EUROGRAPHICS 2012 / S. Bruckner, S. Miksch, and H. Pfister
(Guest Editors)

Volume 31 (2012), Number 3

TasteWeights:
A Visual Interactive Hybrid Recommender System

S. Bostandjiev[1], J. O'Donovan[1], and T Hollerer[1]

[1]Department of Computer Science, University of California Santa Barbara

Abstract

This paper presents an interactive hybrid recommendation system designed to personalize information flow from multiple social and semantic web resources, such as Wikipedia, Facebook, and Twitter for example. The system employs hybrid techniques from traditional recommender system literature, in addition to a novel interactive interface which serves to explain the recommendation process and elicit preferences from the end user. We describe an evaluation study which focuses on a range of interactive and non-interactive hybrid strategies for performing personalization across diverse social and semantic web APIs. Results of the study indicate that both explanation and interaction increase user satisfaction and relevance of predicted content.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Relevance Feedback

1. Introduction

The social web has become the dominant modality for distribution of media and collection of user-provided content such as text articles, feedback ratings, and comments for instance. Recommendation systems play an increasingly important role in this domain as they serve to filter and refine a user's information space according to their personal tastes and current requirements. However, social web APIs and other data sources are constantly evolving, and traditional recommender system techniques such as automated collaborative filtering (CF) [HKR00, RIS*94, SM95] need to be adapted to cater to the changing data environment on the social web. One simple example being that the traditional approach of pre-processing a large, static database of user ratings to produce a correlation matrix (i.e: the Netflix approach) to finding recommendation partners, can not be applied to user preference data on Facebook because of privacy restrictions in their API. However, as we demonstrate in this paper, with some adaptation to the CF algorithm, Facebook data can still be effectively harnessed to produce useful personalized recommendation in a collaborative manner.

The contributions in this paper examine the problem caused by evolving and emergent data sources for a recommender system. Specifically, we present two additions to the traditional processes of recommendation. First, a novel and synergistic approach to combining predictions from multiple sources on the social web, such as social (Facebook), content-based (Wikipedia) and expert-based (Twitter) recommendations. Second, we describe a novel interactive user interface which serves to both explain the provenance of recommended content in a transparent manner, and to elicit preference data and relevance feedback from users at recommendation time (video demo at http://vimeo.com/33389416).

To evaluate our approaches, we introduce *TasteWeights*, a hybrid music recommendation system with an interactive interface, allowing users to both understand and control aspects of the recommendation process that would otherwise go unnoticed. Figure 1 shows a snapshot of the interface, highlighting three social web data sources with a variety of weighting options, along with item recommendations on the right side of the interface. Using this system, a user evaluation was performed with 22 participants. The evaluation used participants' real social connections and music preference data. The study addressed the following core questions:

- What (if any) is the benefit of explanting a hybrid recommendation process through a UI?
- How does interaction at recommendation time affect accuracy and user experience?
- Does a hybrid strategy provide better recommendations than traditional CF (over Facebook music preferences)?
- Which of our three proposed hybridization/weighting strategies performs best?
- What is the impact of the hybrid UI on the diversity of recommended items?

While the *TasteWeights* system (Figure 1) is capable of recommending any media content listed in a Facebook profile, such as books, TV shows and movies for instance, recommendations described in this paper were restricted to music items in order to reduce complexity in our evaluations. The remainder of this paper is organized as follows: First, a brief introduction to *TasteWeights* is presented, followed by an overview of relevant related work is presented, focusing on recommender system interfaces and existing hybrid approaches. Second, we discuss the design and implementation of the *TasteWeights* recommendation system, in

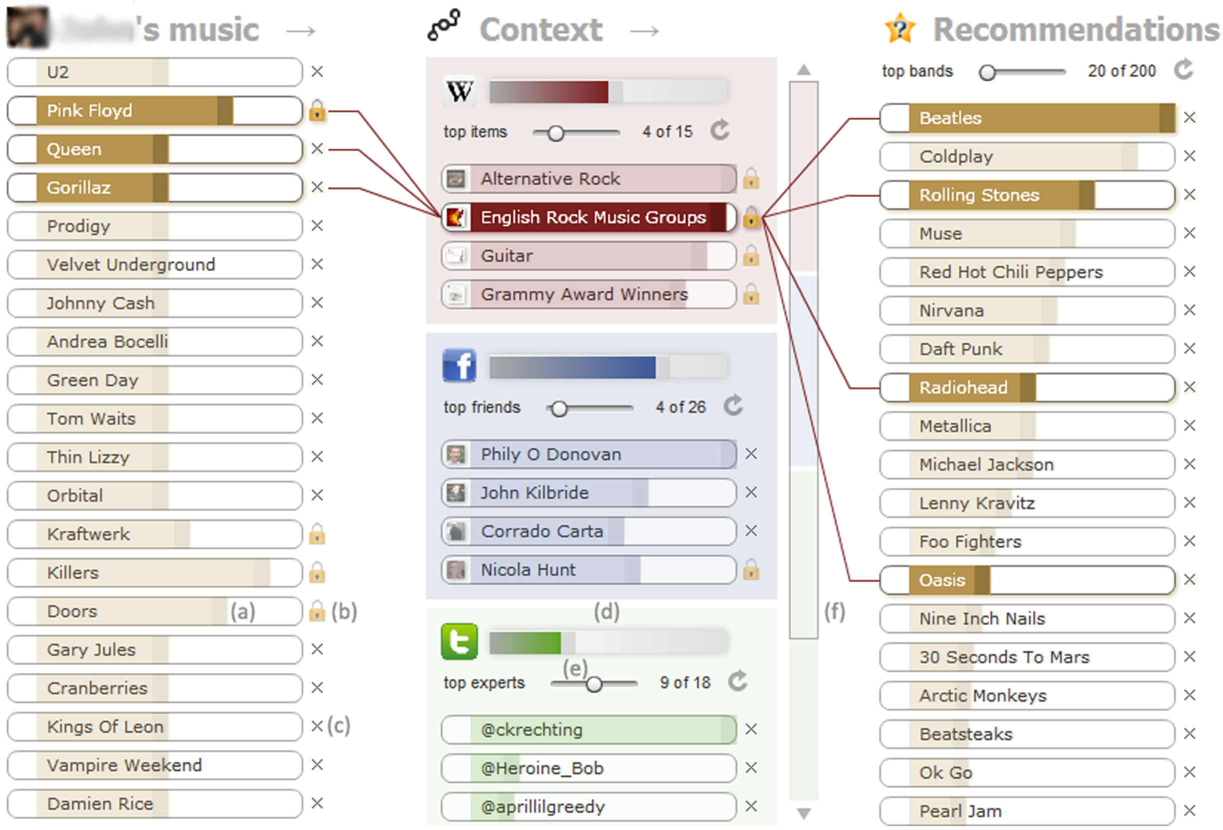

Figure 1: *Screenshot of TasteWeights illustrating the main interaction features: (a) changing the weight of an item (b) restoring the default value of an item (c) removing an item (d) changing the weight of a data source (e) changing the visible portion of a data source (f) navigating data sources*

terms of the underlying recommendation algorithms, hybridization strategies, user interface design and feedback to each algorithm. Next, a description of our evaluation is presented, and results are discussed to answer each of the five questions above. Finally, a summary discussion outlines the authors' perspective on the findings and highlights directions for future work in this area.

2. *TasteWeights* Overview

Figure 1 shows a screenshot of the *TasteWeights* system. The system is organized into three distinct layers and computational steps. These layers are represented as columns in the user interface:

1. *Profile Layer*: This leftmost layer contains the user's profile. In this case, liked music sourced from Facebook API. This layer supports re-rating of profile items through an array of sliders.
2. *Context Layer*: The central or "context" layer contains candidate recommendation partners grouped by domain. In this case, Wikipedia, Facebook and Twitter. These "partners" are any entity that can be used to produce recommendations, for example, a Facebook friend, Twitter expert or Wikipedia article in this case. This layer also contains sliders for weighting of partners, control of hybridization and weighting of domains.
3. *Recommendation Layer*: The rightmost layer contains the combined recommendations from each domain, ranked by relevance. Edges are displayed to illustrate the provenance of each recommended item, and relevance feedback is supported though an array of sliders.

As the system's name, *TasteWeights*, implies, users are encouraged to adjust their tastes via interactive slider-weights (Figure 1) and other UI components. While a user drags a slider, weights of the data items connected via outgoing links change accordingly in real time. For example, in Figure 1, as the user drags a slider for "Pink Floyd" to the right, the value of "English Rock Music Groups" increases simultaneously and so also do the values of Beatles, Rolling Stones, Radiohead, and Oasis. Section 4 describes our design decisions and methodologies in detail. Next, we present an examination of relevant related work in this area.

3. Related Work

Research related to this work falls into two main categories. 1) Recommender system algorithms and their hybrids, and 2) the role of interaction and visualization for recommendation systems. Accordingly, we now present a brief discussion of relevant works in these two areas.

3.1. Recommender Systems

For more than twenty years, research has been ongoing within the AI community on ways to automatically filter and personalize content for web users [SM95]. So called recommender systems aim to tailor a user's information by predicting the right item at the right time. There are a wide variety of approaches to recommendation, such as those algorithms we use through Amazon, YouTube, Netflix, Pandora and many other popular online applications that personalize content for users. Core techniques include content-based recommendation [MMN02a] [GRGP01] which is a rudimentary approach that simply matches text-based descriptions of a candidate item to those in a target users profile. These methods tend to suffer from problems such as narrowness, since they recommend items that are textually similar to those already in a user's profile. Content-based approaches also fall short on non-machine analyzable data such as music for example, where detailed text descriptions of content are generally not available.

By far the most common class of recommendation algorithms are automated collaborative filtering (ACF) approaches [BMZB05] [Kar01] [OWS02] [RIS*94] [SKKR01], ACF attempts to model the normal social process of asking a friend for a recommendation. In brief, ACF algorithms compute a neighborhood of users based on some correlation function (usually Cosine or Pearson's Correlation [OWS02] over vectors of rating data) and use that neighborhood to predict items that a user has not yet seen and that have been "liked" by his/her closest neighbors. ACF methods perform well in rich data environments where there is plenty of overlap in user preference. They are capable of making good predictions on non-machine analyzable data such as music, since they rely on human evaluation of the candidate items. For the same reason, these methods tend to be less narrow than content-based approaches, and can produce serendipitous recommendations [RIS*94]. Another relevant class of recommender system algorithms use knowledge from domain specialists to produce recommendations. These are commonly known as knowledge-based or "expert" recommender systems [RRSK11]. The hybrid models used in *TasteWeights* employ content-based, collaborative and expert-based recommendation strategies over a variety of social web data sources.

3.1.1. Hybrid Approaches

Traditional recommender system techniques such as collaborative, content-based, and knowledge-based filtering, each have unique sets of strengths and limitations. For example, CF suffers from sparsity and early rater problems [RIS*94], while content-based approaches suffer from narrowness and require text descriptions. However, a hybrid approach can use one approach to make predictions where the other fails, resulting in a more robust recommender system [MMN02b]. Burke [Bur02] proposes a taxonomy of different classes of hybrid systems and hybridization designs. For example recommendation algorithms can work in parallel before combining their results, may be pipelined such that the output of one algorithm is the input of the next, or may be combined into one monolithic algorithm. *TasteWeights* falls into the parallelized design class, since our approach firstly generates predictions from individual recommender system techniques, then applies a hybridization strategy afterwards.

3.2. Visualization for Recommender Systems

The focus of this paper is on a visual, interactive interface for a hybrid recommender system. That is, a visual control mechanism that informs users about the ways in which recommendations are combined from the methods discussed in the previous section. Through visualization we are creating an "explanation interface" for our recommender system, and moreover, our focus is to allow the end user to control aspects of the hybridization process through a simple informative and interactive interface. Some research has been carried out into the effects recommendation explanation has on the overall user experience with a system. A prominent work in this field is Herlocker's study of recommendation explanations [HKR00]. Herlocker et al. evaluate a "white box" conceptual model of recommendation as opposed to the run-of-the-mill black box approach. They present a user study where 21 different recommendation interfaces are presented to users, explaining various types of internal information from the recommender algorithm. Their general findings agree with Middleton's [Mid02], in that "explanation interfaces lead to improved acceptance of a predicted rating." Herlocker's work highlights several justifications for explaining recommendations through some form of interface, and those justifications also apply in our design decisions for the *TasteWeights* recommender system. According to Herlocker, explanatory interfaces

- help a user justify and understand the reasoning behind a recommendation, so that confidence can be decided.
- increase a user's sense of involvement. (i.e. keep the user "in the loop")
- educate the user about the recommendation process
- increase user acceptance of recommendations

In addition to these roles of an explanation interface, we posit that interaction can further aid in the recommendation process, namely by

- allowing users to dynamically update their preference profile at recommendation time
- enable user provided ratings directly on recommendation partners (potential producers of recommendations)
- support exploration of "what-if" scenarios based on different profile configurations.

Work in [OSG*08] focused on interactive visualization of genre information to elicit preference-feedback from users to enhance the quality of movie recommendations generated from a large scale data set. Gretarsson et al.'s SmallWorlds system [GOB*10] explored the effect of interactive visualization for a movie recommendation system. They found that an interactive interface helped produce more accurate recommendations and increase user acceptance of the predictions. While commonly associated with movies, music and online shopping, recommendation interfaces can be applied to a wide range of domains, for example, Crnovrsanin et al. apply visual recommendation to the task of network navigation in [CLWM11]. In their approach, collaborative filtering is applied to recommend potentially useful nodes in a network navigation task. The unique contribution in this work is an analysis of factors across both hybrid recommendation systems and interactive explanatory interfaces.

The following sections describe the design and methodologies used to produce hybrid recommendations in our *TasteWeights* system.

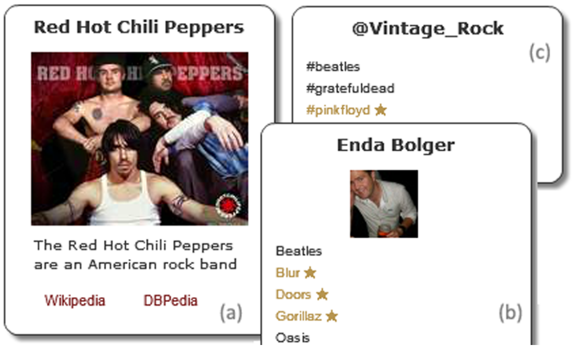

Figure 2: *Additional info shown when a data item gets clicked: (a) profile, recommendation, or Wikipedia item (b) Facebook friend (c) Twitter expert*

4. System Design & Interaction

Following Herlocker's guidelines in [HKR00], TasteWeight was designed to guide the user's understanding of how the hybrid recommender system works under the hood. Burke [Bur02] suggests that recommender systems have three distinct parts: input, background, and suggestions. *TasteWeights* follows a similar design structure, as shown by the three columns in Figure 1. Multiple UI controls allow users to fine-tune their preferences and receive immediate feedback on how their actions affect the output. Users are able to tweak the underlying algorithms by changing weights associated with individual items. As the user moves a slider associated with a weight they can see how that change affects the system as a whole. Once an item weight is changed it is "locked" to the user preference and is not affected by items from incoming connections unless it is actively unlocked by the user (Figure 1(b)). Individual items are enhanced by additional information when clicked, in a detail-on-demand fashion [Shn96]: profile, recommendation, and Wikipedia items are accompanied by an image and abstract, Facebook friends are shown with their profile photos and music profiles, and Twitter experts are accompanied by their items of expertise (Figure 2). Individual items can also be removed (Figure 1(c)). On a larger scale, users are able to express their relative trust in each data source by manipulating a slider for each data source (Figure 1(d)). The system provides dynamic recommendation feedback in real time while these interactions are being performed.

In order to emphasize the hybridity of the system, distinct colors for each data source are used as visual cues. The opacity of the data source box changes proportionally with the weight of the source expressed through its source slider. Any edges connected to a data source item inherit the data source's color. In some cases, the size of the data source column (middle column) can be large and extend beyond the height of the screen. To handle such cases, we have developed two UI features: Firstly, a slider to resize the visible portion of the data source (Figure 1(e)). Secondly, a scrollbar that reveals the current position within the column as a whole, and also expresses the relative source size through color coding (Figure 1(f)).

5. Data Sources

TasteWeights is a general solution that can be applied to a wide range of data sources on the social web. For our evaluations in this paper, we have chosen three popular data source APIs which we can categorize loosely into three different core recommender system techniques: Wikipedia (content-based / semantic). Facebook (collaborative / social), Twitter (expert-based). Before we proceed to discuss specific recommendation models, we present a brief overview of the properties of each data domain.

5.1. Wikipedia

Wikipedia is the most popular community-driven online encyclopedia, consisting of millions of user-provided articles, some of which are templatized and contain both free text and more structured, tabular data. DBpedia is a semantic web resource which crawls structured data from Wikipedia and organizes it into a database that is queryable through a SPARQL endpoint[†]. The database is an RDF store of subject-predicate-object triples, i.e. node-edge-node triples in terms of a link-node graph. Subjects and objects correspond to Wikipedia articles and each predicate is a labeled link between two articles. For example, the band "U2" is linked to the music genre "Alternative Rock" via a link labeled "genre". *TasteWeights* leverages Wikipedia by mapping music items in a user's profile to actual Wikipedia articles. For example, Pink Floyd corresponds to http://en.wikipedia.org/wiki/Pink_Floyd). We obtain the context layer Wikipedia items (the CF equivalent to recommendation 'partners'), which are Wikipedia articles or categories, by querying the DBpedia semantic database [ABK*08].

5.2. Facebook

Facebook is the world's largest online social network, with over 800 million active users [‡]. Although their API is limited by privacy restrictions, some music preference data is still accessible, in general, from direct friends of a user who is authenticated to the API. Facebook music preference data is used to bootstrap the *TasteWeights* system. The user's music profile items all map to specific pages that represent the artists. In the context layer of Figure 1, Facebook items are user's friends (potential recommendation partners) who have at least one liked item in common with the user, i.e. have similar tastes with the user. We query for this data through the Facebook Graph API[§].

5.3. Twitter

Twitter is a hugely popular microblogging service on the social web. Users can upload short text "tweets' through a variety of applications and devices. Twitter is commonly used for propagation of news events and for following expertise on various topics. Accordingly, we incorporate this service to produce expert-based recommendations for our *TasteWeights* system. Specifically, a user's music profile items can be mapped to hash tags. For example, "Pink Floyd" corresponds to the twitter hash tag *#pinkfloyd*). In our implementations, an online service from wefollow.com is used to find Twitter experts on the music in our users' profiles. wefollow.com is a user dictionary that curates lists of the most influential Twitter users for a large number of domains.

[†] http://dbpedia.org/sparql

[‡] http://www.facebook.com/press/info.php?statistics

[§] https://developers.facebook.com/docs/reference/api

6. Approach

Now that we have described each data source, we must provide a description of the various models used to gather data and predictions from each source. In the context of Figure 1, we can think of data and computations as flowing from left-to-right across the three columns. Each data item in the system is associated with a "score" (analogous to a weight) from 0 to 1 that is visually encoded in the slider bars.

6.1. Step 1: Profile Initialization

Facebook is used to source music preference information for every user. The list of music preferences in a profile are used as input to each of the computational models described here. Preference information for music on Facebook is binary– that is, no scaled preference rating is available. Accordingly, each profile item is initialized with a score of 0.5 on a scale of 0 to 1.

6.2. Step 2: Modeling Similarity

The three data sources provide different contexts (potential recommendation partners) for the music profile. Each data context requires a different model/strategy to extract these recommendation source entities (i.e: Facebook friends, Wikipedia articles, Twitter experts). The following sections describe each model individually.

Wikipedia Model Facebook music profile items are mapped to Wikipedia articles through dynamic queries over Google's Search API. For each profile item, a search is performed within the English Wikipedia[¶] and and the top result is selected. Next, (as we discussed in Section 5) a query is issued to DBpedia's SPARQL endpoint for items (articles and categories) that are linked to at least two music items in the active user's profile. This can be viewed as a content-matching approach to generating recommendations. An overall weight for each Wikipedia item (articles or categories) is calculated as the sum of the individual user-provided weights of the profile items it shares links with, as represented by the slider bars in the interface. This value can be represented by the following equation:

$$W_{wiki_i} = \sum_{Linked(profile_j, wiki_i)} W_{profile_j} \tag{1}$$

where $W_{profile_j}$ is the weight of a profile item j.

To generate predictions from our relevant Wikipedia sources, we examine the hyperlinks in each relevant source article and issue queries to DBpedia to retrieve new Wikipedia articles, filtered by type (i.e. "Musical Artist" or "Band"). For example, the article for "Pink Floyd" returns a link to an article about "The Beatles", so "The Beatles" becomes a candidate recommendation from this source.

Facebook Model Our recommendation strategy for Facebook is similar to traditional collaborative filtering, in that the opinions of similar friends are used to generated predictions. These friends are ranked according to their similarity with an active user's taste using a Pearson's correlation coefficient. We have adapted the correlation formula to account for the fact that Facebook items in users' music profiles are binary and do not contain scaled ratings. The similarity of each Facebook friend to the active user is given by:

$$W_{friend_i} = \frac{TWCI_{user,friend_i}}{\sqrt{TWI^2_{user} \cdot TWI^2_{friend_i}}} \tag{2}$$

where $TWCI_{x,y}$ is the total weight of the items x and y like in common, and TWI_x is the total weight of items liked by user x.

To generate predictions from Facebook, we apply a simple collaborative filtering approach wherein the items liked by similar friends (potential recommendation partners) are ranked according to the degree of similarity between the active user and the recommendation partner. Similarity is computed using Pearson's correlation between both users' music preference profiles.

Twitter Model In the Twitter domain, the goal is to source users that have expertise in the items listed in the active user's profile. To do this, we begin by mapping profile items to Twitter hash tags (i.e. Michael Jackson gets mapped to #*michaeljackson*) and so on. Next, we retrieve the top Twitter experts on those items according to wefollow.com for each hash tag. For example Pink Floyd experts are found here: http://wefollow.com/twitter/pinkfloyd. An expert user is classed as a potential producer of recommendations only if he is reported as an expert on at least one band that is not currently in the active users profile. In this manner, we ensure that there is some utility in every potential recommendation partner. For each expert found, recommendations are produced using the following equation to compute a score for each candidate item.

$$S_{expert,item} = \frac{|Experts_{item}| - Rank_{expert,item}}{|Experts_{item}|} \tag{3}$$

where $Rank_{expert,item}$ is the expert's ranking for the item and $|Experts_{item}|$ is the total number of experts for the item. For example, if an expert is ranked 20^{th} out of 100 experts for a specific item the expert gets a score of 0.8 for that item. The overall weight of a Twitter expert is determined by the linear combination:

$$W_{expert_i} = \sum_{Linked(profile_j, expert_i)} (W_{profile_j} \cdot S_{expert,profile_j}) \tag{4}$$

All hash tags that resolve to bands or musical artists that the relevant Twitter experts have knowledge in are potentially recommendable. Individual recommendations are computed over each source by the following equation:

$$W_{rec_i,source_j} = \sum_{Linked(rec_i, item_k)} W_{item_k} \tag{5}$$

Here, the weight of a recommendation i for source j is the sum of of the weights of all items within the data source that are linked

[¶] http://en.wikipedia.org

to the recommendation. In Section 7 we discuss a few different methods for combining the recommendation scores from individual sources. Notice that in order to do the hybrid step we need to create a mapping between recommendations coming from different data sources. For example, the Wikipedia article on the band "Asian Dub Foundation" corresponds to a page in the Facebook graph and to the Twitter hash tag #adf.

7. Hybridization Strategies

Up to now, we have focused on individual recommendation models for each of our three domains. In this section we begin our discussion of hybrid combinations of predictions across the different domains. As pointed out in Section 3, *TasteWeights* uses a parallelized design, that is, predictions are made by each source individually and then combined in a final processing step. Parallelized hybrids are further classified by Burke [Bur02] into mixed, weighted, and switching hybrids. We now present the three strategies used in *TasteWeights*: Weighting, Mixed and Cross-Source.

7.1. Weighted Hybrid

In this approach, a recommendation score is simply the weighted sum of the source recommendation scores for each domain. Weights for each domain are user-configurable through interactive sliders in the *TasteWeights* interface.

$$W_{rec_i} = \sum_{source_j \in sources} (W_{rec_i,source_j} \cdot W_{source_j}) \tag{6}$$

where W_{source_j} is the weight of source j.

Automatically optimizing the set of weights for each data source is desirable, but not trivial. Empirical bootstrapping can be used to calculate an optimal weighting scheme [ZJ09], however, historical data is needed for this approach. The P-Tango system looks into dynamic optimization of weights of a content-based and a collaborative recommender [CGM*99]. In their model, dynamic optimization starts with a uniform distribution of weights and dynamically adjusts the weights to minimize predictive error as users rate more items. This procedure can be applied on a per item and per user basis and the results can be combined and used for new users of the system. The evaluations presented in this paper do not use dynamic weighting, since the focus is on other interactive aspects of the system. For simplicity, our weights were fixed evenly across the three sources.

7.2. Mixed Hybrid

In this approach, recommendations for each source are ranked, and the top-n are picked from each source. This approach only considers relative position in a ranked list and does not include individual recommendation scores. In cases where the a recommendation is produced by multiple sources (i.e. was previously picked from another source) the algorithm simply selects the next recommendation from the ranked list for that source.

7.3. Cross-Source Hybrid

This approach strongly favors recommendations that appear in more than one source. We believe that if a recommendation is generated from more than one domain/algorithm, i.e. by both collaborative filtering (Facebook) and content-based (Wikipedia), then it should be considered as more important. To compute a final recommendation set, the weighted hybrid approach (Section 7.1) is first applied, then each recommendation's weight is multiplied by the number of sources in which it appeared. The following equation describes the the cross-source hybrid approach:

$$W_{rec_i} = \sum_{source_j \in sources} (W_{rec_i,source_j} \cdot W_{source_j}) \cdot |Sources_{rec_i}| \tag{7}$$

where $|Sources_{rec_i}|$ is the number of sources recommendation i was generated by (i.e. 1, 2, or 3).

8. Evaluation

We evaluated aspects of the *TasteWeights* system in terms of both recommendation accuracy and user experience. We compared nine methods: recommendations generated by the three individual sources (Wikipedia, Facebook and Twitter; cf. Section 8.2.1), recommendations produced by the three hybrid methods (Weighted, Mixed, and Cross-Source; cf. Section 7), and recommendations generated by three interaction variants that allowed users to fine-tune their preferences. The interaction variants differed based on how much of the recommendation process the users could reflect on:

Profile Interaction Users could only view and fine-tune the weights of items in their profile (the left column).

Sources In addition to profile tuning, users were able to change the weights on data sources items (middle column).

Full Interaction In addition to profile and sources tuning users could see the effects of their tuning actions on the recommendations (all columns were visible). Note, this is the default interface for the system.

The three different interactive methods could potentially use any of the hybrids as their underlying algorithm. To reduce complexity in our study, the best performing hybrid strategy was chosen for use in the three interactive methods. A pilot study consisting of 7 user trials was performed to find that the cross-sources hybrid outperformed the others.

8.1. Setup

To evaluate recommendation, explanation and interaction components, we performed a controlled user study with the objective of answering the research questions posed in our earlier discussion. 22 people participated in the study, which lasted approximately 47 minutes on average.

To assess the effects of explanation and interaction with the system on helping users understand the recommendation process we did a qualitative analysis based on a post-study questionnaire. We asked questions on how useful the explanation of hybridity was and how users perceived refining different aspects of the system.

We also performed a quantitative analysis on the performance of the nine recommendation methods. We used a within-subjects experimental design. The independent variable was recommendation method and the dependent variable was accuracy, measured in terms of "utility", described as follows.

Each of the nine methods produced a ranked list of recommendations. In order to compare the lists a utility-based ranking metric was applied. Utility of a given recommendation list is gven by the sum of the utilities of the individual recommendations. The utility of each recommendation is the utility of the recommended item discounted by a factor that depends on its position within the recommended list, i.e. recommendations in the beginning of the list carry more weight than those at the end. We use the *R-Score* metric by Breese [BHK98], which assumes that the value of recommendations decline exponentially based on position in the recommended list to yield the following score for each user u:

$$R_u = \sum_u \sum_j \frac{max(r_{ui_j} - d, 0)}{2^{\frac{j-1}{\alpha-1}}} \qquad (8)$$

where i_j is the item in the jth position, r_{ui} is user u's rating of item i, (i.e. 1 to 5 stars), d is Breese's "don't care" threshold (experimentally chosen as 2 stars in our setting), and α is the half-life parameter, which we set to 1.5, controlling the exponential decline of the value of positions in the ranked list.

8.1.1. Participants

In total, 22 users participated in the main study, 12 male and 10 female, ranging in age from 19 to 35. Participants were recruited through a university-wide experimental program and were paid a nominal amount of $10 for their time. Most participants were graduate or undergraduate students and spanned 10 different majors. Pre and post study questionnaires were completed by each participant. Most participants reported that they were regular Facebook users (86% daily, 10% weekly), and that they frequently used Wikipedia (36% daily, 45% weekly). There was a notable drop-off in reported use of Twitter in the study group, with 5% daily users, 18% weekly users and 63% who had never used the microblog. This drop-off may be attributed to the fact that Twitter is predominantly used for maintaining professional networks and for news propagation. Since our system is bootstrapped from a participant's Facebook music profile and associated network, probe questions were asked to assess the amount of available data. On average, participants had 415.6 Facebook friends (notably far larger than the average of 130 for the social network [‖], and Dunbar's optimal number of friend associations (150) . Participants reported that they were familiar with recommender systems such as Pandora and Netflix (3.8 out of 5 on a Likert scale). When asked about their primary methods for discovering new music, participants top choices was "Friends" (45%), then "Pandora" (36%) and "Radio" (23%).

8.1.2. Procedure

After completing the pre-study questionnaire, participants were given an oral explanation of the system controls and approximately 2 minutes to familiarize themselves with the various UI components. Once participants were comfortable with the system interface, they were asked to tweak the system using each of the three interactive methods, described in Section 8.1, which were presented in a random order across all users. After that users were asked to rate a randomized list of output from each of the nine tested methods. Each recommendation strategy produced a ranked list of 200 music recommendations. The purpose of this task was for participants was for participants to rate 15 recommendations produced by each approach on a 5 star Likert scale, 1 being the lowest and 5 being the highest. Outputs from all techniques were presented in a random order and ratings were performed in bulk at the end of the study. If the user didn't know enough about a recommended item to rate it the next item from the ranked list was picked to substitute it. This ensured that the user rates the top 15 items that they are familiar with for each method. Unfortunately, we were unable to assess unknown items (music that the users did not know), since we did not have sufficient time to play each piece of recommended music to elicit feedback. This is an important quality of the recommender system to assess, and we plan to evaluate it in a future user trial over a larger time period. After having rated enough recommendations users were asked to do a post-questionnaire and provide feedback on their perception of the system.

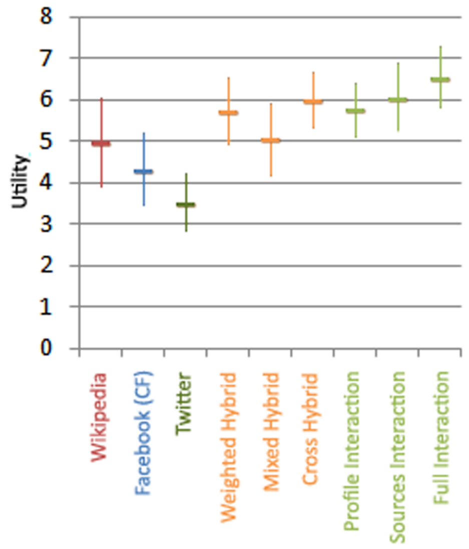

Figure 3: *Plot of means of recommendation methods over utility with 95% confidence intervals.*

8.1.3. Apparatus

The study was conducted on a Windows 7 computer (Intel Core 2 Duo T9600 CPU at 2.8Ghz with 4GB of RAM) and displayed on a 24-inch monitor with 1920x1080 resolution. The web application and database were hosted locally.

8.2. Recommendation Accuracy

Figure 3 presents a plot of the means of the nine methods with 95% confidence intervals. Overall the full interaction method was found to have the highest utility score, while the twitter method

[‖] http://www.facebook.com/press/info.php?statistics

Method 1	Method 2	Diff	Lower	Upper	P Val <
Cross Hybrid	CF	1.678	0.186	3.170	0.016
Sources Int.	CF	1.730	0.238	3.223	0.011
Full Interact.	CF	2.221	0.728	3.713	0.001
Full Interact.	Cross Hybrid	0.542	-0.949	2.035	0.964

Table 1: *Results from a Tukey post-hoc analysis of the recommendation methods: multiple comparisons of means with 95% family-wise confidence level.*

produced the lowest utility score. On average, the hybrid methods performed better than the individual ones, and the interactive methods performed better than the hybrids.

Mauchly's test showed a violation of sphericity against Method (W(35) = 0.005, p = 0.01). We ran one-way repeated-measure ANOVA and made Greenhouse-Geisser correction (ε = 0.46). It revealed a significant effect of the method variable on utility (F(3.72, 52.11) = 8.17, p < 0.01). To assess the statistical significance of pair-wise differences within our methods, a Tukey post-hoc analysis was performed and the results are presented in Table 1. Note that not all pair-wise results are shown but only relevant ones.

8.2.1. Single Source Results

Here, we examine the accuracy of predictions generated from each individual data source. To recap, we examined Facebook (collaborative/social filtering), Wikipedia (semantic/content-based filtering) and Twitter (expert-based recommendations). Interestingly, Wikipedia appears to perform best at producing recommendations with an average utility of 4.96, outperforming the collaborative Facebook prediction model by 15%. Initially this improvement was attributed to the fact that since Wikipedia contains far more detail on very popular bands and other music, it was recommending more popular items, and hence receiving higher ratings than Facebook predictions. However a further analysis of recommendation diversity was performed across each method, revealing that Wikipedia-based recommendations were in fact *more* diverse than Facebook-based recommendations. Detailed analysis of these results are discussed in Section 8.4. "Expert recommendations" sourced from Twitter exhibited the worst performance in terms of recorded accuracy. With an average utility of 3.52, this source was 22% worse than the Facebook source. Based on our observations, it appeared that recommendations derived from Twitter were more obscure than the other two sources.

8.2.2. Hybrid Results

Our second analysis focuses on a comparison of the three hybrid approaches to recommendation. The middle portion of Figure 3 shows the utility score for each approach (*weighted hybrid*, *mixed hybrid* and *cross-source hybrid*). All methods performed well relative to the CF benchmark. CF achieved an average utility of 4.32 (min 2,73, max 9.2). The cross-source hybrid approach, in which we favor recommendations coming from more than one source, showed a utility of 5.99 without any interaction, which is a large relative increase of 38% over the benchmark algorithm. The Tukey pair-wise test showed significant differences between CF and the cross-source hybrid method with p < 0.016. This is a strong indication that hybridization across social web APIs can help to increase predictive accuracy in recommender systems. Even though both weighted and mixed hybrid approaches exhibited higher utility means than CF, 5.71 and 5.05 respectively, there was no statistical significance. We believe that the improvement over CF is a result of increased data sourced from the APIs in the hybrid system (i.e: Twitter and Wikipedia).

8.2.3. Interaction Results

Three methods of interaction with the *TasteWeights* system were tested in the study and the results are shown in Figure 3. To recap, the methods are *profile interaction*: manipulation of ratings on pre-existing music items from the the user's Facebook profile; *source interaction*: direct manipulation of weights on the sources of recommendations, that is, users on Facebook or Twitter, and articles on Wikipedia that are used to produce recommendations; *full interaction*: interaction with all aspects of the system, with recommendations visible during interaction. The full method is the standard use case for the *TasteWeights* system. Figure 3 highlights an improvement in accuracy for interaction with the sources (p < 0.011) and interaction with the full system (p < 0.001) over the benchmark. The only anomalous result was that the *profile interaction* strategy was on average slightly less accurate than the cross-sources hybrid method, with scores of 5.99 and 5.75 respectively, however a Tukey post-hoc analysis showed that this result was not significant (p > 0.05) and may be a result of noise in the data. Results are shown in Table 1. For future work the authors will revisit this experiment with a view to achieving a significant result.

Interaction with the context layer (middle column), exhibited a small increase on average over the automated hybrid approaches. This may be due to the fact that with some context items, it was difficult to understand what items they tend to predict. For example, a user may be very aware of some Facebook friends' music tastes, but would have to do active research to learn about Wikipedia Articles which are used in prediction.

As expected, the full interaction method outperformed all others by a relatively large margin, achieving an accuracy score of 6.54, or a 51% increase over the benchmark. However, we note that this is clearly not a fair comparison since in this method, participants could see the recommendations change as they interacted with the system, meaning that recommendation feedback could inform their interactions. While this is not a fair scientific comparison, we posit that a mechanism which allows such informed, interactive feedback can be beneficial in real world recommender applications.

8.3. Explanation & Transparency

To assess the effects of interactive visualization on the perceived quality of recommendations, a post-questionnaire was completed by all participants. The study also analyzed the role of the interface as an explanatory mechanism for the underlying algorithms, and as a mechanism to help users learn about the underlying data. Looking firstly at factors affecting explanation and learning, the left side of Figure 4 shows on a Likert scale evaluation that users generally viewed the system as informative (4/5). The highest agreement value (4.4/5) was achieved for the "helped understand how I got my recommendations" question, indicating that the system is performing well as an explanation interface. Most participants found value in combining recommendations from different sources (4.1/5). Note that source provenance of each recommendation is preserved through interactive edges in the system. The average score for the "learned about Facebook friends' music

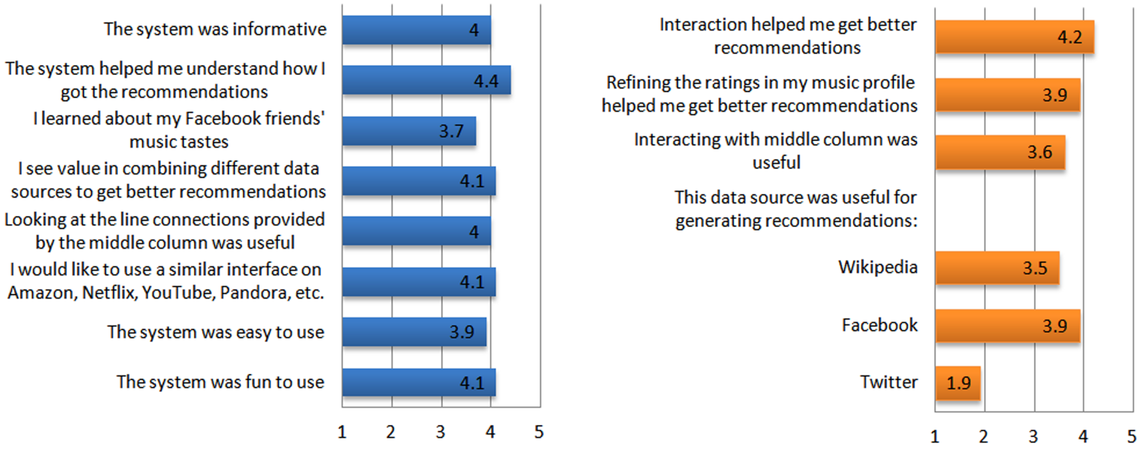

Figure 4: *Post-study questionnaire results: Explanation & Learning (left), Interaction (right)*

tastes" was 3.7/5, which, while still quite positive, was the lowest score for this part of the study. Most participants reported that they would like to see and use *TasteWeights* style interfaces on major recommendation systems such as Amazon, Netflix, Pandora and YouTube. (4.1/5). Importantly, most participants also reported that the system was fun to use (4.1/5).

The graph on the right side of Figure 4 shows results for the perceived quality of each prediction strategy. When compared with our empirical accuracy results, this graph yields some interesting results: Facebook was reported as the most useful source for generating recommendations (3.9 / 5), followed by Wikipedia (3.5/5), with Twitter reported as by far the least useful at 1.9/5. So, perceived usefulness shows a relative improvement of 9.7% for Facebook over Wikipedia, while accuracy from Figure 3 indicates the opposite trend, with Wikipedia outperforming Facebook by a factor of 14%. This increase in perceived utility of Facebook may be a result of participants favoring recommendations that come from real people who they trust and have prior information about, as opposed to "strange" items such as rich descriptions of content from Wikipedia or reported subject experts from Twitter.

8.3.1. Suggested Improvements

At the end of the study we asked participants to comment on what they liked and disliked about the system and suggest ways to improve it. The interface was perceived as aesthetically pleasing and users found it engaging to interact with items linked with lines. A few users found the Twitter source confusing and unintuitive. The most frequent request was to enable manual addition of profile items. In order to increase the quality of Wikipedia data users suggested grouping items in terms of types (i.e. music genres, countries, record companies), and removing seemingly irrelevant types, such as music award winners. Another frequent request was to enable adding trusted friends that do not have overlap in music taste with the user. It was also suggested to add a feedback loop, for example, changing the weight of a recommendation would change the weight of context items linked to it.

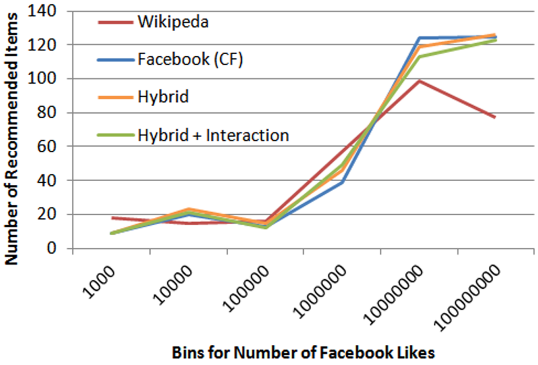

Figure 5: *Diversity of four approaches based on number of predictions in bins that represent number of Facebook 'likes'*

8.4. Recommendation Diversity

As discussed in recommender system literature [HKT*04], predictive accuracy is not a sufficient measure of the performance of a recommender system. Thus far, we have evaluated accuracy from an empirical and a perceptive viewpoint. The "popular item" effect is a common problem in recommender systems, where predictions tend to include many popular items and not enough esoteric items. However, it is the correct prediction of these esoteric items that can have the biggest positive influence on user satisfaction. To examine this phenomenon in the context of our *TasteWeights* system, a "popularity" score was computed for each recommended item. This score is simply the overall number of "likes" that the item had on Facebook. This number ranged from 0 likes for "Banda el Recodo" for example, to 45.4 million likes for "Lady Gaga", with an average of 3.7M and median 715K. This range was grouped into five popularity bins, and Figure 5 shows a graph of the number of predictions made by four of our methods in each bin. As expected, each method predicted far more popular items, ranging from 77 (Wikipedia) to 126 (Hybrid) for the most popular bin, and 9 (all except Wikipedia) to 18 (Wikipedia) at the least popular bin. From this analysis it appears that Wikipedia tends to recommend more obscure items than the other methods, while Facebook tends to predict more popular items, according to our "like" metric. Both the interactive and non-interactive hybrid methods fall between the other methods in terms of recommendation diversity, meaning that they reduce diversity of Wikipedia recommendations and increase diversity of recommendations from Facebook.

9. System Implementation

The system is implemented as a traditional client-server web application accessible through a Facebook application. The front-end is coded in Flash to provide rich user experience in the web using Adobe's Flex** open-source framework and the back-end is written in Java. MySql is used for data storage.

10. Future Work

TasteWeights is a complex system which combines facets from multiple research fields, most notably visualization and recommender systems research. In both areas there are many potential avenues to further explore the problems presented in this paper. For example, to further analyze the hybrid recommender components, it would be useful to compile a taxonomy of popular social web APIs in terms of their utility for generating personalized recommendations; their suitability for hybridization with other APIs, including their benefits and limitations. For instance, "Wikipedia data tends to increase diversity of recommendations"; "Twitter API tends to achieve low accuracy when used as a recommender system data source", etc. On the visualization side, the authors plan a larger scale, online study based around the *TasteWeights* system, to explore aspects of both visualization and interaction which are difficult to assess with the limited number of participants in supervised studies.

11. Discussion and Conclusion

This paper has presented *TasteWeights*, an interactive, hybrid recommendation system which sources recommendations from a range of social web APIs. A supervised user study was performed using the system to explore five research questions relating to visual interactive recommendation systems. The study results indicate that:

- Explaining a hybrid recommendation process through a UI can increase user satisfaction and accuracy (38% increase over a benchmark prediction algorithm).
- Interaction at recommendation time can improve perceived accuracy and user experience.
- Hybrid strategies can provide better recommendations than traditional CF (over Facebook music preferences).
- Our novel cross-source hybridization strategy significantly outperformed all other non-interactive methods tested in this study (with the exception that no significant difference was shown in the data for Wikipedia-based recommendations).
- The hybrid strategy tended to make recommendations from Wikipedia less diverse, while increasing diversity of predictions generated from Facebook.

References

[ABK*08] AUER S., BIZER C., KOBILAROV G., LEHMANN J., CYGANIAK R., IVES Z.: Dbpedia: A nucleus for a web of open data. In *Proceedings of 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)* (November 2008), pp. 722–735. 4

[BHK98] BREESE J. S., HECKERMAN D., KADIE C.: Empirical analysis of predictive algorithms for collaborative filtering. Morgan Kaufmann, pp. 43–52. 7

[BMZB05] BURKE R., MOBASHER B., ZABICKI R., BHAUMIK R.: Identifying attack models for secure recommendation. In *Beyond Personalisation Workshop at the International Conference on Intelligent User Interfaces* (San Deigo, USA., 2005), ACM Press, pp. 347–361. 3

[Bur02] BURKE R.: Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction 12*, 4 (2002), 331–370. 3, 6

[CGM*99] CLAYPOOL M., GOKHALE A., MIRANDA T., MURNIKOV P., NETES D., SARTIN M.: Combining content-based and collaborative filters in an online newspaper. In *In Proceedings of ACM SIGIR Workshop on Recommender Systems* (1999). 6

[CLWM11] CRNOVRSANIN T., LIAO I., WU Y., MA K.-L.: Visual recommendations for network navigation. *Computer Graphics Forum 30*, 3 (June 2011). (EuroVis 2011). 3

[GOB*10] GRETARSSON B., O'DONOVAN J., BOSTANDJIEV S., HALL C., HÃŰLLERER T.: Smallworlds: Visualizing social recommendations. *Comput. Graph. Forum 29*, 3 (2010), 833–842. 3

[GRGP01] GOLDBERG K., ROEDER T., GUPTA D., PERKINS C.: Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval 4*, 2 (2001), 133–151. 3

[HKR00] HERLOCKER J. L., KONSTAN J. A., RIEDL J.: Explaining collaborative filtering recommendations. In *Proceedings of ACM CSCW'00 Conference on Computer-Supported Cooperative Work* (2000), pp. 241–250. 1, 3

[HKT*04] HERLOCKER J. L., KONSTAN J. A., TERVEEN L. G., JOHN, RIEDL T.: Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems 22* (2004), 5–53. 9

[Kar01] KARYPIS G.: Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA* (2001), pp. 247–254. 3

[Mid02] MIDDLETON S. E.: Exploiting synergy between ontologies and recommender systems, semantic web workshop 2002 hawaii, usa, 2002. 3

[MMN02a] MELVILLE P., MOONEY R., NAGARAJAN R.: Content-boosted collaborative filtering. In *In Proceedings of the Eighteenth National Conference on Artificial Intelligence* (2002). 3

[MMN02b] MELVILLE P., MOONEY R. J., NAGARAJAN R.: Content-boosted collaborative filtering for improved recommendations. In *Eighteenth national conference on Artificial intelligence* (Menlo Park, CA, USA, 2002), American Association for Artificial Intelligence, pp. 187–192. 3

[OSG*08] O'DONOVAN J., SMYTH B., GRETARSSON B., BOSTANDJIEV S., HÖLLERER T.: Peerchooser: visual interactive recommendation. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2008), ACM, pp. 1085–1088. 3

[OWS02] O'SULLIVAN D., WILSON D. C., SMYTH B.: Improving case-based recommendation: A collaborative filtering approach. In *Proceedings of the Sixth European Conference on Case Based Reasoning.* (2002), pp. LNAI 2416, p. 278 ff. 3

[RIS*94] RESNICK P., IACOVOU N., SUCHAK M., BERGSTROM P., RIEDL J.: Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work* (1994), pp. 175–186. 1, 3

[RRSK11] RICCI F., ROKACH L., SHAPIRA B., KANTOR P. B. (Eds.): *Recommender Systems Handbook*. Springer, 2011. 3

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations, 1996. 4

[SKKR01] SARWAR B. M., KARYPIS G., KONSTAN J. A., REIDL J.: Item-based collaborative filtering recommendation algorithms. In *World Wide Web* (2001), pp. 285–295. 3

[SM95] SHARDANAND U., MAES P.: Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM*

** http://www.adobe.com/products/flex.html

CHI'95 Conference on Human Factors in Computing Systems (1995), vol. 1 of *Papers: Using the Information of Others*, pp. 210–217. 1, 2

[ZJ09] ZANKER M., JESSENITSCHNIG M.: Case-studies on exploiting explicit customer requirements in recommender systems. *User Model. User-Adapt. Interact. 19*, 1-2 (2009), 133–166. 6