TasteWeights: A Visual Interactive Hybrid Recommender System

Svetlin Bostandjiev, John O'Donovan, Tobias Höllerer
Department of Computer Science
University of California, Santa Barbara
{alex, jod, holl}@cs.ucsb.edu

ABSTRACT

This paper presents an interactive hybrid recommendation system that generates item predictions from multiple social and semantic web resources, such as Wikipedia, Facebook, and Twitter. 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 computing recommendations across diverse social and semantic web APIs. Results of the study indicate that explanation and interaction with a visual representation of the hybrid system increase user satisfaction and relevance of predicted content.

Categories and Subject Descriptors

Information Storage and Retrieval [**H.3.3**]: Information Search and Retrieval - Relevance Feedback

Keywords

User Interfaces, Visual Knowledge Representation, Hybrid Recommender Systems, Data Integration, Social Web

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) [**?**, **?**, **?**] need to be adapted to cater to the changing data environment on the social web. One simple example being that the traditional approach of preprocessing 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.

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 **??** 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. A video demonstration of the system can be watched at [1]. Using this system, a user evaluation was performed with 32 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 explaining a hybrid recommendation process through a user interface?
- How does interaction at recommendation time affect accuracy and user experience?
- Can a hybrid strategy combining different social APIs provide better recommendations than traditional CF (over Facebook music preferences)?

While the *TasteWeights* system (Figure **??**) is capable of recommending any media content listed in a Facebook profile, such as books, TV shows, and movies, recommendations described in this paper were restricted to music items in order to reduce complexity in our evaluations.

RecSys '12 Dublin, Ireland
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$10.00.

[1] `http://youtube.com/watch?v=9_JgynePm9w&hd=1`

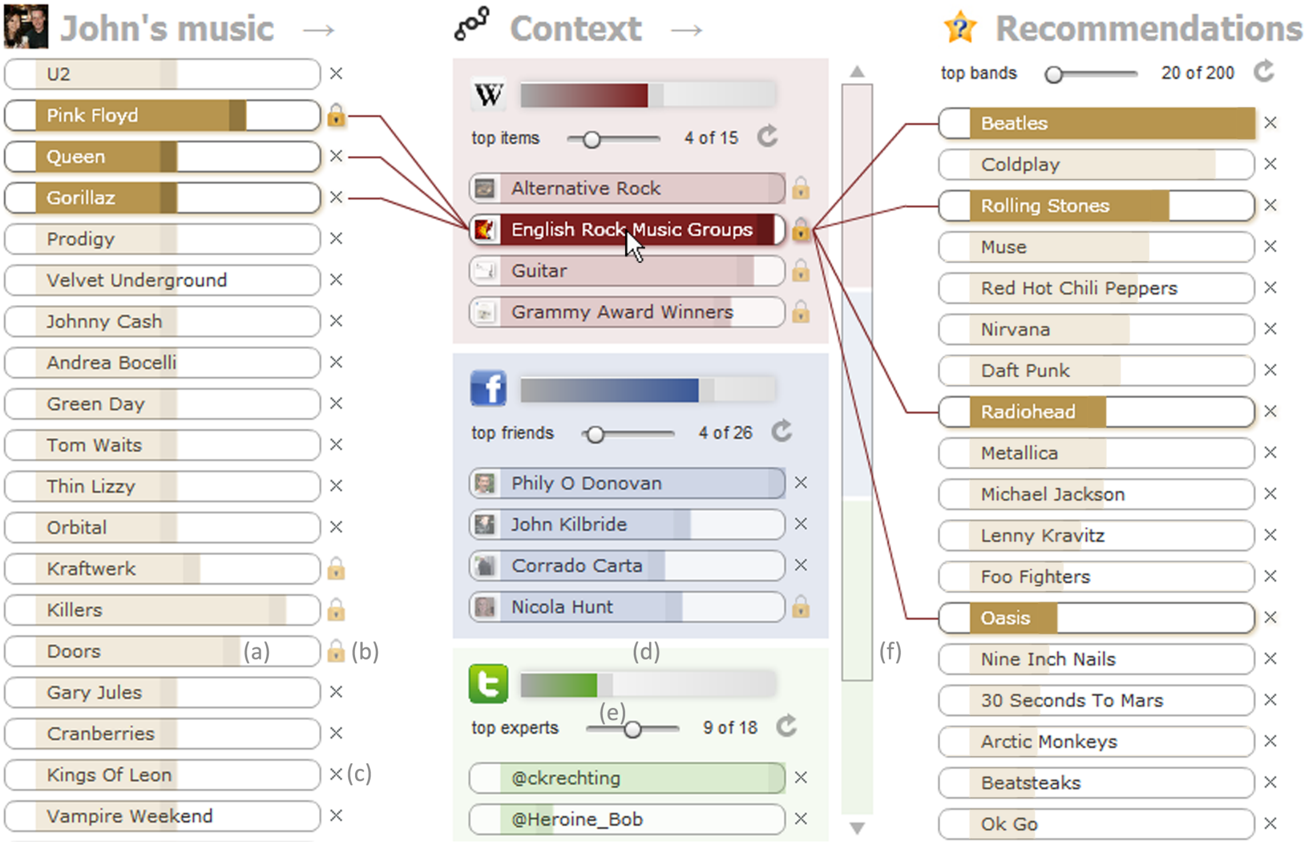


Figure 1: Screenshot of *TasteWeights* illustrating the main interaction features: (a) changing the weight of an item (b) restoring the default value of an item (c) removing an item (d) changing the weight of a data source (e) changing the visible portion of a data source (f) navigating data sources

2. TASTEWEIGHTS OVERVIEW

Figure **??** shows a screenshot of the *TasteWeights* music recommender system. The system is organized into three distinct layers and computational steps.

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 items coming from different data sources, in this case, Wikipedia, Facebook and Twitter, that can be used to produce recommendations. This layer also contains sliders for weighting of these items, and control of hybridization through weighting of sources.

3. *Recommendation Layer*: The rightmost layer contains the combined recommendations from each source, ranked by relevance. Edges are displayed to illustrate the provenance of each recommended item.

As the system's name, *TasteWeights*, implies, users are encouraged to adjust their tastes via interactive slider-weights 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 **??**, 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 **??** 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 the categories of hybrid recommender systems and the role of interaction and visualization for recommendation systems in general.

3.1 Hybrid Recommender Systems

Traditional recommender system techniques such as collaborative filtering (CF) [?, ?, ?, ?, ?], content-based [?, ?], and knowledge-based filtering [?], each have unique sets of strengths and limitations. For example, CF suffers from sparsity and cold start problems [?], 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 [?]. Burke [?] 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

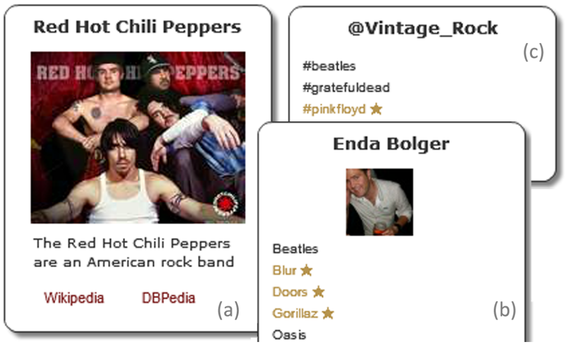
The focus of this paper is on a visual, interactive interface that supports control of a hybrid recommender system. Through visualization we are creating an "explanation interface" for our recommender system, and moreover, our focus is to allow the end user to control aspects of the hybridization, and other elements in the recommendation process through a simple informative and interactive interface. So called "social recommender system" are distinct from traditional collaborative sytems becuase they compute a set of neighbors out of a set of friend's that have pre-existing social connections. We believe that explanation and transparency is particularly important in these systems because the user can associate rich social knowledge of a neighbor at a quick glance. Some research has been carried out into the effects recommendation explanation has on the overall user experience with a system, from how and why analysis [?], to iterative "conversational" explanations such as in [?, ?]. Kay and Lum describe concepts of scrutability [?] as a part of the explanation process, in terms of providing explanations of why individual elements and relations in the underlying model have particular values. A prominent work in this area is Herlocker's study of recommendation explanations [?]. 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 [?], 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 users justify and understand the reasoning behind a recommendation, so that confidence can be decided
- increase users' sense of involvement. (i.e. keep the user "in the loop")
- educate users about the recommendation process
- increase users' 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 during a recommendation session
- enabling users to provide ratings directly on the entities used to produce recommendatons
- supporting exploration of "what-if" scenarios based on different profile configurations

Work in [?] 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 [?] 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. Crnovrsanin et al. apply visual recommendation to the task of network navigation in [?]. 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 Red Hot Chili Peppers are an American rock band

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 [?], *TasteWeights* was designed to improve the user's understanding of how the hybrid recommender system works under the hood. Burke [?] 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 **??**. 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 (Figure **??**(a)). 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 **??**(b)). Individual items are enhanced by additional information when clicked, in a detail-on-demand fashion [?]: 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 **??**). Individual items can also be removed (Figure **??**(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 **??**(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. The size of the data source column (middle column) usually extends beyond the height of the screen. To handle this, we have developed two UI features: Firstly, a slider to resize the visible portion of the data source (Figure ??(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 ??(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 social 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 source.

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. We query Wikipedia for articles and categories that are most relevant to the user's music profile. The results are presented in the top part of the middle (context) layer in Figure ??. We find relevant Wikipedia articles indirectly through DBpedia [?], a semantic web resource that crawls structured data from Wikipedia and organizes it into a database that is queryable through a SPARQL endpoint[2]. The database is an RDF store of subject-predicate-object triples. 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" profile item corresponds to `http://en.wikipedia.org/wiki/Pink_Floyd`).

5.2 Facebook

Facebook is the world's largest online social network, with over 800 million active users [3]. 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 ??, Facebook items are a user's friends who have at least one liked item in common with the user, i.e. have similar tastes to the user. This data is mined through the Facebook Graph API[4].

5.3 Twitter

Twitter is a popular Social web microblogging service. 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 items in the user's music profile. wefollow.com is a user dictionary that curates lists of the most influential Twitter users for a large number of domains.

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 data source. In the context of Figure ??, 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.

Step 1: Profile Initialization. To initialize a user profile, music preference information is gathered though the Facebook graph API. The list of music preferences in the user's profile are used as input to each of the source-specific computational models described in Section ??. 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.

Step 2: Modeling Similarity. The three data sources provide different source items (middle column in Figure ??) that can potentially generate recommended items. Each data source requires a different model/strategy to extract these source items (i.e: Facebook friends, Wikipedia articles, Twitter experts). Those are described in ?? .

Step 3: Generating Predictions. Once relevant items have been collected by each source, the next step is to generate predictions. Individual recommendations are computed over each source by the following equation:

$$W_{rec_i,source_j} = \sum_{Linked(rec_i,item_k)} W_{item_k} \quad (1)$$

Here, the weight of a recommendation i for source j is the sum of the weights of all items within the data source that are linked to the recommendation. In Section ?? we discuss a few different methods for combining the recommendation scores from individual sources.

6.1 Source-specific Models

This section decribes the specific modeling and prediction processes for each data source.

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[5] and and the top result is selected. Next, (as we discussed in Section ??) a query

[2] `http://dbpedia.org/sparql`

[3] `http://facebook.com/press/info.php?statistics`

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

[5] `http://en.wikipedia.org`

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:

$$W_{wiki_i} = \sum_{Linked(profile_j, wiki_i)} W_{profile_j} \tag{2}$$

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

To generate recommendations from Wikipedia, a further query is sent to DBPedia, this time to retrieve new (recommendation) items that are linked to at least two of the relevant Wikipedia items that were found in the previous step. The recommendation items are filtered by type, in the context of music: "Musical Artist" or "Band". For example, as shown in Figure **??**, the article for "Pink Floyd" has a semantic link to the category "English Rock Music Groups", which in turn is linked to "The Beatles". In this manner, "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:

$$W_{friend_i} = \frac{TWCI_{user,friend_i}}{\sqrt{TWI_{user}^2 \cdot TWI_{friend_i}^2}} \tag{3}$$

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.

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`. For each expert found, recommendations are produced using the following equation to compute a score for each candidate item.

$$S_{expert,item} = \frac{|Experts_{item}| - Rank_{expert,item}}{|Experts_{item}|} \tag{4}$$

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:

$$W_{expert_i} = \sum_{Linked(profile_j, expert_i)} (W_{profile_j} \cdot S_{expert,profile_j}) \tag{5}$$

All hash tags that resolve to bands or musical artists that the relevant Twitter experts have knowledge in are potentially recommendable.

6.2 Hybrid Strategies

As pointed out in Section **??**, *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 [**?**] into mixed, weighted, and switching hybrids. We present the three strategies used in *TasteWeights*: Weighted, Mixed and Cross-Source. In order to perform the hybrid step we first need to resolve entities across the different data sources. For example, the system needs to know that the Wikipedia article on the band "Asian Dub Foundation" corresponds to a page in the Facebook graph and to the Twitter hash tag *#adf*. Of the three data sources used in this paper, Wikipedia presents the most evolved semantic graph in terms of completeness and non-redundancy [**?**], and therefore it is the best available resource for entity-resolution. Accordingly, we use Google Search API to map all recommendations to Wikipedia articles to confirm their identities. After this mapping stage we proceed with our different hydrid methods.

Weighted Hybrid. In this approach, a score for each recommended item is simply the weighted sum of the source recommendation scores for each source. Weights for each source 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 [**?**], 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 [**?**]. 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.

Mixed Hybrid. In this approach, recommendations for each source are ranked, and then the top-n are picked from each source, one recommendation at a time by alternating the sources. 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.

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 source/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 **??**) 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:

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

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

7. 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 **??**), recommendations produced by the three hybrid methods (Weighted, Mixed, and Cross-Source; cf. Section **??**), 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 users could reflect on:

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

Sources Interaction. 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.

7.1 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. 32 people participated in the study, which lasted approximately 47 minutes on average.

To assess the effects of explanation and interaction with the system on user experience and understanding of the recommendation process a qualitative analysis was performed 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. Each of the nine methods produced a ranked list of recommendations. To compute the overall accuracy of a given recommendation list we first asked the user to rate the top 15 recommendations in the list and then used Breeze's *R-Score* "utility" metric [**?**] to come up with a utility score for the list. The metric assumes that the value of recommendations decline exponentially based on position in the recommended list. The utility of a given recommendation list for user u is given by:

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

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.

We considered measuring accuracy via more popular approaches including variants of Root Mean Squared Error (RMSE) and Mean Average Error (MAE). However, we opted against using those for two reasons: first, our system's input is music that is not rated by the user but only "liked" so in a way all Facebook likes correspond to 5-star ratings; and second, because we are more concerned with the usefulness of the set of top n recommendations than the complete recommendation list, which usually produced hundreds or thousands of recommendations.

7.1.1 Participants

In total, 32 users participated in the main study, 17 male and 15 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. 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 [6]), 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). When asked about their primary methods for discovering new music, participants top choices was "Friends" (45%), then "Pandora" (36%) and "Radio" (23%).

[6] `http://facebook.com/press/info.php?statistics`

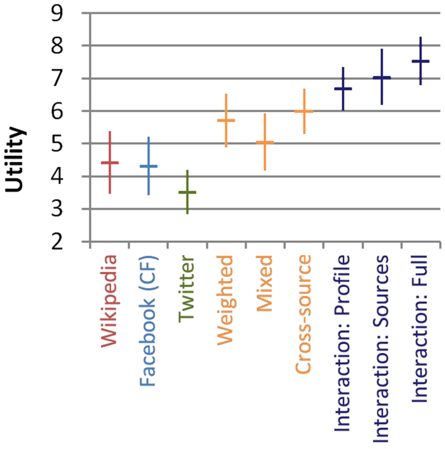


Figure 3: Plot of means of recommendation methods over utility with 95% confidence intervals. The utility metric is described in ??.

Method 1	Method 2	Diff	Lower	Upper	P Val
Cross Hybrid	Wikipedia	1.568	0.119	3.017	0.023
Cross Hybrid	Facebook (CF)	1.678	0.229	3.127	0.011
Cross Hybrid	Twitter	2.477	1.028	3.926	0.000
Full Interaction	Cross Hybrid	1.542	0.935	2.991	0.027

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

7.1.2 *Procedure*

After completing the pre-study questionnaire, participants were given an oral explanation of the system controls and approximately one minute 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 **??**, 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. The purpose of this task 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. In order to rate unknown bands the user was given the chance to look at the band's LastFM[7] page. The page not only contains relevant information about the band but also music samples. After having rated all recommendations users were asked to do a post-questionnaire and provide feedback on their perception of the system.

7.2 Recommendation Accuracy

Figure **??** presents a plot of the means of the nine methods over utility with 95% confidence intervals. Overall, the full interaction method was found to have the highest utility score, while the twitter method 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(44) = 0.005, p = 0.01). We ran one-way repeated-measure ANOVA and made Greenhouse-Geisser correction ($\varepsilon = 0.49$). 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 **??**. Note that not all pair-wise results are shown but only relevant ones.

7.2.1 *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). Based on our small user pool we found no signifcant difference in the three methods. Wikipedia had the highest average utility of 4.42 and "Expert recommendations" sourced from Twitter exhibited the lowest average utility of 3.52. Based on our observations, it appeared that recommendations derived from Twitter were more obscure than the other two sources. However, the authors note that this may a result of the particular recommendation technique used, and not necessarily a reflection on the quality of the underlying data in Twitter.

7.2.2 *Hybrid Results*

Our second analysis focuses on a comparison of the three hybrid approaches to recommendation. The middle portion of Figure **??** shows the utility score for each approach (*weighted*, *mixed*, and *cross-source* hybrid). Only the cross-source hybrid approach, in which we favor recommendations coming from more than one source, performed better than all three single-source methods. The Tukey pair-wise test showed significant differences between the cross-source hybrid method and Wikipedia, Facebook (CF), and Twitter, with p=0.023, p=0.011, and p=0.000 respectively. This is a strong indication that hybridization across social web APIs can help increase predictive accuracy in recommender systems.

7.2.3 *Interaction Results*

The three methods of interaction described in Section **??** were tested in the study and the results are shown in Figure **??**. The full interaction method is the standard use case for the *TasteWeights* system and it exhibited improved performance over the best hybrid approach (p=0.027) indicating that interaction with the full system helped the user get better recommendations. As expected, out of all interactive methods the full interaction one achieved the highest accuracy score of 7.54. 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.

7.3 Explanation & Transparency

[7] http://www.last.fm

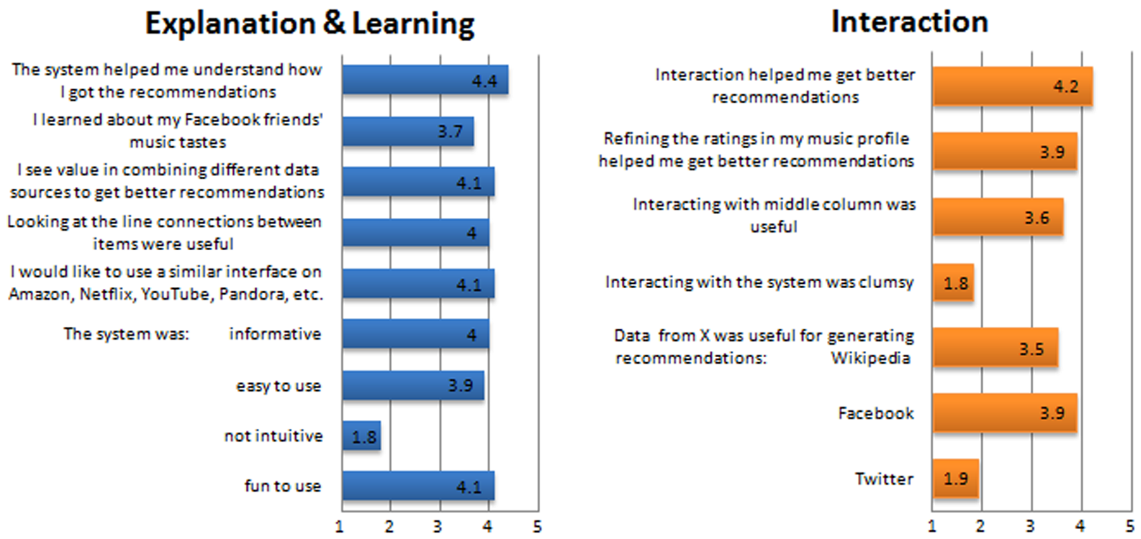


Figure 4: Post-study questionnaire results

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 **??** shows that users generally viewed the system as informative (4 out of 5). The highest agreement value (4.4) 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). Note that source provenance of each recommendation is preserved through interactive edges in the system. 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). Importantly, most participants also reported that the system was fun to use (4.1).

The graph on the right side of Figure **??** shows results for the perceived usefulness of interaction with the system and the quality of each prediction strategy. Users felt that interaction helped them get better recommendations (4.2). Facebook was reported as the most useful source for generating recommendations (3.9), followed by Wikipedia (3.5), with Twitter reported as by far the least useful (1.9). Interestingly, perceived usefulness shows a relative improvement of 9.7% for Facebook over Wikipedia, while accuracy from Figure **??** indicates the Wikipedia slightly outperforming Facebook. 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.

8. 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. 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.

9. CONCLUSION

This paper presented *TasteWeights*, an interactive hybrid recommendation system. The system employs new models for sourcing recommendations from a range of social web APIs and presents hybridization strategies for combining those recommendations. The *TasteWeights* explanatory interface educates users about hybrid recommendation systems and enables them to tweak the underlying algorithms in real-time. A supervised user study was performed using the system to explore research questions relating to visual interactive recommendation systems. The study results indicate that:

- Explaining a hybrid recommendation process through a user interface can increase user satisfaction.
- Interaction at recommendation time can improve recommendation accuracy and user experience.
- Hybrid strategies combining different social APIs can provide better recommendations than traditional CF (over Facebook music preferences).

10. ACKNOWLEDGEMENTS

The authors would like to thank Cha Lee and Sibel Adali for their input on analysis methods and credibility indicators for microblogs, respectively. This work was partially supported by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053; by NSF grant IIS-1058132; by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory contract number FA8650-10-C-7060; and by the U.S. Army Research Laboratory under MURI grant No. W911NF-09-1-0553; The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ARL, AFRL, IARPA, NSF, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.