributes were defined in a sidebar (e.g. Smartphone: denotes a device in the range of HTC Desire X or Nokia Lumia 625). Participants had to rank the 6 options according to their general preference with respect to the particular item domain. After disclosing their preferences in the first part of the questionnaire (see Figure 1 for a translated excerpt of the questionnaire) users had to solve a picture puzzle, where 10 different errors were hidden. The purpose of this task is twofold: first, it distracts users from their thoughts on the ranking tasks and, second, we could use the numerical measure of correctly marked errors to assess how concentrated participants followed the questionnaire. Once participants had finished the first part they handed it in and received the second part of the survey. This way we were able to avoid that participants could have taken a look on their first-round ranking when answering the second part. In the second part participants had again to rank sets of five items from the three item domains. However, in addition to the item characteristics already used in the first-round, additional explanations were given for each item. The *explanation style* acts as the manipulated variable (solely fact-based, argumentative facts and argumentative sentences). Explanation style is permuted within subjects, i.e. participants are confronted with all three explanation styles for a different item domain and in different orders, while the combination of item domain and explanation style is varied between subjects. For each item exactly two arguments, each with two premises and one conclusion, are added as additional information (see examples in Table 2). See Figure 2 for a depiction of two exemplary items from the hiking domain with explanations following the style of argumentative facts.

Finally, the questionnaire controlled for demographic characteristics and checked if participants noticed the intervention, i.e. one question asked what was relevant for ranking the items with multiple answering options. For analysis we selected only participants that considered the *additional explanations* provided in the second part in their ranking decision.

In Figure 3 we sketch the big picture of the study design. Thus, participants rank sets of items from three different domains twice, where item sets in the first and second part of the questionnaire do not overlap. Due to measuring user preferences twice for each domain (without and with intervention of a specific explanation style), we can control for the participants' preferences on item sets and their presentation. We employ an additive model from conjoint analysis, that allows us to estimate the utilities for each item characteristic [1], i.e. the overall utility of an item $y_i$ is computed as the sum $\mu + \sum_Z \beta_Z$, where $\mu$ is a basic utility and

| | |
|---|---|
| **Hiking routes** | |
| **Solely facts** | low altitude<br>easy distance<br>very family-friendly |
| **Argumentative facts** | low altitude<br>easy distance<br>therefore very family-friendly |
| **Argumentative sentences** | This route is of low altitude and easy distance, therefore it is very family-friendly. |
| **Energy plans** | |
| **Solely facts** | 100% renewable energy<br>low environmental impact<br>high sustainability |
| **Argumentative facts** | 100% renewable energy<br>low environmental impact<br>therefore high sustainability |
| **Argumentative sentences** | This energy plan offers 100% renewable energy with a low environmental impact, therefore its sustainability is high. |
| **Mobile phone plans** | |
| **Solely facts** | low basic fee<br>many anytime minutes<br>ideal for heavy use |
| **Argumentative facts** | low monthly basic fee<br>many anytime minutes<br>therefore ideal for heavy use |
| **Argumentative sentences** | This mobile phone plan offers a low monthly basic fee with many anytime minutes, therefore it is ideal for heavy use. |

**Table 2: Example explanations/arguments for each of the three item domains**

$\beta_Z$ denotes the positive or negative utility contributed by a specific item characteristic $Z$ (for instance, the possibility to have your meal on route in the hiking domain). Having estimated the individual utilities of each item characteristic we computed an a priori ranking for the unseen item sets in the survey's second part that is then compared with the observed ranks for each user.

## 3. RESULTS AND DISCUSSION

In total 136 subjects, mostly students from Alpen-Adria-Universtät Klagenfurt, participated in our survey. From each participant we received three rankings in the second part of the survey (one for each domain), i.e. a total of 408 computed rank correlations before cleaning. More than 80% of all participants were young people aged between 18 and 25. Two thirds of our participants were females. All respondents had a high-school degree and a few of them had already a graduation degree from a university. Before analysis we rigorously excluded participants whose answers might be unreliable due to the following criteria:

1. Only respondents who demonstrated a thorough attitude by identifying at least 50% of all hidden errors in the picture puzzle.
2. We asked participants what they considered to be relevant for making their decisions on the rankings. Based on the answers to this multiple choice question we included only respondents who had noticed the additional information (explanations) and excluded all respondents who answered that they relied on their gut feelings.
3. We also asked participants how they experienced this survey with the answering options *interesting*, *challenging*, *boring*, *unclear* and *useless*. For further consideration we only kept respondents that answered *challenging* and were thus captivated by the ranking tasks. We assumed that the option *interesting* is a polite way of saying boring or useless.
4. Finally we cleaned records from the dataset, where the estimation of individual utilities for product characteristics was not reliable, i.e. rank correlation between the a priori rankings based on estimated utility weights and the actual a priori ranking of participants had to be above 0.7.

After applying this extremely restrictive selection procedure we derived at the following size of the dataset (see Table 3). In order to check for the robustness of preferences after introducing argument-based explanations we compute Spearman rank correlation coefficient between the a priori rankings based on estimated utility weights and the empirical rankings by participants. Table 4 reports the averaged Spearman's $\rho$ for each explanation style and aggregated over domains. As can be seen from Table 4 the argumentation-styled facts that included the keyword *therefore* to denote a conclusion reduced the robustness of participants' preferences more than the pure fact-based explanations, i.e. supporting our hypothesis that an argumentative explanation style would influence users more. Argumentative sentences preserved user preferences more than the fact-based explanation styles. Obviously, sentences need more cognitive effort from users to be understood and the effect of the keyword *therefore* was seemingly lost in the sentence structure. The difference between Spearman's $\rho$ in all three categories is statistically significant according to Kruskall-Wallis test (p = 0.037).

| | Hiking | Energy | Mobile |
|---|---|---|---|
| Solely facts | 10 | 12 | 7 |
| Argumentative facts | 6 | 12 | 13 |
| Argumentative sentences | 10 | 5 | 8 |

**Table 3: Respondents per domain and expl. style**

| **Explanation style** | Rank correlation |
|---|---|
| Solely facts | 0.43 |
| Argumentative facts | **0.36** |
| Argumentative sentences | 0.67 |

**Table 4: Robustness of preferences in face of different explanation styles**

In addition we checked for interaction effects between explanation style and product domain. As can be seen from Table 5 fact-based explanation styles lead to less robust preferences than sentence-based explanation styles. Furthermore, argumentative facts seem to reduce participant's robustness of preferences even more than a pure facts based explanation style. The only exception is the hiking domain, where the order between facts and argumentative facts is inverted. However, in this product domain preference robustness is generally lower and it might have been harder for respondents to determine own preferences in the hiking domain than in the other two domains.

| | Hiking | Energy | Mobile |
|---|---|---|---|
| Solely facts | 0.27 | 0.48 | 0.58 |
| Argumentative facts | 0.38 | 0.34 | 0.38 |
| Argumentative sentences | 0.58 | 0.78 | 0.71 |

**Table 5: Spearman's $\rho$ per domain and expl. style**

This study therefore showed, that fact-based explanations and an argumentative explanation style impacted participants' preferences stronger than full sentence explanations. Objections against these conclusions might be the lack of a control group and the paper & pencil design without a real recommendation situation. A control group would allow us to estimate the *natural* stability of preferences between both rounds and without any intervention. However, in this study we were not interested in absolute rank correlation measures, but only in the comparison of robustness of respondents' preferences between different conditions and assumed that some *natural instability* would affect all explanation styles the same way. In order to assess the impact of an argumentative explanation style we wanted to control for other effects and biases as good as possible. The supervised paper & pencil approach allowed us to control for user preferences, the item portfolio and the persuasiveness of the explanation content itself as well as insisting on a high reliability of the measurements by excluding participants, who made arbitrary rankings or did not notice the additional explanations. In a previous study [10] we already compared the sentence-based explanations with a no-explanations control group and observed their positive impact on the perception of the recommender system as a whole. However, one could not isolate the impact on the robustness of preferences by controlling for the different recommendation lists, the different explanation content that would apply to different recommendations or the differing appreciation of the recommendation results themselves by participants.

# 4. CONCLUSIONS

This short paper presented an innovative study design for measuring the impact of different explanation styles on participants' robustness of preferences in face of additional explanations. The results indicate that fact-based explanations have a stronger impact on participants preference stability than sentence-based explanations. Furthermore, the use of the keyword *therefore* indicating a conclusion drawn from premises and an argumentative explanation style had already a measurable impact on participants. Thus arguments and fact-based explanations make users change their minds about the item portfolio and can therefore be valuable features of recommender systems. Limitations or possible lines of future research include varying the complexity of arguments (i.e the number of premises) or its number as well as additional item domains.

# Acknowledgements

Authors acknowledge the financial support from the European Union (EU), the European Regional Development Fund (ERDF), the Austrian Federal Government and the State of Carinthia in the Interreg IV Italien-Österreich programme (project acronym O-STAR).

# 5. REFERENCES

[1] Klaus Backhaus, Bernd Erichson, Wulff Plinke, and Rolf Weiber. *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung*. Springer, Berlin, 12., vollständig überarbeitete auflage. edition, 2008.

[2] Gerhard Friedrich and Markus Zanker. A taxonomy for generating explanations in recommender systems. *AI Magazine*, 32(3):90–98, 2011.

[3] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. How should i explain? a comparison of different explanation types for recommender systems. *Int. J. Hum.-Comput. Stud.*, 72(4):367–382, 2014.

[4] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*. Cambridge Univ Pr, 2010.

[5] W. Sinnott-Armstrong and R. Fogelin. *Cengage Advantage Books: Understanding Arguments*. Wadsworth, 2014.

[6] Nava Tintarev and Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):399–439, 2012.

[7] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: Explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 47–56, New York, NY, USA, 2009. ACM.

[8] L. R. Ye and P. E. Johnson. The impact of explanation facilities in user acceptance of expert system advice. *MIS Quarterly*, 19(2):157–172, 1995.

[9] Kyung Hyan Yoo, Ulrike Gretzel, and Markus Zanker. *Persuasive Recommender Systems - Conceptual Background and Implications*. Springer Briefs in Electrical and Computer Engineering. Springer, 2013.

[10] Markus Zanker. The influence of knowledgeable explanations on users' perception of a recommender system. In Padraig Cunningham, Neil J. Hurley, Ido Guy, and Sarabjot Singh Anand, editors, *RecSys*, pages 269–272. ACM, 2012.

[11] Markus Zanker and Daniel Ninaus. Knowledgeable explanations for recommender systems. In Jimmy Xiangji Huang, Irwin King, Vijay V. Raghavan, and Stefan Rueger, editors, *Web Intelligence*, pages 657–660. IEEE, 2010.

# The Effect of Different Set-based Visualizations on User Exploration of Recommendations

**Katrien Verbert**[1]**, Denis Parra**[2]**, Peter Brusilovsky**[3]
[1]Departement Computerwetenschappen, KU Leuven, Celestijnenlaan 200A, B-3001 Leuven, Belgium
katrien.verbert@cs.kuleuven.be
[2]Department of Computer Science, Pontificia Universidad Católica de Chile, Santiago, Chile
dparra@ing.puc.cl
[3]School of Information Sciences, University of Pittsburgh, 135 North Bellefield Avenue,
Pittsburgh, PA 15260, USA, peterb@pitt.edu

## ABSTRACT

When recommendations fail, trust in a recommender system often decreases, particularly when the system acts like a "black box". To deal with this issue, it is important to support exploration of recommendations by explicitly exposing relationships that can provide explanations. As an example, a graph-based visualization can help to explain collaborative filtering results by representing relationships among items and users. In our work, we focus on the use of visualization techniques to support exploration of *multiple* relevance prospects - such as relationships between different recommendation methods, socially connected users and tags. More specifically, we researched how users explore relationships between such multiple relevance prospects with two set-based visualization techniques: a clustermap and a Venn diagram. A comparative analysis of user studies with these two approaches indicates that, although effectiveness of recommendations increases with the use of a clustermap, the approach is too complex for a non-technical audience. A Venn diagram representation is more intuitive and users are more likely to explore relationships that help them find relevant items.

### Author Keywords

User interfaces for recommender systems; information visualization; user studies.

### ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): User interfaces. H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

### General Terms

Human Factors; Design; Experimentation.

## INTRODUCTION

The design and development of user interfaces for recommender systems has gained increased interest. Such interfaces are researched to provide new capabilities to search, browse, and understand recommendations [8]. Among others, explaining recommendations to provide transparency and to increase trust has been researched extensively [11]. Several approaches have been presented that represent relationships between users and items as a basis to support exploration and transparency [5][3][12]. Most of these existing approaches enable users to explore relationships between two entities, such as relationships between *users* and recommended *items*.

In our work, we focus on the use of set-based visualization techniques to support exploration of *multiple* relevance prospects. In contrast to existing approaches, we enable end-users to interrelate multiple dimensions to support exploration and transparency of recommendations.

We have developed two visual interfaces for exploring relationships between multiple relevance prospects of recommendations. A first user interface (TalkExplorer) uses a clustermap visualization technique that enables users to explore relationships between diverse *recommendations*, *users* and *tags*. A second interface (SetFusion) uses a Venn diagram to support exploration of multidimensional relationships.

The original work on TalkExplorer [12] and SetFusion [7] has been performed independently, with no intention to compare the results of our studies with these sufficiently different systems. At the same time, an extensive set of data collected in several user studies opened an interesting opportunity to uncover the participation puzzle that we observed when comparing the results of two TalkExplorer studies. These results indicate that effectiveness of recommendations increases in a significant way when users are able to interrelate multiple entities. However, when deployed in an open setting, users do no explore such intersections often when a clustermap is used.

In earlier work, we hypothesized that the likely reason for this phenomenon is the complexity of the TalkExplorer interface, especially the challenge of understanding the complex visualization of overlapping sets. However, at that time we were not able to provide any arguments in favor of these hypotheses.

In this paper, we re-assess this hypothesis. The presence of the SetFusion study that was performed in the same system, with similar kind of data and using a similar approach, enables us to compare how users interact with the visualizations. SetFusion explores exactly the kind of interface that we believe could increase exploration of overlaps: Venn Diagrams are known to be both straightforward and standard to visualize set overlaps. We re-process the data of our user studies and analyze how users interact with both interfaces to re-assess their value for exploring recommendations.

This paper is organized as follows: first we present related work in the area of visualizing recommendations and set-based visualization. Then, we introduce TalkExplorer, an interactive clustermap visualization of recommendations, as well as SetFusion, an interactive Venn diagram representation of recommendations. Results of user studies conducted with both interfaces are presented next. Then, we present a comparative analysis of how users interact with these visualizations. Finally, we discuss these results, as well as future research opportunities.

## RELATED WORK

Most existing work in the area of visualizing recommendations focuses on interaction with collaborative filtering recommender systems. PeerChooser [5] is a visual interactive recommender that uses a graph-based representation to show relationships between users and recommended items of a collaborative filtering recommender system. Similarly, SmallWorlds [3] allows exploration of relationships between recommended items and similar friends, in multiple layers of similarity. These systems enable users to explore such relationships as a basis to provide transparency and to support the user to find new relevant items.

Some systems focus specifically on tags that are used by social recommenders. SFViz (Social Friends Visualization) [4] visualizes social connections among users and user interests as a basis to increase awareness in a social network and to help people find potential friends with similar interests. This system uses a Radial Space-Filling (RSF) technique to visualize a tag tree and a circle layout with edge bundling to show a social network.

More recently, TasteWeights [2] has been introduced as a system that allows users to control the importance of friends and peers in social systems to obtain recommendations. Similar to our work, TasteWeights introduces the concept of an interface for hybrid recommender systems. The system elicits preference data and relevance feedback from users at run-time and uses these data to adapt recommendations to the current needs of the user. The idea can be traced back to work of Schafer et al. [9] on meta-recommendation systems. These meta-recommenders provide users with personalized control over the generation of recommendations by indicating how important specific factors are – such as genre of a movie and film length, on a scale from 1 (not important) to 5 (must have). In our work, we extend this concept by visualizing relationships to relevance prospects in order to enhance exploration by end-users of the item space and to increase perceived relevance and meaning of items. More specifically, we use a set-based visualization approach to represent relationships of items to specific relevance factors or prospects. Thus, in addition to enabling end-users to specify which prospects are relevant, we enable them to see how recommendations are related to these prospects with set-based visualization techniques.

Relevance or set-based visualization applies an approach to spatially organize recommendation results. Relevance-based visualization has been originally developed in the field of information retrieval for visualization of search results. For example, for a query that uses three terms, it will create seven set areas to show which results are relevant to each of the three terms, each of two pairs of these terms, and all three terms at the same time. The classic example of set-based relevance visualization is InfoCrystal [10]. The Aduna clustermap visualization [1] approach also belongs to this category offering a more complex visualization paradigm and a better level of interactivity. A strong point of set-based approach is a clear representation to which of the query terms each document is relevant along with grouping documents by this aspect.

The novelty of the approach suggested in our paper is twofold. First, we are using a set-based relevance approach not just with keywords or tags where relevance approaches are usually applied, but with a diverse set or relevance-bearing entities (tags, users, recommendation agents). To the best of our knowledge, this is the first attempt to visually represent recommendations with set-based visualization techniques. The major difference and innovation of our work is that we allow end-users to combine *multiple* relevance prospects in order to increase the perceived relevance and meaning of recommendations. Second, we present two different techniques to visually present these sets: a clustermap visualization, implemented in TalkExplorer [12], and a Venn diagram, implemented in SetFusion [7]. Although the interactive hybrid recommender interface TasteWeights [2] and meta-recommendation systems [9] also allow users to consider three potential sources of relevance to make recommendations, TalkExplorer allows more flexible exploration by visually presenting relationships to relevance prospects with a clustermap, and SetFusion uses a completely different visualization paradigm, relying on a Venn diagram. We present results of user studies with these visualizations that assess the impact of the interfaces on the effectiveness of recommendations, as well as a comparative analysis of how users interact with these representations.

## TALKEXPLORER AND SETFUSION

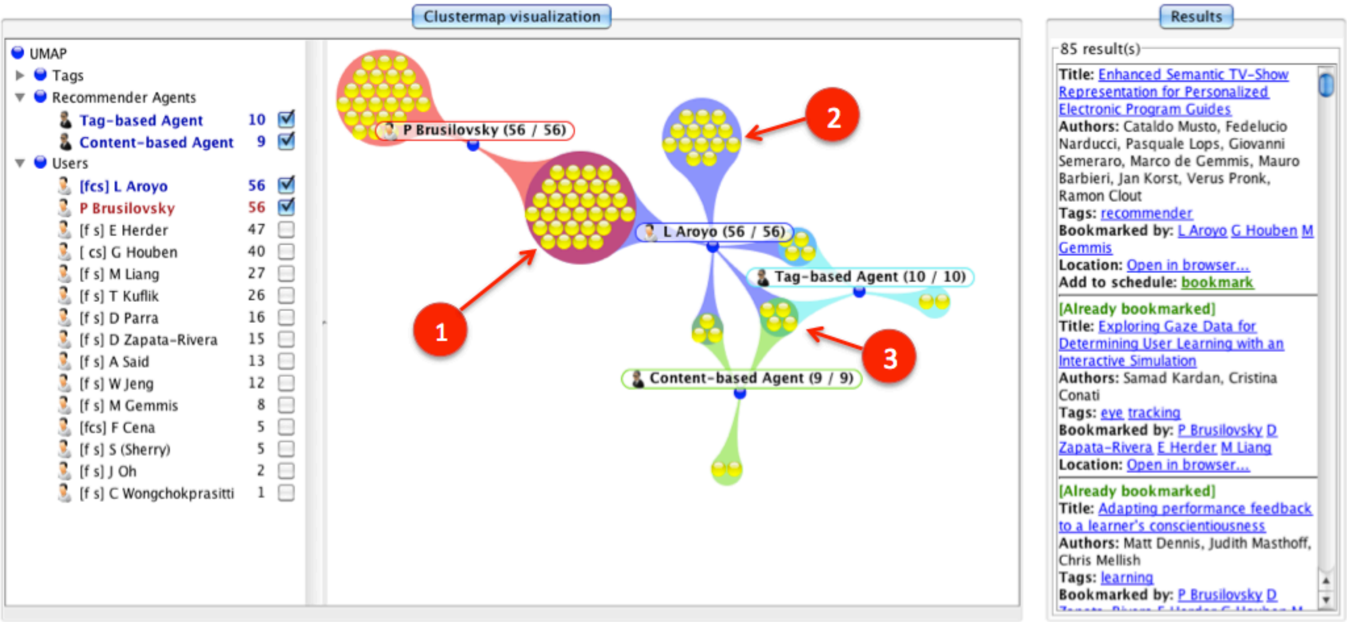
TalkExplorer and SetFusion represent two attempts to implement a visual interactive interface to explore recommendations of research talks at academic conferences. Both visualization interfaces were implemented and released as components of the conference support system *Conference Navigator 3 (CN3)* [6]. Each of the interfaces was developed to explore a range of ideas related to visualization, interactive access, transparency, etc. One of the core ideas essential for the purpose of this paper was integration of several aspects of relevance within the same visualization. We believed that a talk might be perceived by users as relevant for a range of reasons that we call aspects (for example, it could be recommended by one of the recommender engines or bookmarked by a socially connected user). We also believed that talks that are relevant in more that one aspect could be more valuable to the users and that displaying multiple aspects of relevance visually is important for the users in the process of talk exploration. Following these beliefs, TalkExplorer and SetFusion offered two different approaches to visualize talk relevance in a way that helps to identify talks that are relevant for the users in two, three, and even more aspects. Both systems use different versions of set-based visualizations to achieve this goal. The user studies that we ran with both interfaces included specific provisions that enabled us to examine the value of displaying several aspects of relevance. The next sections explain the details of both visualization approaches and results of their evaluation that are relevant for this paper.

## VISUALIZING RELATIONSHIPS IN TALKEXPLORER

The key idea of TalkExplorer is to enable users to explore talks recommended by two recommender engines (presented in the interface as recommender agents) along with talks that were bookmarked or tagged by other system users. The visualization is implemented as a Java applet and uses the Aduna clustermap visualization library [1]. This software library visualizes sets of categorized objects and their interrelationships.

Recommender systems are presented as *agents* and their interrelationships can be explored. In parallel, real users and their bookmarks are shown and users can explore both interrelationships between users as well as interrelationships between agents and users (i.e. which other users have bookmarked talks that are recommended to them by one or more agents). In addition, relationships with tags can be explored to identify relevant items. We are researching whether visualizing these relationships can help users to find relevant talks to attend at a conference, and whether this visualization can provide transparency and increase trust.

TalkExplorer allows users to explore the different entities of the conference by means of three principal components, as shown in Figure 1. On the left side, the entity selection panel allows users to select tags, users and recommender agents that are added and displayed in the canvas area. This canvas area, at the center of the screen, shows a clustermap visualization - i.e., different clusters of talks linked by connected components. The labeled circles in this canvas area represent either real users, recommender agents or tags. Yellow circles represent individual talks, and the bubbles that involve them represent clusters of talks.



● UMAP
▸ ● Tags
▾ ● Recommender Agents

| | | |
|---|---|---|
| Tag-based Agent | 10 | ☑ |
| Content-based Agent | 9 | ☑ |

▾ ● Users

| | | |
|---|---|---|
| [fcs] L Aroyo | 56 | ☑ |
| P Brusilovsky | 56 | ☑ |
| [f s] E Herder | 47 | □ |
| [ cs] G Houben | 40 | □ |
| [f s] M Liang | 27 | □ |
| [f s] T Kuflik | 26 | □ |
| [f s] D Parra | 16 | □ |
| [f s] D Zapata-Rivera | 15 | □ |
| [f s] A Said | 13 | □ |
| [f s] W Jeng | 12 | □ |
| [f s] M Gemmis | 8 | □ |
| [fcs] F Cena | 5 | □ |
| [f s] S (Sherry) | 5 | □ |
| [f s] J Oh | 2 | □ |
| [f s] C Wongchokprasitti | 1 | □ |

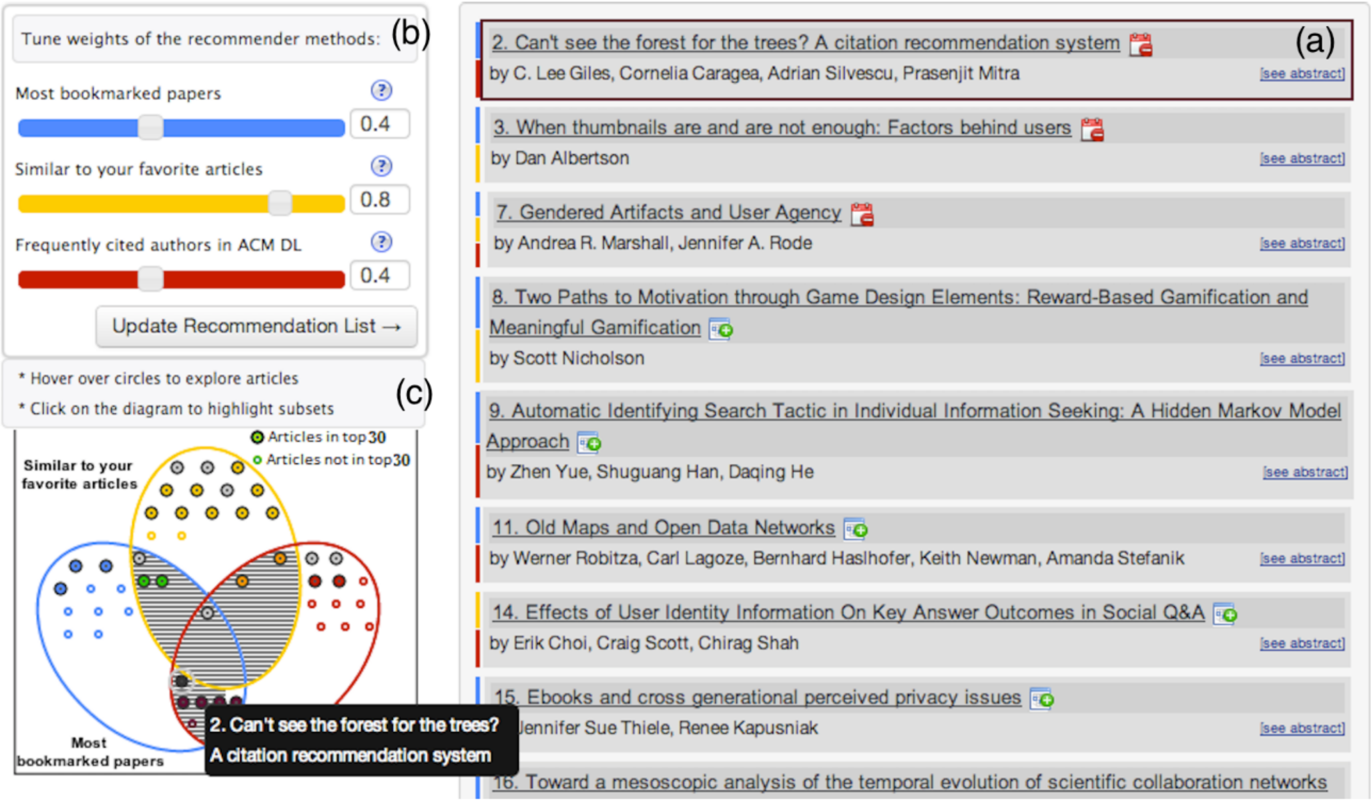
85 result(s)

**Title:** Enhanced Semantic TV-Show Representation for Personalized Electronic Program Guides
**Authors:** Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Giovanni Semeraro, Marco de Gemmis, Mauro Barbieri, Jan Korst, Verus Pronk, Ramon Clout
**Tags:** recommender
**Bookmarked by:** L Aroyo G Houben M Gemmis
**Location:** Open in browser...
**Add to schedule:** bookmark

[Already bookmarked]
**Title:** Exploring Gaze Data for Determining User Learning with an Interactive Simulation
**Authors:** Samad Kardan, Cristina Conati
**Tags:** eye tracking
**Bookmarked by:** P Brusilovsky D Zapata-Rivera E Herder M Liang
**Location:** Open in browser...

[Already bookmarked]
**Title:** Adapting performance feedback to a learner's conscientiousness
**Authors:** Matt Dennis, Judith Masthoff, Chris Mellish
**Tags:** learning
**Bookmarked by:** P Brusilovsky D

**Figure 1:** Screenshot of TalkExplorer. Labeled numbers indicate clusters of talks (yellow circles) which are the result of intersecting talks bookmarked or tagged by real users, or suggested by recommender agents.



Tune weights of the recommender methods:

Most bookmarked papers 0.4

Similar to your favorite articles 0.8

Frequently cited authors in ACM DL 0.4

Update Recommendation List →

* Hover over circles to explore articles

* Click on the diagram to highlight subsets

2. Can't see the forest for the trees? A citation recommendation system

2. Can't see the forest for the trees? A citation recommendation system
by C. Lee Giles, Cornelia Caragea, Adrian Silvescu, Prasenjit Mitra [see abstract]

3. When thumbnails are and are not enough: Factors behind users
by Dan Albertson [see abstract]

7. Gendered Artifacts and User Agency
by Andrea R. Marshall, Jennifer A. Rode [see abstract]

8. Two Paths to Motivation through Game Design Elements: Reward-Based Gamification and Meaningful Gamification
by Scott Nicholson [see abstract]

9. Automatic Identifying Search Tactic in Individual Information Seeking: A Hidden Markov Model Approach
by Zhen Yue, Shuguang Han, Daqing He [see abstract]

11. Old Maps and Open Data Networks
by Werner Robitza, Carl Lagoze, Bernhard Haslhofer, Keith Newman, Amanda Stefanik [see abstract]

14. Effects of User Identity Information On Key Answer Outcomes in Social Q&A
by Erik Choi, Craig Scott, Chirag Shah [see abstract]

15. Ebooks and cross generational perceived privacy issues
Jennifer Sue Thiele, Renee Kapusniak [see abstract]

16. Toward a mesoscopic analysis of the temporal evolution of scientific collaboration networks

**Figure 2:** Screenshot of SetFusion displaying (a) a filtered list of papers recommended, (b) sliders, and (c) the Venn diagram

In Figure 1, two users are shown (P Brusilovsky and L Aroyo), as well as suggestions of the tag-based and content-based recommender agent. The clustermap visualization enables users to explore relationships between items that were suggested to them by these recommender agents and bookmarks of users on the screen. For instance, a user can see which other users have bookmarked a talk that is suggested by a recommender agent by exploring the intersection of the agent and a specific user. In the example presented in Figure 1, the active user (P Brusilovsky) can explore which of the talks he has bookmarked are also bookmarked by user L Aroyo (label 1), which additional talks are bookmarked by L Aroyo but not recommended by an agent (label 2) and which talks are recommended to him by both the content-based and tag-based agent and are also bookmarked by L Aroyo (label 3) - to further filter out the potentially more relevant recommendations.

Finally, the rightmost panel shows the detailed list of talks. This can be a list of all the talks presented in the canvas area, or a subset of them related to the selected entity. If a user clicks on a cluster (for example, the cluster showing talks that were bookmarked by L Aroyo and a specific agent) the list of these talks and their details are presented.

## VISUAL HYBRID RECOMMENDATION IN SETFUSION

SetFusion is inspired by the same set-based approach than TalkExplorer, i.e., allowing users to choose items by combination of multiple prospects of relevance. The main difference is that SetFusion uses a Venn diagram rather than a clustermap with links to show the intersections (fusions).

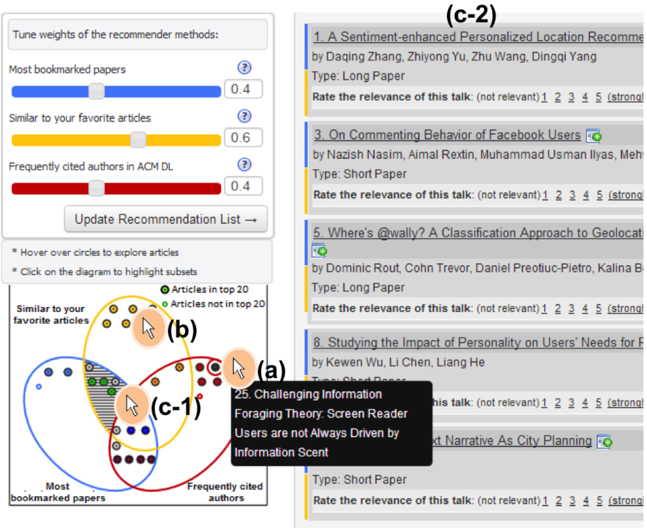
Another difference is the type of entities used as relevance prospects in order to support decision-making. While TalkExplorer uses tags, recommender agents and users, SetFusion mixes three recommendation methods, turning it into a hybrid recommender. The methods that SetFusion allows the user to combine are:

- *Most bookmarked papers*: this method recommends papers based on their popularity, i.e., papers that receive more bookmarks are ranked at the top.
- *Similar to your favorite articles*: this is a content-based recommendation method that considers the papers already bookmarked by the user to create a bag-of-words user profile. With this profile, the method matches the most similar non-bookmarked papers by cosine similarity. In order to make this method more effective, we tuned it using 10-fold cross validation and the final parameters considered filtering out terms with frequency less than three, appearing on less than two documents, and with a minimum length of four letters.
- *Frequently cited authors in the ACM Digital Library*: In this method, we recommended papers based on the popularity of their authors. Papers with authors that have been frequently cited in the ACM digital library are ranked at the top.

In SetFusion, users are provided with certain level of control over these methods: they can tune the importance of each prospect of relevance by adjusting their weight through sliders (Figure 2.b), an interaction method inspired by TasteWeights [2]. Despite these differences, the list of recommended items in SetFusion (Figure 2.a) can be filtered in a similar way to TalkExplorer, by clicking on the ellipse areas or their intersections (Figure 2.c).

Finally, users can interact with the Venn diagram as an inspection and filtering mechanism:

(a) *Hover over the circle*: Each small circle represents a talk, and hovering over one of them displays a dialog with the title of the talk (Figure 3.a).
(b) *Click on a circle*: By clicking in a small circle, the user will highlight the same element in the list of talks at the right side (Figure 3.b).
(c) *Click on a Venn diagram area*: Users can also click on the area surrounded by the big ellipses with white background, and by clicking on such an area, the visualization will become shaded as in Figure 3.c-1 and it will filter the list on the right side to the selected items (Figure 3.c-2).


Tune weights of the recommender methods:

Most bookmarked papers 0.4

Similar to your favorite articles 0.6

Frequently cited authors in ACM DL 0.4

Update Recommendation List →

* Hover over circles to explore articles

* Click on the diagram to highlight subsets

25. Challenging Information Foraging Theory: Screen Reader Users are not Always Driven by Information Scent

1. A Sentiment-enhanced Personalized Location Recomme
by Daqing Zhang, Zhiyong Yu, Zhu Wang, Dingqi Yang
Type: Long Paper
**Rate the relevance of this talk**: (not relevant) 1 2 3 4 5 (strong

3. On Commenting Behavior of Facebook Users
by Nazish Nasim, Aimal Rextin, Muhammad Usman Ilyas, Meh
Type: Short Paper
**Rate the relevance of this talk**: (not relevant) 1 2 3 4 5 (strong

5. Where's @wally? A Classification Approach to Geolocat
by Dominic Rout, Cohn Trevor, Daniel Preotiuc-Pietro, Kalina B
Type: Long Paper
**Rate the relevance of this talk**: (not relevant) 1 2 3 4 5 (strong

8. Studying the Impact of Personality on Users' Needs for F
by Kewen Wu, Li Chen, Liang He
his talk: (not relevant) 1 2 3 4 5 (strong

xt Narrative As City Planning
Type: Short Paper
**Rate the relevance of this talk**: (not relevant) 1 2 3 4 5 (strong

**Figure 3:** User interactions available on the Venn diagram of the SetFusion interface.

## USER STUDIES OF TALKEXPLORER

We have conducted two user studies with TalkExplorer. In the first study, we conducted a controlled experiment with users at two conferences (ACM Hypertext 2012 and UMAP 2012). The number of participants was 21. Users were asked to perform three tasks (exploring users, exploring agents and exploring tags). We recorded the screen and captured think aloud data. This controlled experiment enables to gain first insights into the relative effectiveness of each of these entities and to collect user feedback. Users had high familiarity with visualization techniques (mean 4.2, std. deviation 0.7) and a relatively high familiarity with recommendation techniques (mean 3.7, std. deviation 0.9). Details of this study have been reported in [12].

In the second study (N=18), we have deployed TalkExplorer again at two conferences and asked users to explore the visualization without any specific tasks. Users were free to interact with the visualization and were not required to use any specific components or controls. With this second study, we expected to gain insight into the usefulness of the visualization in an open setting. We wanted to find out how users explore and use the visualization without guidance and what attracts their interests. The analysis of interaction patterns yields less biased data, as users were not constrained to three separate and fixed tasks. In addition, the study was conducted at two conferences in the Technology Enhanced Learning field (EC-TEL 2012 and LAK 2013). Conference attendees have less technical knowledge than participants of the UMAP and Hypertext conferences of the first study. Most of the participants have again knowledge visualization techniques (average 4.23, std. deviation 0.79), but familiarity with recommendation techniques was less high (average 3.15, std. dev. 1.23).

To assess the value of interactive multi-prospect visualization offered by TalkExplorer, we have analyzed the way in which users *explore* and *use* the visualization. In the remainder of this section, we refer to selectable users, agents and tags as *entities* in the visualization. Papers or talks associated with these entities are referred to as *items.* We refer to *intersections* of entities when multiple entities were selected at the same time and their common items, displayed in clusters, were explored.

We measured the *effectiveness* of different combinations of entities to gain insight in the relative success rate of different combinations of entities to find relevant items. *Effectiveness* measures how frequently a specific combination type produced a display that was used to bookmark at least one interesting item. It is calculated as the number of cases where the exploration of this combination type resulted in a bookmark, divided by the total number of times this combination type was explored. For instance, the set of items of single entity (i.e. a user, a tag or a recommender agent) was explored 147 times by participants of study 1. Thirty-two of these sets were used to bookmark a new item. Thus, the effectiveness of exploring the set of items of a specific user is 32/147=22%.

Effectiveness results are summarized in Figure 4. Overall, these results indicate that effectiveness of an explored set increases once more entities are integrated. More specifically, effectiveness increases from 22% (user study 1) and 13% (user study 2) when a single entity is used to 52% (user study 1) and 50% (user study 2) when three entities are used. Effectiveness is significantly higher when multiple entities are used in both studies (p-value 0.003 in study 1, 0.0009 in study 2). These results illustrate that enabling users to explore interrelationships between prospects (sets of items in the overlap of entities) increases the probability of finding a relevant item.

Whereas both user studies demonstrated the clear value of multi-prospect visualization, we can't ignore one interesting difference. Despite the clear value offered by the intersection areas, the number of times that intersections were explored is lower in the second user study: items in the intersection of two entities were explored 28 times in the second user study (versus 53 times in the first user study) and items in the intersection of three entities were explored eight times (versus 29 in the first user study). Items in the intersection of four entities were not explored in the second study. The data are summarized in Figure 4.

Particularly the visualization of intersections of three or four entities seems to be non-intuitive or complex for end-users, as they do not tend to explore these intersections. In the first study, users explored these combinations more often and were more positive about the usefulness of this concept.

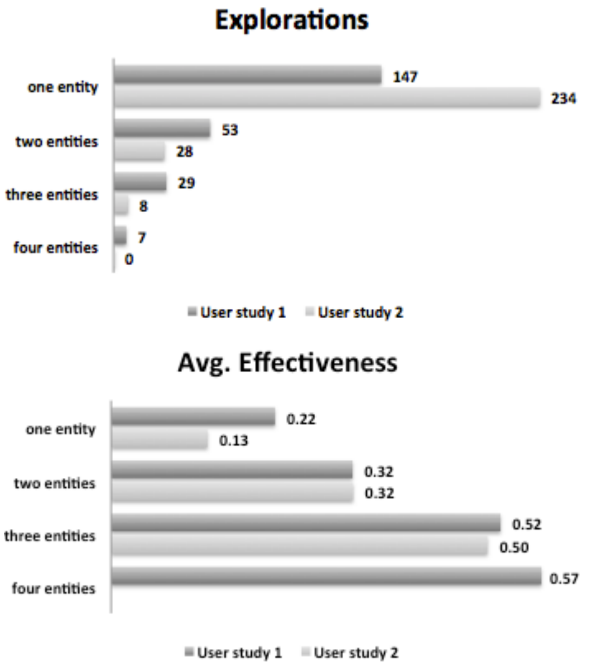


**Figure 4**: Summary results user study 1 and user study 2

To summarize, results of both studies illustrate the usefulness of visualizing multiple prospects. Users are interested to explore users, agents and tags and indicate that these multiple prospects are useful as a basis to find relevant talks. Exploring intersections increases effectiveness, but these intersections are not used often in an open setting.

A likely reason is the complexity of the TalkExplorer interface. A more intuitive way for exploring such overlapping sets are Venn diagrams, which are known to be both straightforward and standard to represents sets and set overlaps. In this paper, we are interested to explore whether it will help if we show overlaps in a more traditional and easy to understand way. SetFusion explores exactly the kind of interface that we believe could increase explorations of overlaps. We present user study results of SetFusion in the next section.

## USER STUDIES OF SETFUSION

In order to test whether the more intuitive representation of the Venn diagram had an effect on increasing CN3 users' engagement and effectiveness with the interface, we conducted a field study using SetFusion to recommend papers during the UMAP 2013 conference. In this study, users were free to access and explore the visualization.

The analysis of user participation and engagement data (Table 1) shows a good effectiveness of the interface in turning user exploration into bookmarked papers. The fraction of users who tried the SetFusion interface among those having a chance to use it was over 50% (50/95).

| Metric | SF UMAP13 |
|---|---|
| # Users exposed to recommendations | 95 |
| # Users who used recommender page | 50 |
| # Users who bookmarked | 14 |
| # Talks explored (user avg.) | 14.9 |
| # Talks bookmarked / user avg. | 103 / 7.36 |
| # People returning to recommender page | 14 (28%) |
| Average time spent in page (seconds) | 353.8 |

**Table 1:** Participation and engagement metrics in the SetFusion interface at UMAP13 conference.

The average number of each type of action in SetFusion during the UMAP 2013 field study is summarized in Figure 5. In parenthesis, the amount of users for each action is shown.

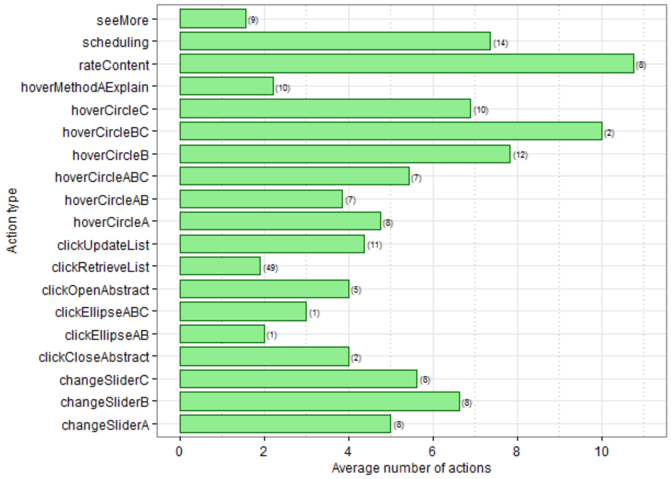


**Figure 5:** Average number of each type of action in SetFusion during the UMAP 2013 field study. In parenthesis, the amount of users performing those actions is shown.

These users explored 14.9 papers on average and bookmarked 7.36 papers, indicating a good level of effectiveness of the interface. Among the users that tried the SetFusion interface, 28% (14 users) bookmarked at least one paper. The same percentage of users came back to SetFusion page for a second time or more. If we consider the total time that users spent on the page among one or more sessions, users spent on average around 6 minutes (353.8 seconds) on the interface.  More detailed study results are reported in [7].

## META-ANALYSIS

The original work on TalkExplorer and SetFusion has been performed independently, with no intention to compare the results of our studies with these sufficiently different systems. At the same time, an extensive set of data collected in the mentioned studies opened an interesting opportunity to uncover the participation puzzle that we observed when comparing the results of two TalkExplorer studies. As presented above, results of our TalkExplorer studies indicate that effectiveness of recommendations increases in a significant way when users are able to interrelate multiple entities (see Figure 4). However, when deployed in an open setting, users do no explore such intersections often when a clustermap is used. The original paper that presents our work on TalkExplorer hypothesized that the likely reason for this phenomenon is the complexity of the TalkExplorer interface, especially the challenge of understanding the complex visualization of overlapping sets. While the Aduna visualization approach is very powerful and makes it possible to present multiple subsets created by overlapping three, four, five and more sets, understanding the picture is a real challenge. We suggested that this leads to the lower use of overlaps in the second study where the users were not specifically requested to do it. We also speculated that the "free" usage of overlaps could be increased when the users get more experience or when a simpler and more traditional visualization such as Venn diagrams will be used. However, at that time we were not able to provide any arguments in favor of these hypotheses.

The presence of the SetFusion study that was performed in the same system, with similar kind of data and using a similar approach, enabled us to re-assess this hypothesis. Indeed, SetFusion explored exactly the kind of interface that we believed could increase the usage of talks that are relevant for more than one prospect. Venn Diagrams are known as both a straightforward and a standard way (i.e., used in high school math classes) to visualize set overlaps. In this context, by re-processing the data of SetFusion study, we could provide some ground behind our complexity hypothesis. Below we present our attempt to re-process the data of the SetFusion study and present it in comparison with the data of the TalkExplorer study.

Figure 6 compares the number of times that sets were explored in all the presented user studies. TE-study 1 is the first (controlled) user study that we conducted with TalkExplorer. TE-study 2 is the second study with TalkExplorer that was conducted in an open setting: i.e. users were free to explore the visualization. Sets of a single entity were explored most in both studies: 147 times or 68% on average in the first study and 234 times or 84% on average in the second study. Sets representing items in the intersection of two entities were explored less often: 53 times or 16% in study 1, 28 times or 10% in study 2. Whereas items in intersections of three entities were still explored relatively often in study 1 (29 times or 11%), exploration of such sets was rare in the second user study: users explored intersections of three entities only eight times (6% on average). Intersections of four entities were not explored in the second study.

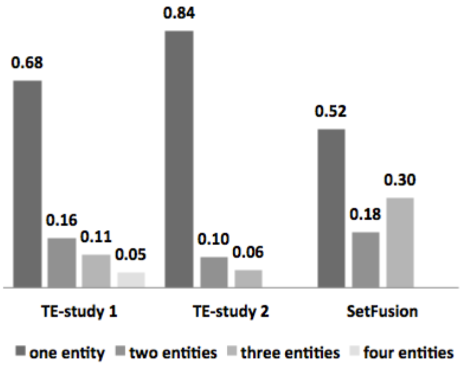


**Figure 6:** comparative analysis avg. number of explorations

Results of SetFusion draw a different picture. With a traditional Venn diagram, users explored items of a single entity in 52% of interactions. 18% of the interactions were explorations of intersections of two entities and 30% were explorations of intersections of three entities. It means that the use of two-entity overlap was higher than in the first TalkExplorer study where the users were specifically asked to do so. The use of three-entity overlap was almost three times higher than in the first controlled TalkExplorer study and five times more than in the second "free" study (TE-study 2).

Thus, our results indicate that with a more intuitive representation, the use of multiple relevance prospects is high even in a free exploration context where the users are not specifically required to use overlaps. There is no real difference between explorations of a single entity (52%) versus multiple entities (18%+30%=48%). Items in the intersection of three entities were explored more often than items in the intersection of two entities – which is an interesting result as such combinations were most effective for finding relevant items in our TalkExplorer studies.

The Venn diagram visualization therefore seems more promising than the clustermap visualization. As multiple entities increase effectiveness of recommendations, the approach would help users to explore those sets that help them find the more relevant items.  A drawback of the approach is that it is typically limited to three entities, whereas a clustermap enables to interrelate more than three entities. Despite this functionality, users did not explore such intersections in our second TalkExplorer study.

In summary, as results of our TalkExplorer study indicate that effectiveness of recommendations increases when multiple entities are interrelated, the Venn diagram approach is likely to better support our hypotheses. The data of the SetFusion study indicates that the approach is more intuitive for users – especially for interrelating multiple entities.

## CONCLUSION AND FUTURE WORK

In this paper, we have presented two approaches that enable end-users to explore recommendations. Both approaches allow end-users to combine *multiple* relevance prospects in order to increase the perceived relevance and meaning of recommendations. The first approach uses a clustermap representation and has been implemented in TalkExplorer. The second approach uses a Venn diagram and has been implemented in SetFusion.

In our user studies of TalkExplorer, we were able to show that effectiveness of recommendations increases significantly when multiple entities are interrelated. However, the clustermap visualization of TalkExplorer seems too complex to use. Users do not tend to explore those intersections that will help them find the more relevant items in an open setting. To make the power of overlaps work in a realistic context, the interface should be easy to understand. Venn diagrams are likely to be a good candidate, as they are known to be straightforward and a standard way for representing set overlaps. By re-processing the data of our SetFusion study that embodies exactly this kind of representation, we were able to show that users explore these intersections frequently. As indicated above, this exploration of overlaps is key, as it helps users to find the items that are likely to be more relevant to them.

In follow up studies, we will leverage this evidence and research more intuitive ways to support exploration of intersections. A follow up study will also include multiple agents (so far, only two agents were shown to the user) and assess the added value of our visualization on top of larger data collections.

## ACKNOWLEDGMENTS

We thank all the participants of the user studies for their participation and useful feedback. Research of Katrien Verbert was supported by postdoctoral fellowship grant of the Research Foundation – Flanders (FWO).

## REFERENCES

1. Aduna clustermap. http://www.aduna-software.com/technology/clustermap
2. Bostandjiev, S., O'Donovan, J. and Höllerer, T. TasteWeights: a visual interactive hybrid recommender system. In *Proceedings of the sixth ACM conference on Recommender systems* (RecSys '12). ACM, New York, NY, USA (2012), 35-42.
3. Gretarsson, B., O'Donovan, J., Bostandjiev, S., Hall, C. and Höllerer, T. SmallWorlds: Visualizing Social Recommendations. *Comput. Graph. Forum*, 29, 3 (2010), 833-842.
4. Gou, L., You, F., Guo, J., Wu, L. and Zhang, X. SFViz: interest-based friends exploration and recommendation in social networks. In *Proceedings of the 2011 Visual Information Communication - International Symposium* (VINCI '11). ACM, New York, NY, USA (2011), 10 pages.
5. O'Donovan, J., Smyth, B., Gretarsson, B., Bostandjiev, S., and Höllerer, T. PeerChooser: visual interactive recommendation. In *Proceedings of the twenty-sixth conference on Human factors in computing systems* (CHI '08). ACM, NY, USA (2008) 1085-1088.
6. Parra, D., Jeng, W., Brusilovsky, P., Lopez, C. and Sahebi, S. Conference Navigator 3: An Online Social Conference Support System. Poster at UMAP 2012. Montreal, Canada.
7. Parra, D., Brusilovsky, P., and Trattner, C. See what you want to see: visual user-driven approach for hybrid recommendation. In Proceedings of IUI '14. ACM, New York, NY, USA (2014) 235-240.
8. Riedl, J. and Dourish, P. Introduction to the special section on recommender systems. *ACM Trans. Comput.-Hum. Interact.* 12, 3 (Sept.2005), 371-373.
9. Schafer, J.B., Konstan, J. A., and Riedl, J. Meta-recommendation systems: user-controlled integration of diverse recommendations. In *Proceedings of the eleventh international conference on Information and knowledge management*, ACM, New York, NY, USA (2002), 43-51.
10. Spoerri, A. InfoCrystal: A visual tool for information retrieval & management. In *Proceedings of the second international conference on Information and knowledge management,* ACM (1993), 11-20.
11. Tintarev, N. and Masthoff, J. Designing and Evaluating Explanations for Recommender Systems. *Recommender Systems Handbook*, (2011), 479-510
12. Verbert, K., Parra, D., Brusilovsky, P., and Duval, E. (2013, March). Visualizing recommendations to support exploration, transparency and controllability. In *Proceedings of IUI'13,* ACM (2013), 351-362
13. Zhao, S., Zhou, M.X., Yuan, Q., Zhang, X., Zheng, W., and Fu, R. Who is talking about what: social map-based recommendation for content-centric social websites. In *Proceedings of the fourth ACM conference on Recommender systems* (RecSys '10). ACM, New York, NY, USA (2010), 143-150.

# A Visualization Interface for Twitter Timeline Activity

Wesley Waldner and Julita Vassileva
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada
{w.waldner, julita.vassileva}@usask.ca

## ABSTRACT

Social media streams are a useful source of current, targeted information, but such a stream can be overwhelming if there are too many sources contributing to it. In order to combat this information overload problem, rather than by filtering the stream, users may be able to more efficiently consume the most impactful content by way of a visualization that emphasizes more recent, popular, relevant, and interesting updates. Such a visualization system should provide means for user control over stream consumption while not excluding any information sources in the stream, allowing users to broaden their source networking without becoming overwhelmed. This paper presents a visualization for the Twitter home timeline that allows users to quickly identify which updates are most likely to be interesting, which updates they have and have not read, and which have been posted most recently. A small-scale pilot study suggests that improvements to the prototype are required before carrying out a larger-scale experiment. The effects of recommendation presentation on subjective measures of recommender accuracy will be studied as future work using this application as a framework.

## Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems – *human information processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering;* H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *user-centered design*.

## Keywords

Recommender systems; Social media; Social visualization

## 1. INTRODUCTION

In public social networks, where status updates can be viewed by any and all users of the system, a social activity stream is a useful tool that can help avoid information overload by collecting in a single location all updates from only those users in one's own social network. Social network users will typically connect with other users they are interested in, and, ideally, their activity stream will therefore consist of updates on topics that match their interest as well. However, it is impossible for all updates to be interesting or relevant to the user. Thus recommender systems can be introduced into social networks to serve two primary purposes. The first is to recommend additional sources of information to the activity stream, which involves adding nodes to one's social network. As the network grows, however, at some point throughput can become so great that it is impractical to consume every new piece of information flowing through the stream. In addition, the quantity of uninteresting content also increases with the interesting content. At this stage users have the option either to reduce the size of their network, resulting in a stream that is easier to handle, or to risk missing some particularly relevant or interesting updates.

The second common use of recommender systems in social activity streams is to try to avoid this problem by filtering the stream to show only the most relevant updates to the user. The ideal filtering recommender would reduce the stream throughput to a manageable amount, and would consistently predict with perfect accuracy the updates that the user would most like to consume. While it is unreasonable to expect perfection, such filtering mechanisms are intuitively useful in dealing with the information overload problem.

The stream filtering approach, however, has some potentially undesirable side effects [3]. Even if the recommender models a user's interests perfectly, she can become trapped inside a "filter bubble," engineered to match her interests at a particular point in time, but making it difficult to discover potentially new areas of interest. More realistically, the stream is also not being filtered perfectly. In either case, it can be difficult for the user to escape the filter bubble to receive serendipitous updates or expand her interests, especially since most filtering mechanisms do not provide much if any control to the user. When consuming filtered streams, users will also have a skewed perception of activity within their network. As preferences and interests may change over time, so too might the behaviour of other members in the network. If updates from these nodes are being filtered out of the stream, this may have unintended consequences on the user who might be interested in these activities but may never know of them because the nodes lie outside of her filter bubble.

Stream filtering, despite its shortcomings, is a commonly-used strategy for dealing with information overload in social activity streams. However, it is possible to emphasize certain updates without filtering others from the stream completely. In systems that show the entire stream by default without filtering, such as Twitter, each update is normally given equal visual prominence regardless of its popularity, relevance, or interest to the user. Therefore, the passive viewer cannot have any awareness of the popularity or social impact of posts just by consuming the basic stream. As a result, users will need to read each update to determine its relevance, at which point their time already will have been spent. Furthermore, if a user has not visited his stream in a while, he will be unable to catch up on the most important updates from that time period without consuming the entire stream.

A stream visualization that simultaneously depicts all updates from within a specific time range and differentiates between the most popular and impactful ones is a potentially useful alternative to stream filtering, as it allows users to explore more or less deeply depending on the amount of time they have available. By using a multi-dimensional nonlinear visualization that recommends and

*IntRS 2014*, October 6, 2014, Silicon Valley, CA, USA.

emphasizes the most important and interesting status updates for a particular user at a particular time, users will have increased awareness of the most impactful updates in their networks, will be able to consume time-relevant updates more effectively and efficiently without needing to filter their social streams, and will have increased trust in the system compared to a system without emphasis that filters out the least interesting updates.

# 2. BACKGROUND

## 2.1 Social Activity Stream Recommendation

There are a number of differences to consider when recommending for social activity streams versus traditional product recommendations. For one, there is usually a much larger amount of non-redundant data. For example, users may find thousands of social updates relevant at any given time. However, if a system is trying to recommend a new camera, the user is likely to buy only one and then not need any more help. Also, social updates may only be relevant for a very short period of time and may be targeted to a specific audience with special knowledge.

Though precision may be more important than recall in recommendations involving items that require a large commitment of time or resources [1, 7], recall intuitively seems to be more important when evaluating social activity stream recommenders. A small number of uninteresting updates appearing throughout the stream will not cost the user much time, perhaps as little as a few seconds, meaning that a lower level of precision may not cause much harm. Incorrect product recommendations, on the other hand, can have a greater negative effect. For example, if a user purchases an item that turns out not to be a good fit she may not be able to return the item to retrieve the money she spent. Conversely, it is undesirable to miss out on very important updates in a social activity stream, meaning that a lower level of recall may cause a great amount of relative harm. Ultimately, user satisfaction is the most important factor. Social activity streams are similar to subscription services in this way: there are no individual purchases to consider, and they interact with the system many times within a short span. What matters most is that people continue to use the system and have a good overall experience.

## 2.2 Visualization

Social visualization is an important aspect of recommender presentation that goes beyond the context in which items are presented and considers the structure that the presented data takes. When used in conjunction with a recommender system, social visualization can help the user understand how the recommender system is working [6]. There are many examples of systems that allow users to visualize their social networks[1]. These tools often simply map the connections between nodes without taking into account the activity of those nodes. However, previous studies have applied visualizations to the realm of social network activity and social activity streams. Some relevant examples are described in Section 6.

# 3. TWITTER STREAM VISUALIZATION

## 3.1 Main Idea

The main goal of this paper and future related work is to show that a multi-dimensional nonlinear visualization that emphasizes recommended content in a users' social activity streams will increase user awareness of impactful updates, increase user trust in the recommender system compared to one that employs filtering, and enable users to more effectively and efficiently consume the most relevant and interesting updates in their streams. To this end, we have developed an application that displays data collected from users' social activity streams in Twitter. The visualization represents updates as circles on a two-dimensional display, with different properties mapped to different visual dimensions (see Table 1 for a listing and Subsection 3.2 for full details). Recency and interest level, two important factors in supporting user awareness of the most relevant social network activity, receive the greatest focus and most prominent visual coding. However, to avoid misleading inferences about activity levels, no updates are filtered out of the system in this visualization, regardless of how irrelevant or uninteresting they may seem. In an effort to provide a more usable product, these updates are de-emphasized so as to be easier to ignore if the user so chooses. A content-based recommender learns from user behaviour and predicts the user's level of interest in every new update that appears in the stream. The visualization design supports chronological consumption of stream content, while highlighting the most relevant content to the targeted user and simultaneously depicting rises and falls in activity levels across the user's network.

## 3.2 Visual Design

### *3.2.1 Two-dimensional Timeline Visualization*

The backdrop for the stream visualization comprises a number of concentric circles about a central point. This point can be thought of as the immediate present. Each background circle, in increasing distance from this central point, represents an older point in time in the past. The distance between circles remains close to constant, but the time represented increases at greater distances from the centre to allow more room at the present where there is less angular spread and where users are more likely to focus their attention in order to read the latest updates. Thus the amount of time since an update was posted is coded in the visualization as distance from the centre. Because of the importance of size in the perception of visual prominence [2], Tweet relevance is coded with circle radius. With this combination of visual mappings, Tweets that are more recent and more relevant to the user will occupy more space close to the central region of the visualization. Appropriate default minimum and maximum values are in place to prevent unreadable results, and users are able to personalize the appearance so that it works best for the throughput level of their stream. More details on personalization options are discussed in Subsection 3.5.4 on the client implementation. The rest of the visual mappings are shown in Table 1.

**Table 1. Mappings between variables and visual dimensions**

| Variable | Visual Dimension |
|---|---|
| Recency | Distance from origin |
| Recommendation strength | Size |
| Popularity | Colour opacity |
| Unread/read | Shape (circle/horizontal line) |

Colour opacity was chosen for Tweet popularity, which is calculated as a normalized sum of the number of retweets and number of favorites. There is some concern that very popular Tweets, even when small due to a weak recommendation value, could dominate visually. However, popularity reflects social impact, which is an important factor for users to understand in order to be socially aware, so popular Tweets should be prominent.

[1] e.g.: http://keylines.com (general); http://mentionmapp.com, http://tweepsmap.com (Twitter); http://socilab.com (LinkedIn); https://immersion.media.mit.edu (email); http://www.touchgraph.com/facebook, https://friend-wheel.com (Facebook);

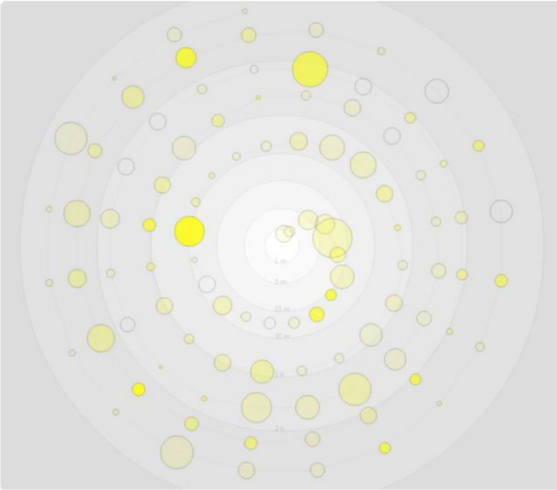
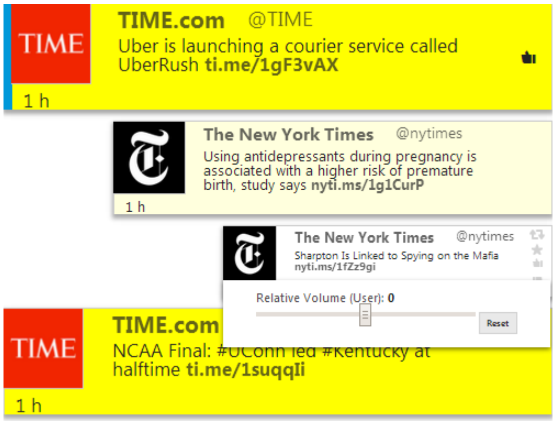


**Figure 1: Two-dimensional stream visualization**

Showing hundreds of complete Tweets onscreen at one time would of course cause overcrowding and would overwhelm the user; this is why circles are being used as placeholders. The actual content of the Tweets is hidden until the user's cursor hovers over one of the circles. On hover, a small card-like element will appear next to the cursor that displays the Tweet's content, including thumbnails of any embedded images, and the Tweet author's user name and avatar. Additionally, there is a linear stream panel that can be docked along the right side of the window. When the user interacts with a Tweet in either view, the corresponding Tweet in the other view (including the circle representation in the visualization) will be highlighted to help draw a connection between the two stream presentations. This may be helpful for a user who is reading the linear stream and wants to see the impact of a particular Tweet in relation to others around it. It also makes it easier to switch back and forth between views at any given time.

### 3.2.2 *Linear Textual Timeline Visualization*



TIME.com @TIME
Uber is launching a courier service called UberRush **ti.me/1gF3vAX**
1 h

The New York Times @nytimes
Using antidepressants during pregnancy is associated with a higher risk of premature birth, study says **nyti.ms/1g1CurP**
1 h

The New York Times @nytimes
Sharpton Is Linked to Spying on the Mafia nyti.ms/1fZz9gi

Relative Volume (User): **0**

TIME.com
NCAA Final: #UConn led #Kentucky at halftime **ti.me/1suqqIi**
1 h

**Figure 2: Screenshot of the textual linear stream showing all three recommendation tiers**

As mentioned, a textual timeline was also presented to complement the two-dimensional visualized timeline. Here, rather than using a continuous scale, recommendation scores are mapped to three discrete tiers. Tweets in the highest tier are larger in area as well as font size, have a stronger yellow colour, and are aligned further to the left. Tweets in the lowest tier are the smallest, have no colour, and are aligned further to the right, while the middle tier Tweets are in between the two extreme tiers in all qualities. The elements used to represent Tweets in this timeline view are exactly identical to the cards that pop up in the two-dimensional visualization, both visually and functionally. Tweets are displayed from top to bottom in chronological order, from newest to oldest.

## 3.3 Feature Design

Since this design builds upon the existing infrastructure of a major social network, features are already available to users for communication. However, these existing features have some limitations. A recommender needs a way to infer the utility of a particular item for a particular user. In Twitter, a user's appreciation for a Tweet can be explicitly indicated by a "retweet" or "favorite" action. One potential downside to these built-in actions is that they are completely public: any Twitter user can see which Tweets you have retweeted or favorite, which, depending on the situation, can be an incentive or deterrent to performing those actions. For the purposes of training a recommender, it would be preferable to have private ways of indicating interest for those situations in which a user might not want to publicize her opinion. Twitter also does not provide a way to indicate disinterest in a Tweet. To address these shortcomings, this application provides "like" and "dislike" functions, which are used exclusively to train the recommender. These two actions are denoted by familiar "thumb-up" and "thumb-down" icons.

Another feature, implemented to complement the recommender, is a manual user influence scale, which is shown in Figure 3 as "Relative Volume (User)". Users can manually adjust a recommendation multiplication factor that is effective across all Tweets by a particular member of their follow network by using a slider that can scale their influence up or down. For example, if the minimum influence level is chosen for User A, then all Tweets from this user will be shown as if they were given the minimum possible recommendation value from the recommender system. Similarly, if the maximum level were chosen, all Tweets from this user would be shown as the maximum recommendation level. The scale is quasi-continuous, and the chosen value is used as a multiplier as a final step after the initial value is passed from the recommender system running on the server.

A filtering feature was also added in order to test how trust in is affected when users, rather than the system, have full control over filtering. Users can move two sliders, one labelled "Min" and the other labelled "Max", to select a range of recommendation scores to allow through the filter. Setting the minimum value higher will exclude Tweets with low scores, while setting the maximum value lower will exclude Tweets with high scores.

## 3.4 Implementation Details

### 3.4.1 *Overview*

The software implementation of this application consists of three basic components: a client, server, and database. The server connects directly to the Twitter API and to the database and sends only the necessary updates to the client, which consists of the graphical user interface and visualization. A full-JavaScript software stack was used to develop the application.

### 3.4.2 *Recommender*

The recommender system implemented is similar to the one described by Wang et al. [9] to identify the most interesting updates from the Twitter user's home timeline. Users are given the ability through the graphical user interface to rate individual Tweets as interesting or uninteresting by clicking the "like" and "dislike" icons. These ratings are sent to the server and stored so that the recommender can be trained in the future as the user continues to give new ratings. As with any recommender system, more data is better: getting users to contribute ratings is one of the most important problems in social computing, but in this system users are encouraged to rate more and more Tweets as they see highly-rated Tweets that they are not interested in. These high ratings will appear to the user to be out of place, and with a single click they can be corrected. As new ratings are provided, the recommender will be re-trained and the interface updated; this quick feedback provides additional incentive to the users to continue training.

The recommender uses a naïve Bayes classifier trained using features from the rated Tweets stored in the database to predict whether unrated Tweets are interesting to the authenticated user. Then all unrated Tweets are classified as interesting or uninteresting. Using the Bayesian probability model, the posterior probabilities of the Tweet belonging to each of the two classes is calculated. The overall recommendation score from 0 to 1 is then determined by calculating the probability of the Tweet being interesting *given* the assumption, used for simplicity, that it is either interesting or uninteresting. Then, where $T$ is the Tweet being classified, $I$ is the set of interesting Tweets, and $U$ is the set of uninteresting Tweets, we have:

$$score = P\big(T \in I \big| T \in (U \cup I)\big)$$

Using the conditional probability formula for dependent events, we get:

$$score = \frac{P([T \in I] \cap [T \in (U \cup I)])}{P[T \in (U \cup I)]}$$

Since $T$ can only be an element of $I$ if it is also an element of $U \cup I$, the numerator can be simplified. The denominator can also be expressed as a simple sum because the sets $U$ and $I$ are mutually exclusive by definition. So we have:

$$score = \frac{P(T \in I)}{P(T \in I) + P(T \in U)}$$

In other words, the total recommender score is the ratio of the posterior probability that the Tweet is interesting to the sum of the posterior probability that the Tweet is interesting and the posterior probability that the Tweet is uninteresting. This will result in an average score (close to 0.5) when a Tweet fits equally well into either category and a more extreme score (closer to 0 or 1) when the Tweet fits into one of the two classes exceptionally well.

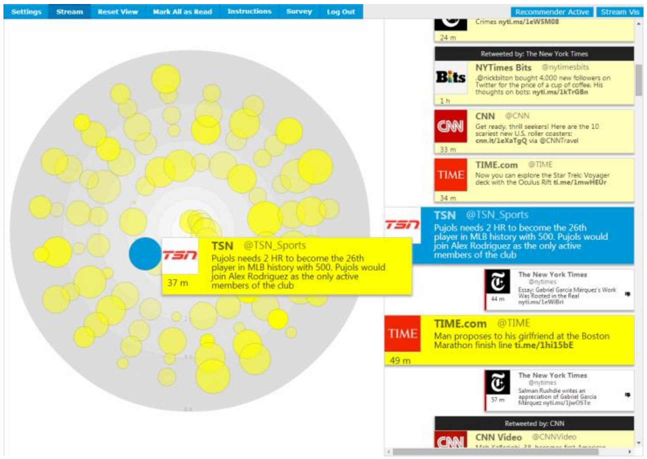
The following features are included in the classification procedure:

- Content author
- Content retweeter (if applicable)
- All hashtags
- All user mentions
- Tweet type(s): photo, link, retweet, reply, quote, manual retweet, and/or comment
- Number of retweets
- Number of favorites
- Length of text
- Number of numeric digits

The features are all used in an attempt to classify different types of Tweets. For example, a user may be partial to relatively long Tweets containing many numbers and no links that have been retweeted many times. The naïve Bayes classifier treats each feature as independent, however, so interactions between these features will not be accurately represented. A recommender that will take these interaction effects into account is left for future work. It would be interesting to try to classify Tweets based on topic to improve the recommender. Sriram [5] presents some promising work that uses text mining to classify different types of Tweets, while Wang et al. [9] used text mining to improve recommendations with similar machine learning techniques to those used here.

### 3.4.3 *Client*

HTML5 canvas was considered for rendering the visualization, but elements and event handlers would be easier to manage if each component was a node in the DOM tree. Instead, Scalable Vector Graphics (SVG) technology was used to allow for creation of vector images, which can scale to arbitrary sizes without losing detail. SVG elements are defined using XML and can be used in HTML5 markup just like regular DOM elements. Because all of the graphics are scalable, we added a feature that allows the user to zoom in and out to the position of the mouse cursor by scrolling the mouse wheel. This is perhaps the greatest benefit of using SVG instead of HTML5 canvas. Panning in the visualization is also allowed by clicking on an open area and dragging the cursor in any direction.



Crimes nyti.ms/1eW5M08
24 m

Retweeted by: The New York Times
NYTimes Bits @nytimesbits
.@nickbilton bought 4,000 new followers on Twitter for the price of a cup of coffee. His thoughts on bots: nyti.ms/1kTrGBn
1 h

CNN @CNN
Get ready, thrill seekers! Here are the 10 scariest new U.S. roller coasters: cnn.it/1eXaTgQ via @CNNTravel
33 m

TIME.com @TIME
Now you can explore the Star Trek: Voyager deck with the Oculus Rift ti.me/1mwHEUr
34 m

TSN @TSN_Sports
Pujols needs 2 HR to become the 26th player in MLB history with 500. Pujols would join Alex Rodriguez as the only active members of the club
37 m

The New York Times @nytimes
Essay: Gabriel García Márquez's Work Was Rooted in the Real nyti.ms/1eWiBri
44 m

TIME.com @TIME
Man proposes to his girlfriend at the Boston Marathon finish line ti.me/1hi15bE
49 m

The New York Times @nytimes
Salman Rushdie writes an appreciation of Gabriel García Márquez nyti.ms/1jwOSTe
57 m

Retweeted by: CNN
CNN Video @CNNVideo

**Figure 3: The client application running in a web browser**

# 4. PILOT STUDY

## 4.1 Goals

Before carrying out a large-scale quantitative study using this visualization tool, a smaller pilot study was necessary to identify pain points, streamline the experimental process, and determine the best way to collect the necessary relevant data. The pilot experiment tested the usability of the system and the appropriateness of the variable coding arrangement. Feedback was gained from the users on the following qualities of the system:

- Usefulness of the visual-emphasis approach to presenting recommendations
- Usefulness of user-controlled filtering feature
- Usability in general
- Sources of particular difficulty

## 4.2 Procedure

Two Twitter users were recruited via Facebook and were required to complete, in order, all of the tasks listed in this subsection.

### 4.2.1 *Explore and Rate Tweets*

Users were required to rate Tweets to train the recommender. To do this, they were instructed to read through either the textual or visualization timeline in chronological order, rating especially interesting and uninteresting Tweets along the way. Thirty ratings were sufficient to produce what users deemed to be accurate recommendations in a previous small-scale study using only the textual timeline with the three tiers of recommendation strength [8], so the recommender was activated after thirty ratings. At this point users were to make any necessary adjustments to the default settings now that the size of the Tweets had changed to reflect recommendation scores.

### 4.2.2 *Timeline Reading*

Users were instructed to traverse their timelines chronologically, reading *only* the emphasized Tweets, first using the textual timeline, and then using the two-dimensional visualization.

### 4.2.3 *User Volume*

In order to evaluate the usefulness of the User Volume feature, users were instructed to identify some users they wanted to see more or less of in their timeline and then to use the User Volume slider to make that user's updates more or less visually prominent.

### 4.2.4 *Filtered Timeline Reading*

Finally, users adjusted the Filter settings to test the recommender and visualization's joint effectiveness in another way. First they increased the minimum filter amount to show only the most highly-recommended Tweets, and then they reset and decreased the maximum filter amount to show only the least highly-recommended Tweets.

### 4.2.5 *Survey*

A link to a questionnaire appeared after the recommender became active. Users completed this survey as the final step in the study.

## 4.3 Survey Responses

The survey consisted of a 20-part questionnaire. The questions were broken down into the following categories:

1. Twitter usage
2. Recommendation presentation
3. Recommender performance
4. Design feedback

The results for categories 2–4 are outlined in the following sections. Responses for categories 2 and 3 were on a six-point Likert scale.

### 4.3.1 *Recommendation Presentation*

Users were asked the following set of three questions for both the textual stream and the visualized stream:

1. How easy was it to read only the most emphasized Tweets in your timeline?
2. How easy was it to ignore the de-emphasized Tweets in your timeline?
3. How easy was it to read through all Tweets in the timeline together in chronological order while the recommender was active?

Responses to these questions are shown in Tables 2 and 3.

**Table 2. Recommendation presentation responses for the textual stream**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Emphasized** | - | - | - | - | 1 | 1 |
| **De-emphasized** | - | - | - | - | 1 | 1 |
| **Combined** | - | - | - | 1 | 1 | - |

**Table 3. Recommendation presentation responses for the visualized stream**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Emphasized** | 1 | - | - | 1 | - | - |
| **De-emphasized** | - | - | - | 1 | 1 | - |
| **Combined** | 1 | - | 1 | - | - | - |

Generally the response to the textual recommendation presentation was very positive, while response to the two-dimensional stream visualization was mixed. Both users found it at least as difficult to read the entire stream chronologically in both cases as it was to read only the emphasized Tweets or ignore the de-emphasized Tweets. This can be considered a positive result because it suggests that recommendation emphasis may be a viable alternative to filtering for stream consumption.

### 4.3.2 *Recommender Performance*

With regard to recommender performance, the following questions were asked:

1. How accurate was the recommender in emphasizing interesting Tweets?
2. How accurate was the recommender in de-emphasizing uninteresting Tweets?
3. How strongly do you agree with the following statement? "As you increased the minimum Filter value, the application showed a generally more interesting timeline."
4. How strongly do you agree with the following statement? "As you decreased the maximum Filter value, the application showed a generally less interesting timeline."

Responses to these questions are shown in Table 4.

**Table 4. Recommender performance responses**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Interesting** | - | - | - | - | 2 | - |
| **Uninteresting** | - | - | - | 1 | 1 | - |
| **More Interesting** | - | - | - | - | 2 | - |
| **Less Interesting** | - | - | - | - | 2 | - |

Subjective evaluations of recommendation accuracy do not necessarily tell the whole story, but it is a very important component, especially in social activity stream recommendation. It is possible that an unbiased test of the recommender using pre-determined ratings in training and test sets and cross-validation would tell a different story and that users are more forgiving of recommendations that are slightly off or just better than the alternative. Users may especially be forgiving in this setting because reading an uninteresting Tweet causes little harm. The naïve Bayes classifier used here should infer preferences of users quite well if they follow others tweeting about only a narrow range of topics, but to get a more reliable indication of recommender performance using this subjective method of testing, a larger sample size is needed. On the other hand, the results are promising given the small amount of effort required to train the recommender.

### 4.3.3 *Design Feedback*

With regard to the user interface and feature design, the following questions were asked:

1. How useful was the "User Volume" feature?
2. Which timeline presentation style would you most prefer for regular use?
3. What did you like most about the user interface?
4. What did you like least about the user interface?
5. Which application feature did you like most?
6. Which application feature did you like least?
7. Do you have any other comments or suggestions?

The first question had responses on a four-point Likert scale, while the second question asked the users to choose between the textual and visualized versions of the timeline. The others were all text fields that allowed for open-ended responses.

Tweet interest can sometimes fluctuate greatly even within the set of Tweets from a given user. Because of this fact, it was unclear how helpful the "User Volume" feature would be, which allows users to manually adjust the influence of Tweet authorship on recommendation scores. However, both users reported that the feature was useful.

When asked which timeline presentation style they would most prefer for regular use, participants were given the choice between showing everything equally, showing everything with varying levels of emphasis, and filtering out the most uninteresting Tweets. Neither participant said that they would prefer everything to be shown equally, while each of the other two options was selected once. Without more participants these responses are not very useful, but it does suggest an appetite for users to have some processing done on the content in their stream, as not all updates are created equal.

The open-ended responses revealed some useful suggestions for future improvement. Users found the two-dimensional visualization relatively difficult to use and understand, suggesting that the presentation and interface could be more intuitive. The greatest source of trouble was lag due to frequent re-calculations in the application's script, which used the AngularJS framework. Significant performance enhancements may be possible, but has proven difficult without removing one of the visualizations from the page. Simplifying the two-dimensional visualization would reduce the need for so much processing to constantly be done. Creating a custom JavaScript framework optimized for this particular application would allow for maximum flexibility, but would require much more development time and would add much complexity.

In designing the visualization, we attempted to mitigate the performance problems by allowing users to limit the number of Tweets displayed on the page at one time, and in testing this seemed to work well. It is unclear whether the users missed reading about this feature in the instructions or if it did not have the same positive effect in their environments. It may also not be as practical in higher-throughput streams to limit the number of Tweets shown too much.

A larger sample size is desirable before writing off the two-dimensional visualization as a tool for stream consumption, but it would likely benefit from some design changes. It is possible that the visualization is better served as a complementary view to provide social activity awareness and a general view to support a primary linear textual stream. Some possible reasons users preferred the textual stream are that it supports a more passive browsing style, shows more information at one time, is more familiar, and contains larger targets for mouse interaction. More information will be gathered about the weaknesses of the existing system, and more usability testing will be done to improve it before carrying out a large-scale user study.

## 4.4 Limitations

The greatest drawback to this pilot study was the limited sample size. Of course a pilot study using even a small number of participants is more helpful than none at all, since it forces the designer to consider implications of releasing a system to the public further in advance. While many of the comments were very helpful, it is impossible to make any firm conclusions about the results gathered from the Likert-scale questions because of the small sample.

In general, the questions asked in the questionnaire were subjective and may have been positively biased, though some of the answers on the extreme negative end of the scale suggest this was not an issue for all participants. A quantitative study comparing the two presentation styles to measure interaction data, user preference, and subjective assessments of recommendation accuracy would be much more likely to avoid such biases and give more useful results.

# 5. PROPOSED EXPERIMENT

## 5.1 Goals

The main goal of the proposed large-scale experiment is to investigate the effects of recommendation presentation methods on users' subjective evaluations of the underlying recommender mechanism. In other words, we want to determine if the different ways of presenting social activity stream recommendations to users will affect how accurate they perceive the recommender to be. To measure this, metrics of trust, transparency, persuasiveness, effectiveness, efficiency, and satisfaction will be collected.

## 5.2 Design

In order to eliminate as many potential biases as possible, as well as to study interaction effects between different factors, a $2^2$ factorial experiment design will be used. Participants will randomly be assigned to one of two groups, one of which will use the visualized stream, while the other half uses the linear textual stream. Meanwhile, half of each of those groups will be divided by presentation methods of visual emphasis with user-controlled filtering or automated filtering where hidden updates are recoverable but not shown in the main timeline. The participants will have no knowledge of the existence of the other groups.

**Table 5. Treatment combinations for the proposed experiment**

| | |
|---|---|
| Textual Stream & Emphasis | Visualized Stream & Emphasis |
| Textual Stream & Filtering | Visualized Stream & Filtering |

In contrast to the brief pilot study conducted and described in this paper, the proposed experiment will take place over a period of two weeks, with participants using the system several times throughout that period. Several questionnaire responses will be required so as to measure the evolution of participant opinion over time. The questions will be similar to those used in the pilot study, but will focus more on recommender performance and trust and less on aspects of usability. User interaction data may also be collected and analyzed. We would like to recruit 100 participants so that an adequate sample size is reached for each factor group.

## 5.3 Expected Results

We expect that participants who use the systems with visual emphasis will rate the equivalent recommender system as being more accurate than will those using the systems with automatic filtering. Besides higher raw subjective scores for recommender accuracy, we expect to observe the following three results:

- Filtering will cause decreased trust
- Emphasis will cause increased transparency
- Emphasis will cause increased persuasiveness.

Trust, transparency, and persuasiveness, as they relate to recommendations, have been defined by Tintarev and Masthoff [6].

It is unknown whether the interface (textual vs. visualized) will have any effect, but any such effects will be observed. Participants may perceive more trustworthiness in the text stream case because less information is being "hidden" until the user interacts with the interface, but the visualization shows additional information that the text stream does not. For example, the visualization codes popularity and shows more data on the screen at one time. These factors may not be factors at all, or they may cancel each other out. Whatever the result, it will serve to guide future development of such systems for consumption of social activity streams.

## 6. RELATED WORK

As mentioned, the typical approach to the primary problem of information overload in social streams is to use some form of stream filtering. Naturally, there has been plenty of work done in this area, and several examples of stream filtering can even be found in the major social networking sites. Facebook's news feed, for example, reorganizes updates using an unknown algorithm of which post date is only one of multiple factors. It also is able to filter out particular updates, and this filter can be trained by user feedback. This method of reorganizing information, however, can mislead the users with respect to social activity since updates are not presented in chronological order.

The issue of recommendation in Twitter timelines in particular from a filtering approach has been tackled by Sriram [5]. In addition to a naïve Bayes classifier, C4.5 decision tree and sequential minimal optimization algorithms were used to classify Tweets into categories such as "news," "opinion," "deals," and "events." Support was also added for user-defined classes, which could be a useful addition to this project. Adding user-defined classes beyond just "interesting" and "uninteresting," but using the tiered model and visual emphasis instead of stream filtering could be a possible direction for future enhancement. Sriram attained a very high level of categorization accuracy using a more complex feature set that may be worth emulating in future work as well.

Wang et al. [9] also studied recommendations of updates across both Twitter and Facebook, focusing only on recommendation effectiveness without suggesting filtering as a solution to the information overload problem. They studied the value of textual and non-textual features in accurately predicting whether an update will be liked, disliked, or neutral. Machine learning algorithms such as decision trees, support vector machines, Bayesian networks, and radial basis functions were compared for performance. This paper was a helpful starting point for generating recommendations from basic features of social activity stream updates.

Some of the drawbacks of information filtering in social streams have been addressed by Nagulendra and Vassileva [3]. The "Filter Bubble" visualization in social networking site Madmica, shown in Figure 5, allows users to view which updates have been hidden, and it also gives control to show or hide posts on certain topics from certain users. However, it remains difficult to get a sense of where posts belong in the context of the social stream without restoring them to a visible status. This is likely not as important for Madmica as it is in Twitter, where updates may quickly become less relevant as they age, but it is one reason this project explores a complete view with emphasis rather than a filtered stream.

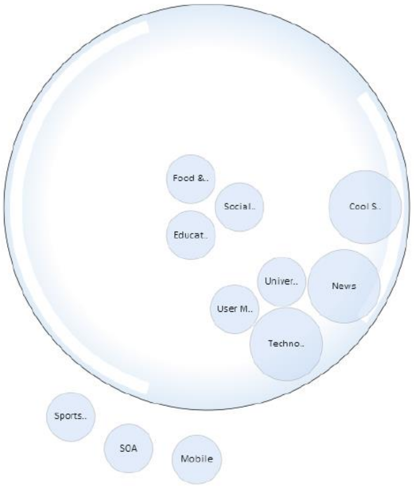
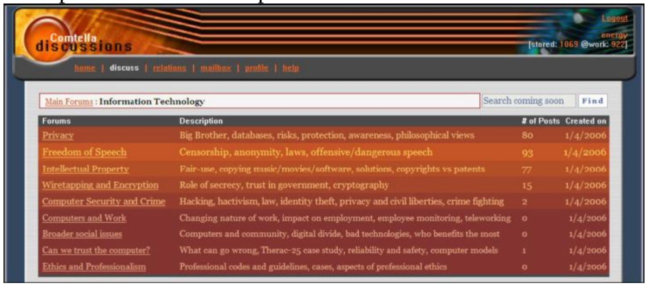


**Figure 4: Filter bubble visualization in Madmica**

Webster and Vassileva's work in the Comtella-D online discussion forum [11] was the original inspiration for the strategy of recommendation presentation using emphasis rather than filtering. In their system, recommendations are made collaboratively by and for other members of the community. The most recommended posts are shown in a brighter colour and with larger text in order to be visually attractive and more noticeable. The chosen colours in that case fit with an "energy" metaphor, with the more recommended posts displaying more life while the least recommended posts have a dull and lifeless appearance. However, a horizontal offset was not employed in this system, and the method of collaborative recommendations used within this closed community is not replicable in the vast open world of Twitter.



[stored: 1069 @work: 922]

Main Forums : Information Technology

| Forums | Description | # of Posts | Created on |
|---|---|---|---|
| Privacy | Big Brother, databases, risks, protection, awareness, philosophical views | 80 | 1/4/2006 |
| Freedom of Speech | Censorship, anonymity, laws, offensive/dangerous speech | 93 | 1/4/2006 |
| Intellectual Property | Fair-use, copying music/movies/software, solutions, copyrights vs patents | 77 | 1/4/2006 |
| Wiretapping and Encryption | Role of secrecy, trust in government, cryptography | 15 | 1/4/2006 |
| Computer Security and Crime | Hacking, hactivism, law, identity theft, privacy and civil liberties, crime fighting | 2 | 1/4/2006 |
| Computers and Work | Changing nature of work, impact on employment, employee monitoring, teleworking | 0 | 1/4/2006 |
| Broader social issues | Computers and community, digital divide, bad technologies, who benefits the most | 0 | 1/4/2006 |
| Can we trust the computer? | What can go wrong, Therac-25 case study, reliability and safety, computer models | 1 | 1/4/2006 |
| Ethics and Professionalism | Professional codes and guidelines, cases, aspects of professional ethics | 0 | 1/4/2006 |

**Figure 5: Visual emphasis of collaborative recommendations in Comtella-D**

Rings[2] [4] is a visualization system for Facebook friend networks that codes recency, quantity of recent posts, and average social impact of those posts. The system successfully increased user awareness of lurkers and the most active recent contributors in one's own network but did not focus on which individual posts were most impactful, choosing rather to focus on the users and their relative activity levels within the friend network. The information that the visualization provided was interesting for users and was not easily discoverable through Facebook's own default interface, but it was not necessarily useful for popular Facebook functions such as everyday social stream consumption. The visual design was the main inspiration for the visualization described in

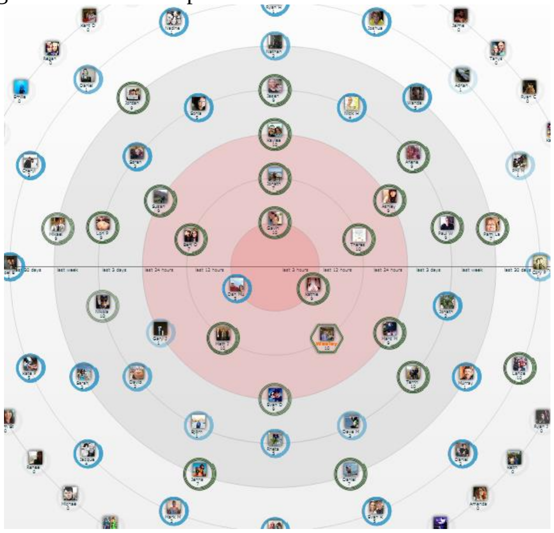


**Figure 6: Screenshot of the Rings Facebook visualization**

[2] http://rings.usask.ca

this paper. This new design also attempts to address some of the shortcomings of Rings by facilitating Twitter's typical use cases.

KeepUP [10] visualizes a user's network of influence in an RSS recommender system that allows for user interaction. While it does primarily model the network rather than the posts, it also tracks topics that each user has commonly liked or disliked. The transparency provided and affordance of user control over others' influence on recommendations allows users to shape their own filter bubble. The User Volume feature provided in the visualization system described in this paper was adapted from the idea that users can choose which members of their network should have the most influence on their recommendations.

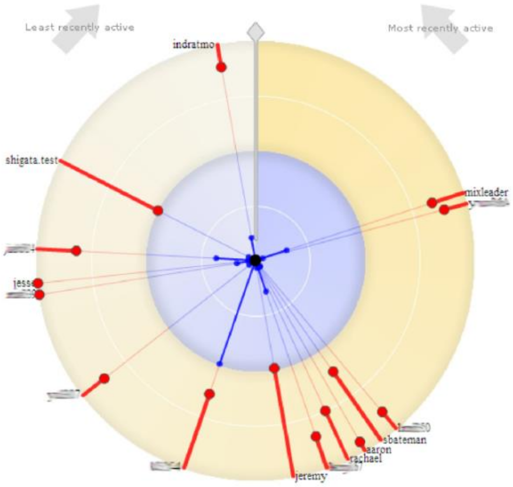


**Figure 7: Visualization of neighbour influence in KeepUP**

# 7. SUMMARY

This paper expands on work done in the area of social visualization and recommender systems by developing an application that can be used to study the effects of recommendation presentations on subjective measures of recommender performance. It is understood from the related work that visual emphasis can be a useful way to draw users' attention to more interesting or relevant content in social activity streams and that giving users control over stream filtering can increase their trust in these systems. The ultimate result that all of this is working toward is improved social activity streams wherein users spend more of their time reading the content best suited to them personally and are more aware of the full extent of activity in their social networks. Gaining a better understanding of the user and of how design decisions affect user opinions of the systems recommending and presenting that content is an important next step toward achieving those goals.

# 8. FUTURE WORK

Besides carrying out the experiment outlined in this paper, this application can be extended in a number of different ways for future research with the goal of understanding how best to increase user awareness and present recommendations of time-relevant updates in social activity streams. Potential future work that would extend or expand upon the research described here includes:

- Determining the optimal number of ratings required to strike the right balance between recommender effectiveness and user satisfaction.
- Improving Tweet classification in the recommender system, including accounting for interaction effects of classification features.
- Incorporating text mining to enhance classification and recommendation based on topics.
- Incorporating more user control by allowing users to specify why they liked or disliked a particular Tweet, including the ability to identify combinations of contributing factors.

The ultimate goal with this future work is to enhance the user experience through effective recommendations and presentations. Explanations and control are facets of recommender systems research that can lead to greater user acceptance, satisfaction, and trust in these systems. Applications of these facets to this unfiltered social activity stream recommender concept will be explored in greater detail in the future.

# 9. REFERENCES

[1] Gunawardana, A. & Shani, G. 2009. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *Journal of Machine Learning Research* 10, 2935–2962.

[2] Healey, C.G. 1992. Visualization of Multivariate Data Using Preattentive Processing. M.Sc. thesis, The University of British Columbia.

[3] Nagulendra, S. & Vassileva, J. 2013. Providing Awareness, Understanding and Control of Personalized Stream Filtering in a P2P Social Network. In *Collaboration and Technology: 19th International Conference, CRIWG 2013, Wellington, New Zealand, Oct. 30–Nov. 1, 2013, Proceedings*, 61–76.

[4] Shi, S. 2011. Keeping up with the Information Glut by Visualizing Patterns of Posting by Friends on Facebook. M.Sc. thesis, University of Saskatchewan.

[5] Sriram, B. 2010. Short Text Classification in Twitter to Improve Information Filtering. M.Sc. thesis, The Ohio State University.

[6] Tintarev, N. & Masthoff, J. 2007. A Survey of Explanations in Recommender Systems. In *ICDEW '07 Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, 801–810. DOI= http://dx.doi.org/10.1109/ICDEW.2007.4401070.

[7] Tyler, S.K. & Zhang, Y. 2008. Open Domain Recommendation: Social Networks and Collaborative Filtering. In *ADMA '08 Proceedings of the 4th International Conference on Advanced Data Mining and Applications*, 330–341. DOI= http://dx.doi.org/10.1007/978-3-540-88192-6_31.

[8] Waldner, W. & Vassileva, J. 2014. Emphasize, Don't Filter! Displaying Recommendations in Twitter Timelines. In *RecSys'14 Proceedings of the 8th ACM Conference on Recommender Systems*, Oct. 6–10, 2014. DOI= http://dx.doi.org/10.1145/2645710.2645762.

[9] Wang, Y., Zhang, J., & Vassileva, J. 2010. Towards Effective Recommendation of Social Data across Social Networking Sites. In *AIMSA'10 Proceedings of the 14th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, 61–70.

[10] Webster, A.S. & Vassileva, J. 2007. The KeepUP Recommender System. In *RecSys'07 Proceedings of the 2007 ACM Conference on Recommender Systems*, 173–176. DOI= http://dx.doi.org/10.1145/1297231.129726

[11] Webster, A.S. & Vassileva, J. 2006. Visualizing personal relations in online communities. In *AH'06 Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 223–233. DOI=http://dx.doi.org/10.1007/11768012_24.