New Objective Functions for Social Collaborative Filtering

Joseph Noel[†∗], Scott Sanner[†∗], Khoi-Nguyen Tran[∗], Peter Christen[∗], Lexing Xie[∗]
Edwin V. Bonilla[†∗], Ehsan Abbasnejad[†∗], Nicolás Della Penna[†∗]

[†]NICTA, [∗]Australian National University

jinonoel@gmail.com, scott.sanner@nicta.com.au,
{khoi-nguyen.tran, peter.christen, lexing.xie}@anu.edu.au,
{edwin.bonilla, ehsan.abbasnejad}@nicta.com.au, nikete@gmail.com

ABSTRACT

This paper examines the problem of social *collaborative filtering* (CF) to recommend items of interest to users in a social network setting. Unlike standard CF algorithms using relatively simple user and item features, recommendation in social networks poses the more complex problem of learning user preferences from a rich and complex set of user profile and interaction information. Many existing *social CF* methods have extended traditional CF *matrix factorization*, but have overlooked important aspects germane to the social setting. We propose a unified framework for social CF matrix factorization by introducing novel objective functions for training. Our new objective functions have three key features that address main drawbacks of existing approaches: (a) we fully exploit feature-based user similarity, (b) we permit direct learning of user-to-user information diffusion, and (c) we leverage co-preference (dis)agreement between two users to learn restricted areas of common interest. We evaluate these new social CF objectives, comparing them to each other and to a variety of (social) CF baselines, and analyze user behavior on live user trials in a custom-developed Facebook App involving data collected over five months from over 100 App users and their 37,000+ friends.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

social networks, collaborative filtering, machine learning

1. INTRODUCTION

Given the vast amount of content available on the Internet, finding information of personal interest (news, blogs, videos, movies, books, etc.) is often like finding a needle in a haystack. Recommender systems based on *collaborative filtering* (CF) [15] aim to address this problem by leveraging the preferences of similar users in a user population.

As the web has become more social with the emergence of Facebook, Twitter, LinkedIn, and most recently Google+, this adds myriad new dimensions to the recommendation problem by making available a rich labeled graph structure of social content from which user preferences can be learned and new recommendations can be made. In this socially connected setting, no longer are web users simply described by an IP address (with perhaps associated geographical information and browsing history), but rather they are described by a rich user profile (age, gender, location, educational and work history, preferences, etc.) and a rich history of user interactions with their friends (comments/posts, clicks of the like button, tagging in photos, mutual group memberships, etc.). This rich information poses both an amazing opportunity and a daunting challenge for machine learning methods applied to social recommendation — how do we fully exploit rich social network content in recommendation algorithms?

Many existing *social CF* (SCF) approaches [10, 11, 19, 6, 12, 8] extend *matrix factorization* (MF) techniques such as [16] used in the non-social CF setting. These MF approaches have proved quite powerful and indeed, we will show empirically in Section 5 that existing social extensions of MF outperform a variety of other non-MF SCF baselines. The power of CF MF methods stems from their ability to project users and items into latent vector spaces of reduced dimensionality where each is effectively grouped by similarity; in turn, the power of many of the SCF MF extensions stems from their ability to use social network evidence to further constrain (or regularize) latent user projections.

Despite providing state-of-the-art performance on SCF problems, we notice that existing SCF MF objective functions can be improved in three key aspects, which form the basis for our key algorithmic contributions in this paper:

(a) **Learning user similarity:** In existing SCF MF objectives, the mapping from user features to user similarity is fixed. It will be desirable to learn such similarity among a large number of profile attributes from data, such as two users are more similar when they have the same gender or age. To address this, we extend existing *social regularization* and *social spectral regularization* objectives to incorporate *user features* when learning user-user similarities in a *latent* space.

(b) **Direct learning of user-to-user information diffusion:** Existing SCF MF objectives do not permit directly modeling user-to-user information diffusion according to the social graph structure. For example, if a certain user *always* likes content liked by a friend, this cannot be directly *learned* by optimizing existing

SCF MF objectives. To address this, we define a new hybrid SCF method where we *combine* the *collaborative filtering (CF) matrix factorization (MF) objective* used by Matchbox [17] with a *linear content-based filtering (CBF) objective* used to model direct user-user information diffusion in the social network.

(c) **Learning restricted interests:** Existing SCF MF objectives treat users as globally (dis)similar although they may only be (dis)similar in specific areas of latent interest. For example, a friend and their co-worker may both like technology-oriented news content, but have differing interests when it comes to politically-oriented news content. To address this, we define a new social co-preference regularization method that *learns from pairs of user preferences* over the same item to learn *user similarities in specific areas* — a contrast to previous methods that typically enforce global user similarity when regularizing.

The key application contribution of our paper is to evaluate the proposed recommendation algorithms in online human trials of a custom-developed Facebook App for link recommendation. We use data collected over five months from over 100 App users and their 37,000+ friends. Results show that feature-based social spectral regularization outperforms (i) a range of existing CF and SCF baselines, (ii) performs as well on friend recommendations as direct modeling of information diffusion features, and (iii) provides better social regularization than the co-preference approach.

In addition, deploying our algorithm on a real social network provided us with a number of interesting observations from user behavior and feedback discussed in Section 5.3. For example, click feedback correlates weakly with like ratings. Also, the most popular links may be liked by the most people, but they are not liked by everyone on average.

In the rest of this paper, Section 2 provides a succint overview of CF and SCF algorithms, Section 3 proposes three novel objective functions to address (a)–(c), Section 4 dicusses the details of our Facebook application for link recommendation, Section 5 presents two rounds of evaluation with further analysis of user data in our social recommendation setting, and Section 6 concludes this study.

2. DEFINITIONS AND BACKGROUND

Collaborative filtering (CF) [15] is the task of predicting whether, or how much, a user will like (or dislike) an item by leveraging knowledge of that user's preferences *as well as those of other users*. While collaborative filtering need not take advantage of user or item features (if available), a separate approach of *content-based filtering* (CBF) [7] makes individual recommendations by generalizing from the item features of those items the user has explicitly liked or disliked. What distinguishes CBF from CF is that CBF requires item features to generalize whereas CF requires multiple users to generalize; however, CBF and CF are not mutually exclusive and recommendation systems often combine the two approaches [1]. When a CF method makes use of item and user features as well as multiple users, we refer to it as CF although in some sense it may be viewed as a combined CF and CBF approach.

We define *social CF* (SCF) as the task of CF augmented with additional social network information such as the following:

- Expressive personal profile content: gender, age, places lived, schools attended; favorite books, movies, quotes; online photo albums (and associated comment text).

- Explicit friendship or trust relationships.

- Content that users have personally posted (often text, images, and links).

- Content of public (and if available, private) interactions between users (often text, images and links).

- Evidence of external interactions between users such as being jointly tagged in photos or videos.

- Expressed preferences (likes/dislikes of posts and links).

- Group memberships (often for hobbies, activities, social or political discussion).

We note that CF is possible in a social setting without taking advantage of the above social information, hence we include CF baselines in our later experiments on SCF.

2.1 Notation

We present all algorithms for CF and SCF using the following mathematical notation:

- N users. For methods that can exploit user features, we define an I-element user feature vector $\mathbf{x} \in \mathbb{R}^I$ (alternately if a second user is needed, $\mathbf{z} \in \mathbb{R}^I$). For methods that do not use user feature vectors, we simply assume $\mathbf{x}$ is an index $\mathbf{x} \in \{1 \ldots N\}$ and that $I = N$.

- M items. For methods that can exploit item features, we define a J-element feature vector $\mathbf{y} \in \mathbb{R}^J$. The feature vectors for users and items can consist of any real-valued features as well as $\{0, 1\}$ features like user and item IDs. For methods that do not use item feature vectors, we simply assume $\mathbf{y}$ is an index $\mathbf{y} \in \{1 \ldots M\}$ and that $J = M$.

- A (non-exhaustive) data set D of single user *preferences* of the form $D = \{(\mathbf{x}, \mathbf{y}) \rightarrow R_{\mathbf{x},\mathbf{y}}\}$ where the binary *response* $R_{\mathbf{x},\mathbf{y}} \in \{0 \text{ (dislike)}, 1 \text{ (like)}\}$.

- A (non-exhaustive) data set C of *co-preferences* (cases where *both* users $\mathbf{x}$ and $\mathbf{z}$ expressed a preference for $\mathbf{y}$ – not necessarily in agreement) derived from D of the form $C = \{(\mathbf{x}, \mathbf{z}, \mathbf{y}) \rightarrow P_{\mathbf{x},\mathbf{z},\mathbf{y}}\}$ where co-preference class $P_{\mathbf{x},\mathbf{z},\mathbf{y}} \in \{-1 \text{ (disagree)}, 1 \text{ (agree)}\}$. Intuitively, if *both* user $\mathbf{x}$ and $\mathbf{z}$ liked or disliked item $\mathbf{y}$ then we say they *agree*, otherwise if one liked the item and the other disliked it, we say they *disagree*.

- A similarity rating $S_{\mathbf{x},\mathbf{z}}$ between any users $\mathbf{x}$ and $\mathbf{z}$. This is used to summarize all social interaction between user $\mathbf{x}$ and user $\mathbf{z}$ in the term $S_{\mathbf{x},\mathbf{z}} \in \mathbb{R}$. A definition of $S_{\mathbf{x},\mathbf{z}} \in \mathbb{R}$ that has been useful is the following average-normalized measure of user interactions:

$$Int_{\mathbf{x},\mathbf{z}} = \frac{\# \text{ interactions b/w } \mathbf{x} \text{ and } \mathbf{z}}{\frac{1}{N(N-1)} \sum_{\mathbf{x}',\mathbf{z}' \neq \mathbf{x}'} \# \text{ interactions b/w } \mathbf{x}' \text{ and } \mathbf{z}'}$$

$$S_{\mathbf{x},\mathbf{z}} = \ln \left(Int_{\mathbf{x},\mathbf{z}} \right) \qquad (1)$$

How "# interactions between $\mathbf{x}$ and $\mathbf{z}$" is explicitly defined is specific to a social network setting and hence

we defer details of the particular method user for evaluations in this paper to Section 4.2.4.

We also define $S_{\mathbf{x},\mathbf{z}}^{+}$, a *non-negative* variant of $S_{\mathbf{x},\mathbf{z}}$:

$$S_{\mathbf{x},\mathbf{z}}^{+} = \ln\left(1 + Int_{\mathbf{x},\mathbf{z}}\right) \tag{2}$$

- A set $friends_{\mathbf{x}}$ such that $\mathbf{z} \in friends_{\mathbf{x}}$ iff $\mathbf{z}$ is officially denoted as a *friend* of $\mathbf{x}$ on the social network.

Having now defined notation, we proceed to survey a number of CBF, CF, and SCF algorithms including all of those compared to or extended in this paper.

2.2 Content-based Filtering (CBF)

Since our objective in this work is to classify whether a user likes an item or not (i.e., a binary objective), we focus on binary classification-based CBF approaches. While a variety of classifiers may work well, we choose the *support vector machine* (SVM) [5] since it is well-known for its state-of-the-art classification performance.

For the experiments in this paper, we use a *linear* SVM (implemented in the *LibSVM* [4] toolkit) with feature vector $\mathbf{f} \in \mathbb{R}^{F}$ derived from $(\mathbf{x}, \mathbf{y}) \in D$, denoted as $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ and tuning parameter C. A linear SVM learns a weight vector $\mathbf{w} \in \mathbb{R}^{F}$ such that $\mathbf{w}^{T}\mathbf{f}_{\mathbf{x},\mathbf{y}} > 0$ indicates a like (1) classification of $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ and $\mathbf{w}^{T}\mathbf{f}_{\mathbf{x},\mathbf{y}} \leq 0$ indicates a dislike (0) classification.

A detailed list of features $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ used in the SVM for the Facebook link recommendation task evaluated in this paper are defined as in Section 4.2.3 — these include *user features* such as age and gender (binary) and *item features* such as popularity (number of times the item was shared). Going one step beyond standard CBF, our SVM features also include *joint* user and item features from the social network, in particular binary *information diffusion* [3] features for *each* friend $\mathbf{z} \in friends_{\mathbf{x}}$ indicating if $\mathbf{z}$ liked (or disliked) $\mathbf{y}$. Crucially we note that our SVM implementation of CBF using social network features actually represents a *social CBF* extension since it can learn when a friend $\mathbf{z}$'s preference for items are predictive of user $\mathbf{x}$'s preferences.

2.3 Collaborative Filtering (CF)

2.3.1 *k-Nearest Neighbor*

One of the most common forms of CF is the nearest neighbor approach [2]. There are two main variants of nearest neighbors for CF, *user-based* and *item-based* — both methods generally assume that no user or item features are provided, so here $\mathbf{x}$ and $\mathbf{y}$ are simply respective user and item indices. When the number of users is far fewer than the number of items, it has been found that the user-based approach usually provides better predictions as well as being more efficient in computation [2]; this holds for the evaluation in this paper, so we focus on the user-based approach.

Given a user $\mathbf{x}$ and an item $\mathbf{y}$, let $\mathcal{N}(\mathbf{x} : \mathbf{y})$ be the set of N *user* nearest neighbors of $\mathbf{x}$ that have also given a rating for $\mathbf{y}$, where "nearest" is determined by the cosine similarity (i.e., normalized dot product) $Sim_{\mathbf{x},\mathbf{z}}$ between two vectors of ratings for users $\mathbf{x}$ and $\mathbf{z}$ (when both have rated the same item). Following [2], the predicted rating $\hat{R}_{\mathbf{x},\mathbf{y}} \in [0, 1]$ that the user $\mathbf{x}$ gives item $\mathbf{y}$ can then be calculated as

$$\hat{R}_{\mathbf{x},\mathbf{y}} = \frac{\sum_{\mathbf{z} \in N(\mathbf{x}:\mathbf{y})} Sim_{\mathbf{x},\mathbf{z}} R_{\mathbf{z},\mathbf{y}}}{\sum_{\mathbf{z} \in N(\mathbf{x}:\mathbf{y})} Sim_{\mathbf{x},\mathbf{z}}} \tag{3}$$

2.3.2 *Matrix Factorization (MF) Models*

Another common approach to CF attempts to factorize an (incomplete) matrix R of dimension $I \times J$ containing observed ratings $R_{\mathbf{x},\mathbf{y}}$ (note that $\mathbf{x}$ and $\mathbf{y}$ are assumed to row and column indices of $R_{\mathbf{x},\mathbf{y}}$) into a product $R \approx U^{T}V$ of real-valued rank-K matrices U and V:

$$U = \begin{bmatrix} U_{1,1} & \ldots & U_{1,I} \\ \vdots & U_{k,i} & \vdots \\ U_{K,1} & \ldots & U_{K,I} \end{bmatrix} \quad V = \begin{bmatrix} V_{1,1} & \ldots & V_{1,J} \\ \vdots & V_{k,j} & \vdots \\ V_{K,1} & \ldots & V_{K,J} \end{bmatrix}$$

In this initial MF setting, we *do not* leverage user and item features; hence, we let $U_{\mathbf{x}}$ and $U_{\mathbf{y}}$ pick out the respective $\mathbf{x}$th and $\mathbf{y}$th *columns* of U and V such that $U_{\mathbf{x}}^{T}V_{\mathbf{y}}$ acts as a measure of affinity between user $\mathbf{x}$ and item $\mathbf{y}$ in their respective K-dimensional latent spaces $U_{\mathbf{x}}$ and $V_{\mathbf{y}}$.

However, there remains the question of how we can learn U and V given that R is incomplete (i.e., it contains missing entries since D is generally non-exhaustive). The answer is simple: we need only define a reconstruction error objective we wish to minimize as a function of U and V and then use gradient descent to optimize it; formally then, we can optimize the following MF objective [16]:

$$\sum_{(\mathbf{x},\mathbf{y}) \in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - U_{\mathbf{x}}^{T}V_{\mathbf{y}})^{2} \tag{4}$$

While this objective is technically bilinear, we can easily apply an *alternating gradient descent* approach to approximately optimize it and determine good projections U and V that (locally) minimize the reconstruction error of the observed responses $R_{\mathbf{x},\mathbf{y}}$ (see e.g. [16]).

2.3.3 *Social Collaborative Filtering (SCF)*

In this work on *social CF* (SCF), we focus on extending MF-based SCF approaches as they allow us to incorporate flexible objective functions that take into account a vast array of social network information. Additionally, we can learn the parameters of the proposed algorithms from data by using gradient-based optimization.

To date, there are essentially two general classes of MF methods applied to SCF of which the authors are aware. The first class of social MF methods can be termed as *social regularization* approaches in that they constrain the latent projection of users according to social network information. There are two closely related social regularization methods that directly constrain $U_{\mathbf{x}}$ and $U_{\mathbf{z}}$ for user $\mathbf{x}$ and $\mathbf{z}$ based on evidence $S_{\mathbf{x},\mathbf{z}}$ of interaction between $\mathbf{x}$ and $\mathbf{z}$. The first class of methods are simply termed *social regularization* [19, 6] where $\langle \cdot, \cdot \rangle$ denotes an inner product:

$$\sum_{\mathbf{x}} \sum_{\mathbf{z} \in friends_{\mathbf{x}}} \frac{1}{2}(S_{\mathbf{x},\mathbf{z}} - \langle U_{\mathbf{x}}, U_{\mathbf{z}} \rangle)^{2} \tag{5}$$

The second class of methods are termed *social spectral regularization* [12, 8]:

$$\sum_{\mathbf{x}} \sum_{\mathbf{z} \in friends_{\mathbf{x}}} \frac{1}{2}S_{\mathbf{x},\mathbf{z}}^{+}\|U_{\mathbf{x}} - U_{\mathbf{z}}\|_{2}^{2}. \tag{6}$$

We refer to the latter as *spectral* regularization methods since they are identical to the objectives used in spectral clustering [13]. The idea behind both variants of social regularization should be apparent: the larger $S_{\mathbf{x},\mathbf{z}}$ or $S_{\mathbf{x},\mathbf{z}}^{+}$, the

more $U_\mathbf{x}$ and $U_\mathbf{z}$ need to be similar (according to slightly different metrics) in order to minimize the given objective.

The *SoRec* system [11] proposes a slight twist on social spectral regularization in that it learns a third $N \times N$ (n.b., $I = N$) *interactions matrix* Z, and uses $U_\mathbf{z}^T Z_\mathbf{z}$ to predict user-user interaction preferences in the same way that standard CF uses V in $U_\mathbf{x}^T V_\mathbf{y}$ to predict user-item ratings. *SoRec* also uses a sigmoidal transform $\sigma(o) = \frac{1}{1+e^{-o}}$ since $\bar{S}_{\mathbf{x},\mathbf{z}}$ is $S_{\mathbf{x},\mathbf{z}}$ restricted to the range $[0,1]$ (e.g., $\bar{S}_{\mathbf{x},\mathbf{z}} = \sigma(S_{\mathbf{x},\mathbf{z}})$):

$$\sum_\mathbf{z} \sum_{\mathbf{z} \in friends_\mathbf{x}} \frac{1}{2}(\bar{S}_{\mathbf{x},\mathbf{z}} - \sigma(\langle U_\mathbf{x}, Z_\mathbf{z} \rangle))^2 \qquad (7)$$

The second class of SCF MF approaches represented by the single exemplar of the *Social Trust Ensemble* (STE) [10] can be termed as a *weighted friend average* approach since this approach simply composes a prediction for item $\mathbf{y}$ from an α-weighted average ($\alpha \in [0,1]$) of a user $\mathbf{x}$'s predictions *as well as* their friends ($\mathbf{z}$) predictions (as evidenced by the additional $\sum_\mathbf{z}$ in the objective below):

$$\sum_{(\mathbf{x},\mathbf{y}) \in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - \sigma(\alpha U_\mathbf{x}^T V_\mathbf{y} + (1-\alpha) \sum_{\mathbf{z} \in friends_\mathbf{x}} U_\mathbf{x}^T V_\mathbf{z}))^2 \qquad (8)$$

As for the MF CF methods, all MF SCF methods can be optimized by alternating gradient descent on the respective matrix parameterizations; we refer the interested reader to each paper for further details.

3. NEW OBJECTIVE FUNCTIONS FOR SCF

Having surveyed previous CF work, especially MF-based CF and SCF methods, we now present the major conceptual contributions of the paper. We begin by introducing a unified matrix factorization framework for optimizing all MF objectives evaluated in this paper — old and new.

3.1 A Composable Objective Framework

We take a composable approach to MF-based (S)CF, where an optimization objective *Obj* is composed of sums of one or more objective components:

$$Obj = \sum_i \lambda_i Obj_i \qquad (9)$$

Because each objective may be weighted differently, a weighting term $\lambda_i \in \mathbb{R}$ is included for each component. In the current work, we manually tuned each λ_i, except for the *first i* in $\sum_i$, which can be set as $\lambda_i = 1$ without loss of generality.

Most target predictions in this paper are binary ($\{0,1\}$), therefore a sigmoidal transform $\sigma(o) = \frac{1}{1+e^{-o}}$ of a prediction $o \in \mathbb{R}$ may be used to squash it to the range $[0,1]$. Where the σ transform may be optionally included, this is written as $[\sigma]$. While σ transforms are generally advocated for real-valued regressor outputs when used in a classification setting, we note that our experiments showed little variation in results whether including or omitting it, although including it tended to slow the convergence of gradient optimization. Nonetheless, where appropriate, we include the possibility of a σ transform since it may prove useful in other settings.

3.2 Existing Objective Functions

For completeness, we first cover a number of known objective components that are used in the objectives evaluated and extended in this paper. A discussion of gradient

optimization along with all necessary derivatives for these objectives is provided in Appendix A.

3.2.1 Matchbox-style Matrix Factorization (Obj_{pmcf})

In Section 2.3.2, we discussed an MF objective (4) that *did not* make use of user or item features. However, if we *do* have user feature vector $\mathbf{x}$ and item feature vector $\mathbf{y}$, we can respectively represent the latent projections of user and item as $(U\mathbf{x})_{1\ldots K}$ and $(V\mathbf{y})_{1\ldots K}$ and hence use $\langle U\mathbf{x}, V\mathbf{y} \rangle = \mathbf{x}^T U^T V \mathbf{y}$ as a measure of affinity between user $\mathbf{x}$ and item $\mathbf{y}$. Substituting the feature-based $\mathbf{x}^T U^T V \mathbf{y}$ for the featureless $U_\mathbf{x}^T V_\mathbf{y}$ in (4), we arrive at the form of the basic CF objective function used in *Matchbox* [17] — although Matchbox used Bayesian optimization methods, we can easily express its main objective in the following log likelihood form:

$$Obj_{pmcf} = \sum_{(\mathbf{x},\mathbf{y}) \in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{x}^T U^T V \mathbf{y})^2 \qquad (10)$$

3.2.2 Regularization of U, V & $\mathbf{w}$ (Obj_{ru}, Obj_{rv}, $Obj_\mathbf{w}$)

To help in generalization, it is important to regularize any free matrix parameters U and V (e.g., from Section 3.2.1) or vector parameters $\mathbf{w}$ (e.g., from Section 2.2) to prevent overfitting when dealing with sparse data. This can be done with a simple L_2 regularizer that models a spherical Gaussian prior on the parameters. This regularization component can be specified for U, V, and $\mathbf{w}$ as follows:

$$Obj_{ru} = \frac{1}{2}\|U\|_{\text{Fro}}^2 = \frac{1}{2}\text{tr}(U^T U) \qquad Obj_{rv} = \frac{1}{2}\text{tr}(V^T V)$$

$$Obj_{rw} = \frac{1}{2}\|\mathbf{w}\|_2^2 = \frac{1}{2}\mathbf{w}^T \mathbf{w} \qquad (11)$$

3.3 New Objective Functions

Now we return to our observations concerning the deficiencies of existing SCF MF methods as outlined in Section 1 and propose new objective functions to address these issues. Gradient-based optimization for these new objectives and all necessary derivatives are covered in Appendix A.

3.3.1 Feature Social Regularization (Obj_{rs} & Obj_{rss})

Our previous discussion of SCF methods in Section 2.3.3 covered three different methods for *social regularization* — that is, constraining users to be similar based on evidence from the social network. However, none of these previous three SCF social regularization methods exploited user features in the *learning* process; more precisely $U_\mathbf{x}$ and $U_\mathbf{z}$ were regularized, but not the feature-based latent spaces $U\mathbf{x}$ and $U\mathbf{z}$. Hence if a gender difference in $\mathbf{x}$ and $\mathbf{z}$ was the crucial factor to differentiating the latent spaces of each user, this could be *learned* if we had a way of socially regularizing $U\mathbf{x}$ and $U\mathbf{z}$ directly. To address this, we provide two variants of *feature-based social regularization*.

The first new objective is an extension of simple *social regularization* [19, 6] that incorporates user features:

$$Obj_{rs} = \sum_\mathbf{x} \sum_{\mathbf{z} \in friends(\mathbf{x})} \frac{1}{2}(S_{\mathbf{x},\mathbf{z}} - \langle U\mathbf{x}, U\mathbf{z} \rangle)^2 \qquad (12)$$

$$= \sum_\mathbf{x} \sum_{\mathbf{z} \in friends(\mathbf{x})} \frac{1}{2}(S_{\mathbf{x},\mathbf{z}} - \mathbf{x}^T U^T U\mathbf{z})^2$$

Alternately, we could extend *social spectral regularization* [12, 8] by incorporating user features into its objective:

$$Obj_{rss} = \sum_{\mathbf{x}} \sum_{\mathbf{z} \in friends(\mathbf{x})} \frac{1}{2} S_{\mathbf{x},\mathbf{z}}^{+} \|U\mathbf{x} - U\mathbf{z}\|_2^2 \qquad (13)$$

$$= \sum_{\mathbf{x}} \sum_{\mathbf{z} \in friends(\mathbf{x})} \frac{1}{2} S_{\mathbf{x},\mathbf{z}}^{+} (\mathbf{x} - \mathbf{z})^T U^T U (\mathbf{x} - \mathbf{z})$$

While we note these extensions are straightforward, they have the simple property that they allow the system to learn the latent user projection matrix U *as a function of user features* in order to minimize the social regularization penalty. Just as the Matchbox objective in Section 3.2.1 allows us to exploit user and item features in MF-based CF, these new social regularization objectives permit more flexibility in exploiting user features in *learning* user similarity.

3.3.2 Hybrid Information Diffusion + SCF (Obj_{phy})

One major weakness of MF methods is that they cannot model direct joint features over user and items — they must model user and item features independently in order to compute the independent latent projections $U\mathbf{x}$ and $U\mathbf{z}$. Unfortunately, this prevents standard MF objectives from modeling direct user-to-user information diffusion [3] — the unidirectional flow of information (e.g., links) from one user to another. This is problematic because if user $\mathbf{x}$ *always* likes what $\mathbf{z}$ has posted or liked, then we would like to shortcut the latent representation and simply learn to recommend user $\mathbf{z}$'s liked or posted items to user $\mathbf{x}$.

We fix this deficiency of MF by introducing another objective component in addition to the standard MF objective, which serves as a simple linear regressor for such information diffusion observations. The resulting hybrid objective component then becomes a combination of latent MF and linear regression objectives.

For the linear regressor $\mathbf{w}^T \mathbf{f}_{\mathbf{x},\mathbf{y}}$, we make use of the *same* weight vector $\mathbf{w}$ and feature vector $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ mentioned in Section 2.2; $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ is fully defined for our empirical evaluation in Section 4.2.3. As previously noted, $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ includes *joint* user and item features from the social network, in particular binary *information diffusion* [3] features for *each* friend $\mathbf{z} \in friends_{\mathbf{x}}$ indicating if $\mathbf{z}$ liked (or disliked) $\mathbf{y}$. As a consequence, learning $\mathbf{w}$ allows the linear regressor to predict in a personalized way for any user $\mathbf{x}$ whether they are likely to follow their friend $\mathbf{z}$'s preference for $\mathbf{y}$.

Formally, to define our hybrid information diffusion plus SCF objective, we additively combine the output of the linear regression prediction with the Matchbox prediction:

$$Obj_{phy} = \sum_{(\mathbf{x},\mathbf{y}) \in D} \frac{1}{2} (R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{w}^T \mathbf{f}_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{x}^T U^T V y)^2 \quad (14)$$

3.3.3 Co-preference Regularization (Obj_{cp})

A crucial aspect missing from other SCF methods is that while two users may not be globally similar or opposite in their preferences, there may be sub-areas of their interests which can be correlated to each other. For example, two friends may have similar interests concerning technology news, but different interests concerning political news. *Co-preference regularization* aims to learn such selective co-preferences. The motivation is to constrain users $\mathbf{x}$ and $\mathbf{z}$ who have similar or opposing preferences to be similar or opposite in the *same* latent item space relevant to item $\mathbf{y}$.

We use $\langle \cdot, \cdot \rangle_{\bullet}$ to denote a re-weighted inner product. The purpose of this inner product is to "mask" enforcement of la-
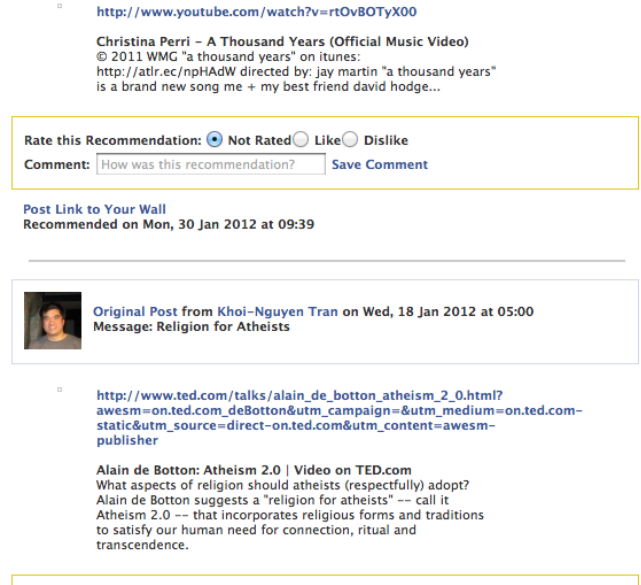
Figure 1: The Facebook LinkR App showing two link recommendations to a user. The first link recommendation is from a non-friend and hence only shows the link description. The second link recommendation is from a friend and includes the friend's commentary on the link as well as the link description. Users have the option of liking or disliking each recommendation as well as providing feedback.

tent space similarities or dissimilarities between users to be restricted to the same latent spaces as the co-preferred items. To this end, the objective component for *co-preference regularization* along with its expanded form is

$$Obj_{cp} = \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in C} \frac{1}{2} (P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \langle U\mathbf{x}, U\mathbf{z} \rangle_{V\mathbf{y}})^2 \qquad (15)$$

$$= \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in C} \frac{1}{2} (P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \mathbf{x}^T U^T \operatorname{diag}(V\mathbf{y}) U\mathbf{z})^2$$

We might also define a *social co-preference spectral regularization* approach, but our experiments so far have not indicated this works as well as the above objective.

In contrast to social regularization defined previously, co-preference regularization does not require knowledge of friendships or user interactions to determine co-preferences and hence can enforce regularization constraints between *all* users.

4. EVALUATION FRAMEWORK

Next we discuss our Facebook Link Recommendation App (LinkR), data collected, and our evaluation methodology.

4.1 Link Recommendation App on Facebook

To evaluate existing and newly proposed (S)CF methods discussed in this paper, we created a Facebook application (i.e., a Facebook "App") that recommends links to users every day, where the users may give their feedback on the links indicating whether they *liked* it or *disliked* it. Figure 1 shows our Facebook LinkR App as it appears to users.

The functionalities of the LinkR application on a daily basis are as follows:

1. Collect new data shared by users and their friends.

2. Initiate retraining of all active (S)CF link recommendation algorithms on the latest collected data. C and D from Section 2.1 are populated from all explicit likes and dislikes observed via the Facebook LinkR App and all "likes" observed via the Facebook interface.

3. Post-retrain, recommend three links to the users according to their assigned recommendation algorithm.

4. Collect feedback from the users on whether they liked or disliked the recommendations as well as any additional commentary the user wishes to provide.

Details of (S)CF link recommendation algorithms and user assignments will be discussed shortly; first we cover data collected by the LinkR App and used by the recommenders.

4.2 Facebook Data Collected

At its peak membership, 111 users had elected to install the Facebook App developed for this project. From this user base, we were able to gather data on over 37,626 users and 605,847 links in total by the end of the evaluation period.

4.2.1 User Data

Data that are collected and used to define the user feature vector $\mathbf{x}$ introduced in Section 2.1 for the LinkR Facebook App are defined as follows:

- $[\mathbf{x}_{id} = id] \in \{0, 1\}, \forall id$: every unique Facebook ID (user) recorded in the App was assigned its own binary indicator in $\mathbf{x}$; all id indicators are mutually exclusive.

- $gender \in \{0(\text{female}), 1(\text{male})\}$.

- $age \in \mathbb{N}$.

We note that the indicator of friendships for $\mathbf{x}$ is stored in the $friends_{\mathbf{x}}$ set defined in Section 2.1 and used in various previous objective definitions, but not explicitly stored in $\mathbf{x}$.

4.2.2 Link Data

Data that are collected and used to define the item feature vector $\mathbf{y}$ introduced in Section 2.1 for the LinkR Facebook App are defined as follows:

- $[\mathbf{y}_{poster} = id] \in \{0, 1\}, \forall id$: binary indicator feature for the id of the user who posted the link; all such binary indicator features are mutually exclusive.

- $[\mathbf{y}_{wall} = id] \in \{0, 1\}, \forall id$: binary indicator feature for the id of the user on whose wall the link was posted; all such binary indicator features are mutually exclusive.

- Count of total link "likes" on Facebook.

- Count of total link shares on Facebook.

- Count of total link comments posted on Facebook.

4.2.3 Joint User and Link Data

The feature vector $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ used in Sections 2.2 and 3.3.2 for the LinkR Facebook App is defined as the *concatenation* of $\mathbf{x}$, $\mathbf{y}$ (above) and the following additional social network information diffusion features:

- $\mathbf{z}$ liked $\mathbf{x} \in \{0, 1\}, \forall \mathbf{z} \in friends_{\mathbf{x}}$: for every friend $\mathbf{z}$ of user $\mathbf{x}$, we have a binary information diffusion feature indicating whether user $\mathbf{z}$ liked item $\mathbf{y}$ (recall that $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ is built w.r.t. a specific user $\mathbf{x}$ and item $\mathbf{y}$).

4.2.4 Interaction Data

We define *# interactions between user* $\mathbf{x}$ *and user* $\mathbf{z}$ in Section 2.1 as follows (all interactions are equally weighted):

1. Being friends.

2. Posting, liking, or commenting on an item (link, photo, video, or message) on a user's wall.

3. Being tagged together in the same photo or video.

4. Attending the same school or class, playing sports together, working together for a company or on a project.

4.3 Live Online Recommendation Trials

LinkR users were randomly and blindly assigned one of four algorithms in each of two live trials (algorithm details in Section 5). The rationale for assigning a single recommendation algorithm to a user was to obtain survey feedback from a user on their assigned algorithm to understand qualitative recommendation issues from a holistic algorithm perspective not necessarily obvious from quantitative measures alone.

Figure 1 shows the LinkR App interface, which displays both friend and non-friend link recommendations and allows the user to rate each link as like or dislike and provide optional feedback. LinkR recommended three links per day to avoid position bias and information overload. In early testing, users commented that many links older than two weeks were outdated or broken so LinkR only recommends links posted in the past two weeks that the user has not already posted, liked, or disliked. Based on first trial feedback, in the second trial we avoided recommendations of (i) non-English links and (ii) links lacking a text description.

5. EMPIRICAL RESULTS

Here we analyze results collected during two LinkR trials.[1]

5.1 First Trial

In our first trial, we evaluated four (S)CF algorithms:

1. **k-Nearest Neighbor (KNN)**: Section 2.3.1 (N=50)

2. **Support Vector Mach. (SVM)**: Section 2.2 (C=2)

3. **Matchbox (Mbox)**: Matchbox CF (λ=10^2, K=5)

$$Obj_{pmcf} + \lambda Obj_{ru} + \lambda Obj_{rv}$$

4. **Social Matchbox (Soc. Mbox)**: user feature socially regularized Matchbox SCF (λ_{rs}=10^{-3}, λ=10^2, K=5)

$$Obj_{pmcf} + \lambda_{rs} Obj_{rs} + \lambda Obj_{ru} + \lambda Obj_{rv}$$

Objectives for (Soc.) Mbox were given in Section 3 and optimized via gradient descent as in Appendix A. λ's for (Soc.) Mbox were tuned prior to the start of the trial by a systematic line (grid) search over 10^n for $n \in \{-5, -4, \dots, 5\}$ to maximize accuracy on 25% held-out data, training on the other 75%. This was repeated for $K \in \{3, 5, 7, 10, 15, 20, 30\}$ to find the best K. N and C were tuned similarly via line search over $N \in \{1, 2, \dots, 250\}$ and $C \in [10^{-4}, 10^4]$.

Trial 1 – Aug. 25, 2011 to Oct. 13, 2011

	SMB	MB	SVM	KNN	Total
Users All	26	26	28	28	108
Users ≥ 10	13	9	13	5	40
Users ≥ 30	9	3	11	3	26
Ratings All	819	526	901	242	2508
Ratings ≥ 10	811	505	896	228	2440
Ratings ≥ 30	737	389	851	182	2159
Clicks All	383	245	413	218	1259

Trial 2 – Oct. 14, 2011 to Feb. 10, 2012

	SMB	Sp.MB	Sp.CP	SHyb.	Total
Users All	27	27	29	28	111
Users ≥ 10	15	11	8	12	46
Users ≥ 30	12	9	5	10	36
Ratings All	1434	882	879	614	3809
Ratings ≥ 10	1411	878	863	602	3754
Ratings ≥ 30	1348	850	802	570	3570
Clicks All	553	320	278	199	1350

Table 1: Number of users assigned per algorithm in the first and second trials. ≥ 10 (≥ 30) indicates data for the subset of users with at least 10 (30) ratings. Data from non-rating users (and their friends) was important for the performance of all algorithms.

First trial details are provided in Table 1 (top); algorithm performance is shown in Figure 2 (top). 95% binomial proportion confidence intervals (using the asymmetrical Wilson score interval method [18]) are shown for the combined data for all users of each algorithm. While user usage varies, this method of combining all user data for CF system performance analysis is a standard evaluation approach for CF systems; notably, RMSE over *all combined ordinal user ratings* was used for determining the winner of the Netflix prize[2].

Except for Mbox, most algorithms performed comparably on non-friend recommendations. For friend recommendations, Soc. Mbox performed best, where it appears that social regularization helped it effectively find latent representations of friends with similar interests; it performed better than Mbox without social regularization and SVM which attempted to explicitly model information diffusion from friends. While KNN had low usage, it would be statistically unlikely to match Soc. Mbox's performance on friend recommendations — to do so if the amount of KNN data were doubled, nearly all new ratings would have to be "like".

5.2 Second Trial

For the second trial, Soc. Mbox was included as a baseline since it was the top performer from the first trial. The remaining three algorithms were all relatively orthogonal Soc. Mbox extensions or variants based on the *three novel objective functions* defined in Section 3.3 (all used $K=5$):

1. **Social Matchbox (Soc. Mbox)**: unchanged

2. **Spectral Matchbox (Sp. Mbox)**: $(\lambda_{rss}=10^{-3}, \lambda=10)$
$$Obj_{pmcf} + \lambda_{rss}Obj_{rss} + \lambda Obj_{ru} + \lambda Obj_{rv}$$

3. **Social Hybrid (Soc. Hybrid)**: $(\lambda_{rs}=10^{-3}; \lambda=10^4)$
$$Obj_{phy} + \lambda_{rs}Obj_{rs} + \lambda Obj_{ru} + \lambda Obj_{rv} + \lambda Obj_{rw}$$

4. **Spectral Copreference (Sp. CP)**: $(\lambda_{cp}=10^{-4}; \lambda=10)$
$$Obj_{pmcf} + \lambda_{cp}Obj_{cp} + \lambda Obj_{ru} + \lambda Obj_{rv}$$

---

[2] http://www.netflixprize.com/

All objectives are defined in Section 3 and optimized via gradient descent as in Appendix A. λ and K parameters for Soc. Mbox were left unchanged from the first trial in order to use it as a fixed comparative baseline across both trials. We chose $K=5$ for all other recommendation algorithms to provide a controlled comparison with Soc. Mbox. With $K=5$ fixed, all other λ parameters were tuned prior to the start of the second trial using the same systematic grid search methodology as described for the first trial.

Second trial details are provided in Table 1 (bottom); on the start of the second trial, users were notified that they would be randomly assigned to new algorithms and encouraged to re-engage with the LinkR App if they had not been using it. Two email reminders were sent during the trial.

Second trial results using the same evaluation methodology as described for the first trial are shown in Figure 2 (bottom). Following are key observations from this trial:

- Soc. Mbox did not perform as well in the second trial as it had in the first trial. We hypothesize that Soc. Mbox may have performed better if λ_{rs} and λ were better tuned for the amount of data in the second trial. To evaluate this hypothesis, in the following table, we show the accuracy of Soc. Mbox at predicting link likes/dislikes on second trial data, training on 75% of the data and testing on the remaining 25%:

	$\lambda_{rs}=10^{-1}$	$=10^{-2}$	$=10^{-3}$	$=10^{-4}$	$=10^{-5}$
$\lambda=10^1$	0.325	0.307	0.301	0.437	0.540
$\lambda=10^2$	0.306	0.301	**0.300**	0.295	0.300
$\lambda=10^3$	0.297	0.301	0.307	0.300	0.301

Here we show results for the second trial parameter settings in bold that achieve prediction accuracy of 0.300; however if both λ and λ_{rs} are reduced, we note a substantial improvement to 0.436 and 0.540. It appears less regularization of U and V is needed in the presence of the additional data in the second trial and the accuracy differential here suggests the need to periodically re-tune parameters to maintain optimal performance.

- Spec. Mbox performed exceedingly well in the second trial and this suggests that our novel feature-based *spectral* social regularization is likely a better method of regularization for U than the social regularization of Soc. Mbox. Even when considering the best achievable performance of Soc. Mbox (54%), this would still fall well below Spec. Mbox's impressive 65%.

- Soc. Hybrid statistically ties Spec. Mbox at recommending friend links (where it can learn user-to-user information diffusion), but performs less well on non-friend links (where there is no such diffusion). These results suggest that the space- and compution-efficient low-dimensional learning of Spec. Mbox can recommend friend links just as well as Soc. Hybrid's modeling of explicit user-to-user information diffusion.

- Given that each LinkR user shared co-preferences with 535.1 other users on average (indicating that this data is far from sparse), it would appear from the performance of Spec. CP that co-preferences serve as a somewhat noisy social regularization constraint compared to social regularization based on interactions between friends as exemplified by Spec. Mbox's performance.
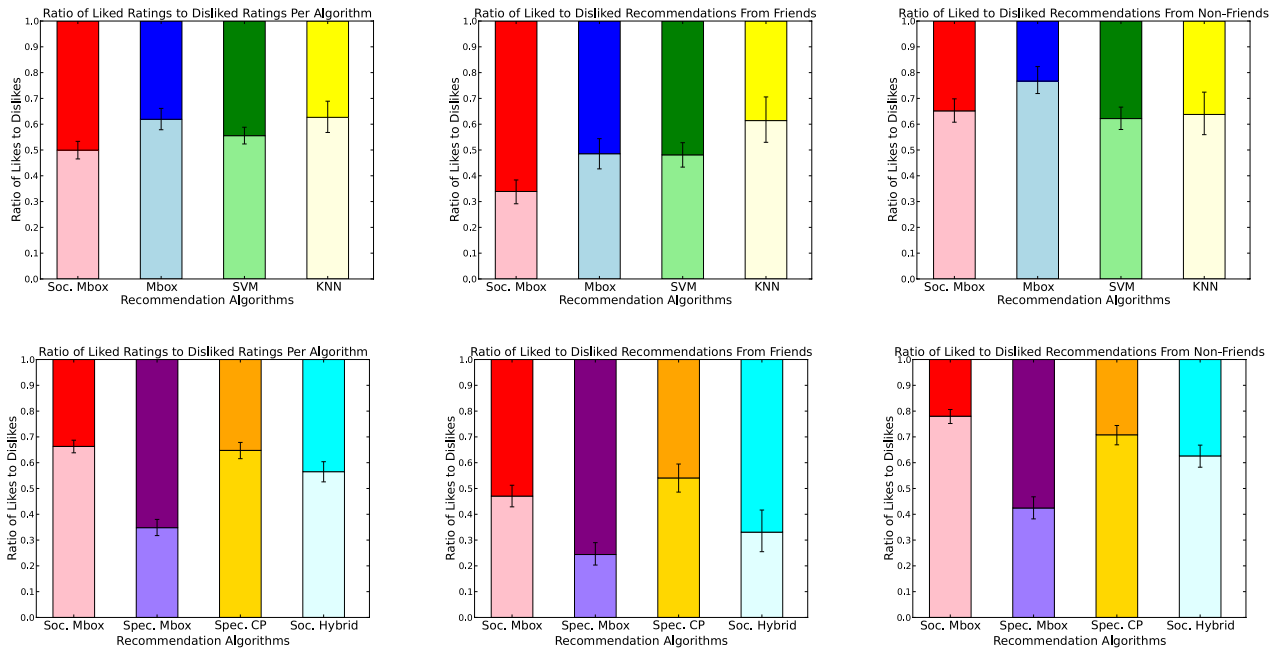
Figure 2: Stacked bar graphs of online results for the first (top) and second (bottom) user trials. The fraction of likes is displayed above the fraction of dislikes. Results are also broken down by link type: (left) all, (center) friend only, (right) non-friend only. 95% binomial proportion confidence intervals are shown.

5.3 User Behavior and Data Analysis

Overall, users had a general bias to like links recommended by friends more than non-friends; importantly, we note that users could see the names and comments of friends whose links were recommended, indicating the importance of context in recommendation. Next we briefly analyze other user behavior and data collected during both trials of the LinkR App that can be helpful in building future SCF systems.

5.3.1 Click evidence

In Figure 3(a), we observe the ratings of links that users clicked on. The most important thing we notice here is that even though users clicked on a link, they were somewhat likely to rate it as a dislike (roughly $\frac{2}{3}$ like to $\frac{1}{3}$ dislike).

One might hypothesize that perhaps users clicked on links more often with no description to find out what they were and most often disliked them — this might explain the high number of dislikes for clicked links. However, examining both Figures 3(b) and (c), we observe that whether a description was present had a relatively minor impact on whether a link was clicked or liked, so we cannot infer that the disliked links were simply the ones lacking a description.

Then the insight from this analysis is extremely important for SCF recommendation design because it states that click data is a somewhat weak indicator of likes and that even if one could predict clicks with perfect accuracy, this would only yield roughly $\frac{2}{3}$ accuracy for likes prediction.

5.3.2 Impact of Popularity

In Figures 3(d) and (e) we analyze the impact of global link popularity (in terms of total shares on Facebook) on how much LinkR App users liked a link. The trend is clear for both friend (d) and non-friend (e) links: users tend to like the most popular (top quartile) links the least compared

Individual Link Comments			User Survey Comments
Comment Type	#	%	want more control over
not interested	88	36.5%	recommendations made
wrong language	37	15.4%	(music, blogs, news)
really liked it!	35	14.5%	want option to see > 3
bad YouTube	25	10.4%	recommendations
seen it already	25	10.4%	links need description /
problem / dead	20	8.3%	context or explanation
outdated	7	2.9%	of recommendation
miscellaneous	4	1.7%	more variety, diversity

Table 2: Individual link comments (aggregated) and notable user survey requests (paraphrased).

to all other quartiles. In general, users tended to most prefer links that were somewhat popular (middle quartiles). From this we can infer that while the most popular links may be liked by the most people, they are not liked by everyone on average; this suggests that link popularity should not be weighted too heavily in determining link recommendations.

5.3.3 Link and Survey Comments

We collected individual link recommendation comments in the LinkR App as shown in Figure 1 and we also ran a user survey toward the end of both trials to collect qualitative feedback on the overall LinkR user experience. Due to space limitations, we briefly summarize this data in Table 2.

Table 2 (left) shows link comments classified into general classes and ranked by frequency. Users were easily annoyed (i) if they could not read the language of the link — this issue was addressed with a language filter in the second trial, (ii) if a YouTube or other link was inaccessible — YouTube links were some of the most popular links on Facebook and were also frequently removed for copyright violations, (iii) if a user had seen a similar link topic already — e.g., users quickly
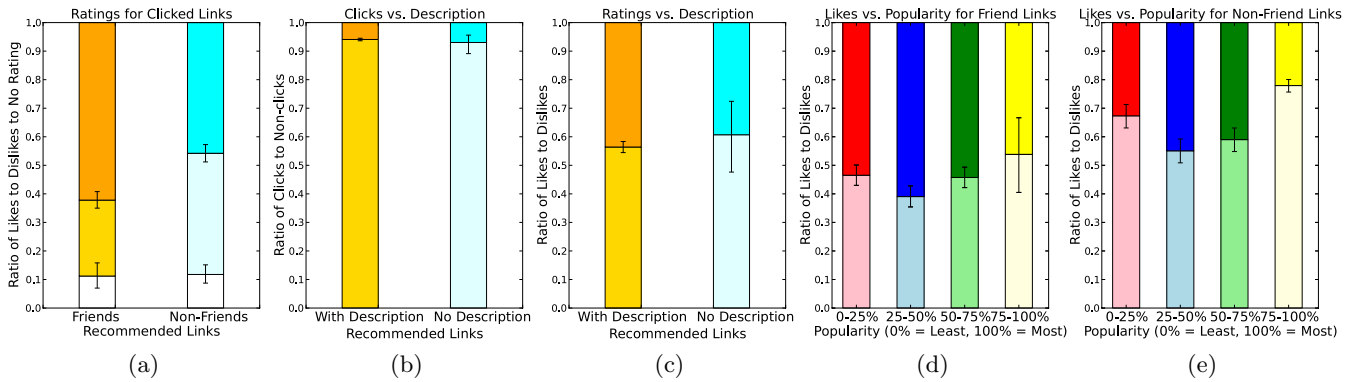
Figure 3: Stacked bar graphs for rating and click data collected during both trials. The fraction of likes (or clicks) is displayed above the fraction of dislikes (or non-clicks) – and above the fraction of not-rated links for (a). 95% binomial proportion confidence intervals are shown. (a) ratings for clicked links, (b) clicks vs. description, (c) ratings vs. description, (d) ratings vs. quartile of popularity for friends, and (e) non-friends.

got tired of repeated news links on Steve Jobs death during the second trial even though the links had different content, or (iv) if links were deemed to be outdated — e.g., a news article had been superceded by more recent information. On the other hand, if a user was pleased with a recommendation that they were not otherwise aware of, they often indicated this. In Table 2 (right) we show four notable user survey comments requesting modifications to the LinkR experience.

6. CONCLUSIONS

In this paper, we evaluated existing algorithms and proposed new algorithms for social collaborative filtering via the task of link recommendation on Facebook. Importantly, we outlined three main deficiencies in existing social collaborative filtering (SCF) matrix factorization (MF) techniques and proposed novel objective functions that (a) extended existing social regularization SCF approaches to incorporate user features by drawing on ideas from Matchbox [17], (b) modeled direct user-to-user information diffusion, and (c) modeled restricted common interests among all users (friend and non-friend) with social co-preference regularization.

We evaluated existing baselines and then evaluated algorithms based on optimization of these new objectives in Section 5 via live online user trials taking place over five months with over 100 Facebook App users and data from over 37,000 Facebook users. Results show that our novel feature-based social spectral regularization extension of Matchbox achieves an overall correct "likes" prediction rate of 65% — performance far exceeding that of all other algorithms trialed.

User feedback has opened up many new possibilities for further improving the SCF user experience. Future work can include: providing explanations for recommended content; incorporating *genre* features to provide a fine-grained model of user preference among different content; and enforcing diversity among recommendations to prevent redundancy.

Acknowledgements

7. REFERENCES

[1] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40:66–72, March 1997.

[2] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *ICDM-07*, 2007.

[3] J. Brown and P. Reinegen. Social ties and word-of-mouth referral behavior. *Journal of Consumer Research*, 1(3):350–362, 1987.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a Library for Support Vector Machines*, 2001.

[5] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[6] P. Cui, F. Wang, S. Liu, M. Ou, and S. Yang. Who should share what? item-level social influence prediction for users and posts ranking. In *International ACM SIGIR Conference (SIGIR)*, 2011.

[7] K. Lang. NewsWeeder: Learning to filter netnews. In *12th International Conference on Machine Learning ICML-95*, pages 331–339, 1995.

[8] W.-J. Li and D.-Y. Yeung. Relation regularized matrix factorization. In *IJCAI-09*, 2009.

[9] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, Aug 1989.

[10] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR-09*, 2009.

[11] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: Social recommendation using probabilistic matrix factorization. In *CIKM-08*, 2008.

[12] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM-11*, 2011.

[13] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS 14*, 2001.

[14] K. B. Petersen and M. S. Pedersen. The matrix cookbook, 2008.

[15] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40:56–58, March 1997.

[16] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS 20*, 2008.

[17] D. H. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online bayesian recommendations. In *WWW-09*, pages 111–120, 2009.

[18] E. B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Assoc. (JASA)*, 22:209–212, 1927.

[19] S. H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: Joint friendship and interest propagation in social networks. In *WWW-11*, 2011.

APPENDIX

A. GRADIENT-BASED OPTIMIZATION

We seek to optimize sums of the objectives in Section 3 and will use gradient descent for this purpose.

For the overall objective, the partial derivative w.r.t. parameters $\mathbf{a}$ are as follows:

$$\frac{\partial}{\partial \mathbf{a}} Obj = \frac{\partial}{\partial \mathbf{a}} \sum_i \lambda_i Obj_i = \sum_i \lambda_i \frac{\partial}{\partial \mathbf{a}} Obj_i \qquad (16)$$

Anywhere a sigmoidal transform occurs $\sigma(o[\cdot])$, we can easily calculate the partial derivatives as follows

$$\frac{\partial}{\partial \mathbf{a}} \sigma(o[\cdot]) = \sigma(o[\cdot])(1 - \sigma(o[\cdot]))\frac{\partial}{\partial \mathbf{a}} o[\cdot]. \qquad (17)$$

Hence anytime a $[\sigma(o[\cdot])]$ is optionally introduced in place of $o[\cdot]$, we simply insert $[\sigma(o[\cdot])(1-\sigma(o[\cdot]))]$ in the corresponding derivatives below.

Because most objectives below are not convex in U, V, or $\mathbf{w}$, we apply an *alternating gradient descent* approach [16]. In short, we take derivatives of U, V, and $\mathbf{w}$ in turn while holding the others constant. Then we apply gradient descent in a round-robin fashion until we've reached local minima for all parameters; for gradient descent on one of U, V, or $\mathbf{w}$ with the others held constant, we apply the L-BFGS optimizer [9] with derivatives defined below.

Before we proceed to our objective gradients, we define abbreviations for three useful vectors:

$$\begin{aligned}
\mathbf{s} &= U\mathbf{x} & \mathbf{s}_k &= (U\mathbf{x})_k; \ k = 1\dots K \\
\mathbf{t} &= V\mathbf{y} & \mathbf{t}_k &= (V\mathbf{y})_k; \ k = 1\dots K \\
\mathbf{r} &= U\mathbf{z} & \mathbf{r}_k &= (U\mathbf{z})_k; \ k = 1\dots K
\end{aligned}$$

All matrix derivatives used for the objectives below can be verified in [14].

$$\frac{\partial}{\partial U} Obj_{pmcf} = \frac{\partial}{\partial U} \sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2} \left(\underbrace{(R_{\mathbf{x},\mathbf{y}} - [\sigma] \overbrace{x^T U^T V \mathbf{y}}^{o_{\mathbf{x},\mathbf{y}}})}_{\delta_{\mathbf{x},\mathbf{y}}} \right)^2$$
$$= - \sum_{(\mathbf{x},\mathbf{y})\in D} \delta_{\mathbf{x},\mathbf{y}}[\sigma(o_{\mathbf{x},\mathbf{y}})(1 - \sigma(o_{\mathbf{x},\mathbf{y}}))]\mathbf{t}\mathbf{x}^T$$

$$\frac{\partial}{\partial V} Obj_{pmcf} = \frac{\partial}{\partial V} \sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2} \left(\underbrace{(R_{\mathbf{x},\mathbf{y}} - [\sigma] \overbrace{x^T U^T V \mathbf{y}}^{o_{\mathbf{x},\mathbf{y}}})}_{\delta_{\mathbf{x},\mathbf{y}}} \right)^2$$
$$= - \sum_{(\mathbf{x},\mathbf{y})\in D} \delta_{\mathbf{x},\mathbf{y}}[\sigma(o_{\mathbf{x},\mathbf{y}})(1 - \sigma(o_{\mathbf{x},\mathbf{y}}))]\mathbf{s}\mathbf{y}^T$$

$$\frac{\partial}{\partial U} Obj_{ru} = \frac{\partial}{\partial U} \frac{1}{2} \mathrm{tr}(U^T U) = U \qquad \frac{\partial}{\partial V} Obj_{rv} = V$$
$$\frac{\partial}{\partial \mathbf{w}} Obj_{rw} = \frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} = \mathbf{w}$$

$$\frac{\partial}{\partial U} Obj_{rs} = \frac{\partial}{\partial U} \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} \frac{1}{2} \left(\underbrace{S_{\mathbf{x},\mathbf{z}} - \mathbf{x}^T U^T U \mathbf{z}}_{\delta_{\mathbf{x},\mathbf{y}}} \right)^2$$
$$= - \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} \delta_{\mathbf{x},\mathbf{y}} U(\mathbf{x}\mathbf{z}^T + \mathbf{z}\mathbf{x}^T)$$

$$\frac{\partial}{\partial U} Obj_{rss} = \frac{\partial}{\partial U} \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} \frac{1}{2} S_{\mathbf{x},\mathbf{z}}^+ (\mathbf{x} - \mathbf{z})^T U^T U (\mathbf{x} - \mathbf{z})$$
$$= \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} S_{\mathbf{x},\mathbf{z}}^+ U(\mathbf{x} - \mathbf{z})(\mathbf{x} - \mathbf{z})^T$$

$$\frac{\partial}{\partial \mathbf{w}} Obj_{phy} = \frac{\partial}{\partial \mathbf{w}} \sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2} \left(\underbrace{R_{\mathbf{x},\mathbf{y}} - [\sigma] \overbrace{\mathbf{w}^T \mathbf{f}_{\mathbf{x},\mathbf{y}}}^{o_{\mathbf{x},\mathbf{y}}^1} -[\sigma]\mathbf{x}^T U^T V \mathbf{y}}_{\delta_{\mathbf{x},\mathbf{y}}} \right)^2$$
$$= - \sum_{(\mathbf{x},\mathbf{y})\in D} \delta_{\mathbf{x},\mathbf{y}}[\sigma(o_{\mathbf{x},\mathbf{y}}^1)(1 - \sigma(o_{\mathbf{x},\mathbf{y}}^1))]\mathbf{f}_{\mathbf{x},\mathbf{y}}$$

$$\frac{\partial}{\partial U} Obj_{phy} = \frac{\partial}{\partial U} \sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2} \left(\underbrace{R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{w}^T \mathbf{f}_{\mathbf{x},\mathbf{y}} - [\sigma]\overbrace{\mathbf{x}^T U^T V \mathbf{y}}^{o_{\mathbf{x},\mathbf{y}}^2}}_{\delta_{\mathbf{x},\mathbf{y}}} \right)^2$$
$$= - \sum_{(\mathbf{x},\mathbf{y})\in D} \delta_{\mathbf{x},\mathbf{y}}[\sigma(o_{\mathbf{x},\mathbf{y}}^2)(1 - \sigma(o_{\mathbf{x},\mathbf{y}}^2))]\mathbf{t}\mathbf{x}^T$$

$$\frac{\partial}{\partial V} Obj_{phy} = \frac{\partial}{\partial V} \sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2} \left(\underbrace{R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{w}^T \mathbf{f}_{\mathbf{x},\mathbf{y}} - [\sigma]\overbrace{\mathbf{x}^T U^T V \mathbf{y}}^{o_{\mathbf{x},\mathbf{y}}^2}}_{\delta_{\mathbf{x},\mathbf{y}}} \right)^2$$
$$= - \sum_{(\mathbf{x},\mathbf{y})\in D} \delta_{\mathbf{x},\mathbf{y}}[\sigma(o_{\mathbf{x},\mathbf{y}}^2)(1 - \sigma(o_{\mathbf{x},\mathbf{y}}^2))]\mathbf{s}\mathbf{y}^T$$

$$\frac{\partial}{\partial U} Obj_{cp} = \frac{\partial}{\partial U} \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2} \left(\underbrace{P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \mathbf{x}^T U^T \mathrm{diag}(V\mathbf{y})U\mathbf{z}}_{\delta_{\mathbf{x},\mathbf{z},\mathbf{y}}} \right)^2$$
$$= - \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \delta_{\mathbf{x},\mathbf{z},\mathbf{y}} \mathrm{diag}(V\mathbf{y})U(\mathbf{x}\mathbf{z}^T + \mathbf{z}\mathbf{x}^T)$$

In the following, $\circ$ is the Hadamard elementwise product:

$$\frac{\partial}{\partial V} Obj_{cp} = \frac{\partial}{\partial V} \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2} (P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \mathbf{x}^T U^T \mathrm{diag}(V\mathbf{y})U\mathbf{z})^2$$
$$= \frac{\partial}{\partial V} \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2} \left(\underbrace{P_{\mathbf{x},\mathbf{z},\mathbf{y}} - (\overbrace{U\mathbf{x}}^{\mathbf{s}} \circ \overbrace{U\mathbf{z}}^{\mathbf{r}})^T V \mathbf{y}}_{\delta_{\mathbf{x},\mathbf{z},\mathbf{y}}} \right)^2$$
$$= - \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \delta_{\mathbf{x},\mathbf{z},\mathbf{y}}(\mathbf{s} \circ \mathbf{r})\mathbf{y}^T$$