New Objective Functions for Social Collaborative Filtering

Joseph Noel*, Scott Sanner†*, Khoi-Nguyen Tran*, Peter Christen*, Lexing Xie*
Edwin Bonilla†*, Ehsan Abbasnejad†*
*Australian National University, †National ICT Australia
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

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 demonstrate that optimizing the new objectives significantly outperforms a variety of CF and social CF baselines on live user trials in a custom-developed Facebook App involving data collected over two months from over 100 App users and their 34,000+ 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 a user population under the assumption that similar users will have similar preferences.

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, 18, 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* to learn user-user similarities in the latent space.

(b) **Direct learning of user-to-user information dif-**

fusion: 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 three months from over 100 active App users and their 34,000+ friends. Results confirm that each of the proposed objectives — feature-based social regularization, direct modeling of information diffusion features, and co-preference regularization — improve performance against a range of existing SCF baselines.

In addition, deploying our algorithm on a real social network provided us with a number of interesting observations on training and user behavior discussed in Section 5.3. For example, click feedback correlates very 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 on a variety of metrics with further analysis of behavioral trends 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 $D = \{(\mathbf{x}, \mathbf{y}) \to R_{\mathbf{x},\mathbf{y}}\}$ 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 $C = \{(\mathbf{x}, \mathbf{z}, \mathbf{y}) \to P_{\mathbf{x},\mathbf{z},\mathbf{y}}\}$ 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:

$$Int_{\mathbf{x},\mathbf{z}} = \frac{\#\text{ interactions between } \mathbf{x} \text{ and } \mathbf{z}}{\frac{1}{N^2}\sum_{\mathbf{x}',\mathbf{z}'} \#\text{ interactions between } \mathbf{x}' \text{ and } \mathbf{z}'}$$

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

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 $\mathbf{f}_{\mathbf{x},\mathbf{y}}$. 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 used in the SVM for the Facebook link recommendation task evaluated in this paper are defined in Section 4.2 — 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 computations [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 *user* nearest neighbors of $\mathbf{x}$ that have also given a rating for $\mathbf{y}$ and let $S_{\mathbf{x},\mathbf{z}}$ be the cosine similarity (i.e., normalized dot product) 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

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

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) into a product $R \approx U^T V$ of real-valued rank-K matrices U and V:

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

In this initial MF setting, we *do not* leverage user and item features; hence, the $\mathbf{x}$ and $\mathbf{y}$ indices simply pick out the respective rows and 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* [18, 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., $\bar{S}_{\mathbf{x},\mathbf{z}} = \sigma(S_{\mathbf{x},\mathbf{z}})$):

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

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):

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

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:

$$Obj = \sum_i \lambda_i Obj_i \tag{9}$$

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 last 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 objective in the following log likelihood form:

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

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:

$$Obj_{ru} = \frac{1}{2}\|U\|_{\mathrm{Fro}}^2 = \frac{1}{2}\operatorname{tr}(U^T U) \qquad Obj_{rv} = \frac{1}{2}\operatorname{tr}(V^T V)$$
$$Obj_{rw} = \frac{1}{2}\|\mathbf{w}\|_2^2 = \frac{1}{2}\mathbf{w}^T\mathbf{w} \tag{11}$$

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, it *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* [18, 6] by incorporating user features into that objective:

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

Alternately, we could extend *social spectral regulariza-*

tion [12, 8] by incorporating user features into the objective:

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

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. 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 combine the output of the linear regression prediction with the Matchbox prediction, to get a hybrid objective component. The full objective function for this hybrid model then becomes

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

While again we note that this simple hybrid MF plus linear regression objective is straightforward, the ability to use joint user and item features to model information diffusion between users turns out to be extremely powerful in our later experiments.

3.3.3 Co-preference Regularization (Obj_{rsc})

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 latent 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 latent 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

$$Obj_{rsc} = \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in C} \frac{1}{2} (P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \langle U\mathbf{x}, U\mathbf{z} \rangle_{V\mathbf{y}})^2 \quad (15)$$
$$= \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in C} \frac{1}{2} (P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \mathbf{x}^T U^T \operatorname{diag}(V\mathbf{y}) U\mathbf{z})^2$$

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 find correlations between users who are not friends — this important observation will indeed surface in our forthcoming experimental evaluation.

4. EVALUATION FRAMEWORK

In this section we discuss our Facebook Link Recommendation (LinkR) application, definitions for notation in the context of this application, 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 everyday, 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 data that have been shared by users and their friends on Facebook.
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. Following retraining, 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 the (S)CF link recommendation algorithms and user assignments will be discussed shortly, but first we cover the specific data collected by the LinkR App and made available for use by the recommendation algorithms.

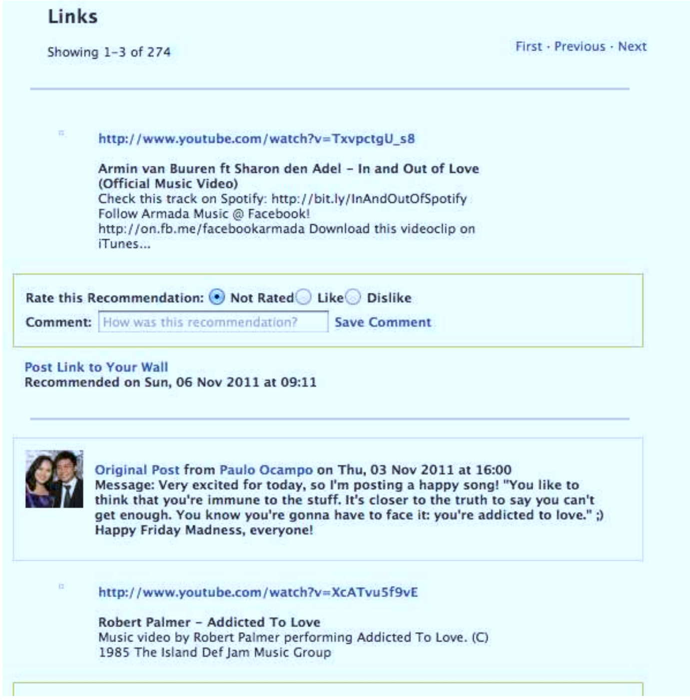

Figure 1: The Facebook LinkR App showing two link recommendations to a user. The first link recommendation is from a non-friend and hence only shows the link description. The second link recommendation is from a friend and includes the friend's commentary on the link as well as the link description. Users have the option of liking or disliking each recommendation as well as providing feedback.

4.2 Facebook Data Collected

At its peak membership, 108 users had elected to install the Facebook App developed for this project. From this user base, we were able to gather data on 34,860 users and 437,023 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 such indicators are enforced to be 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

The Facebook social network interactions between users $\mathbf{x}$ and $\mathbf{z}$ that we count (equally weighted) to define *# interactions between* $\mathbf{x}$ *and* $\mathbf{z}$ in Section 2.1 are defined as follows:

1. Being friends.
2. Posting, liking, or commenting on an item (link, photo, video, photo, or message) on a user's wall.
3. Being tagged together in the same photo or video.
4. Tagging themselves as attending the same school or class, playing sports together, working together for the same company or on the same project.

4.3 Live Online Recommendation Trials

For the recommendations made to the LinkR application users, we select only links posted in the most recent two weeks that the user has not already liked (or disliked). We use only the links from the last two weeks since an informal user study has indicated a preference for recent links. Indeed, older links have a greater chance of being outdated and are also likely to represent broken links that are no longer accessible. We have chosen to recommend three links per day with the aim of avoiding position bias and information overload that may occur with longer recommendation lists.

For the live trials, Facebook users who installed the LinkR application were *randomly assigned one of four algorithms in each of two trials*. Specific algorithms trialed will be discussed in Section 5. LinkR users were not informed which algorithm was assigned to them. As demonstrated in Figure 1, we distinguish our recommended links into two major classes: (1) links that were posted by the LinkR user's friends, and (2) links that were posted by users other than the LinkR user's friends. LinkR users were encouraged to rate each link as like or dislike. In turn these ratings became part of the training data for the recommendation algorithms, and thus were used to improve the performance of the algorithms over time. LinkR users were allowed to provide feedback comments on specific links if they wished;

Algorithm	Users
Soc. Mbox	26
Mbox	26
SVM	28
KNN	28

Algorithm	Users
Soc. Mbox	26
Spec. Mbox	25
Spec. CP	27
Soc. Hybrid	25

Table 1: Number of users assigned per algorithm in the first trial (left) and second trial (right).

based on repeated user commentary on annoyances from the first trial period, in the second trial period we avoided recommendations of (i) non-English links and (ii) links lacking any available textual description to display.

5. EMPIRICAL RESULTS

In this section we discuss the algorithms evaluated in our two online trials[1] and additional analysis regarding trends and patterns in our social recommendation setting.

5.1 First Trial

In our first trial, our objective was to evaluate four CF and SCF approaches to establish which was the most promising direction for SCF extension:

1. **k-Nearest Neighbor (KNN)**: See Section 2.3.1.
2. **Support Vector Machines (SVM)**: See Section 2.2.
3. **Matchbox (Mbox)**: simple Matchbox CF

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

4. **Social Matchbox (Soc. Mbox)**: feature-based *socially regularized* Matchbox SCF

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

KNN and MBox may be viewed as pure CF methods. As discussed previously, both SVM and Soc. Mbox may be viewed as SCF methods; all objectives for these were given in Section 3 and were both optimized via gradient descent as outlined in Appendix A.

The first live user trial was run from August 25, 2011 to October 13, 2011 with 108 users and yielded 2,493 combined like and dislike ratings of recommended links over the 49 day period. Algorithms were assigned randomly to the users with assignment counts shown in Table 1 (left).

As shown in Figure 2, Soc. Mbox was the best performing algorithm in the first trial and in fact was the only algorithm to receive more like ratings than dislike ratings. This suggests that using social regularization in conjunction with MF-based CF does indeed provide more useful information than simple MBox without social regularization. We also note a significant drop in performance between recommending friend links and recommending non-friend links, indicating that users had a bias to like links recommended by friends more (importantly, we note that users could see the names and comments of friends whose links were recommended).

[1]All code used in these experiments is available at `http://code.google.com/p/social-recommendation/`. The conditions of our ethics approval #2011/142 from the Australian National University for conducting human trials on Facebook require our privacy policy (`http://dmm.anu.edu.au/linkr/website/pp.php`) to prohibit public sharing of data collected during these experiments.

5.2 Second Trial

For the second online trial, we again chose four algorithms to randomly assign to the LinkR application users. Social Matchbox was included again as a baseline since it was the best performing algorithm in the first trial. The remaining three algorithms were all relatively orthogonal extensions (or variants) of Social Matchbox based on the *three novel objective functions* defined in Section 3.3:

- **Social Matchbox (Soc. Mbox)** : unchanged.
- **Spectral Matchbox (Spec. Mbox)**:

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

- **Social Hybrid (Soc. Hybrid)**:

$$\lambda_{phy} Obj_{phy} + \lambda_{rs} Obj_{rs} + \lambda Obj_{ru} + \lambda Obj_{rv} + \lambda Obj_{rw}$$

- **Spectral Co-preference (Spec. CP)**:

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

All objectives are defined in Section 3 and optimized via gradient descent as outlined in Appendix A.

The second trial with the above algorithms ran from October 13, 2011 to November 5, 2011 with 103 users and yielded 1,435 like/dislike ratings of recommended links over the 24 day period; 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. The distribution of the algorithms to the users is shown in Table 1 (right).

Results for the second trial are shown in Figure 3, following are the key observations:

- Soc. MBox did not perform as well in the second trial as it had in the first trial. One key difference in the second trial is that all of the first trial data was available for training and thus Soc. MBox may have proved more difficult to optimize given this larger quantity of training data.

- Spec. Mbox clearly performed the best in the second trial and this suggests that spectral social regularization is likely a better method of regularization than the original social regularization variant.

- Soc. Hybrid does an excellent job of capturing information diffusion in the social network and hence performs best on recommending friend links and most poorly on non-friend links (where there is no direct user-to-user information diffusion). Soc. Hybrid slightly outperforms Spec. MBox, but not statistically significantly given the more limited data in the second trial. Nonetheless, Soc. Hybrid clearly outperforms its Soc. MBox variant that lacks information diffusion features.

- Spec. CP leads to a statistically significant improvement on recommending non-friend links over Soc. MBox and Soc. Hybrid, which perform poorly on recommending non-friend links. Clearly then Spec. CP is benefitting somewhat in learning from copreferences since it can socially regularize across two users that are *not* friends, unlike the other approaches.

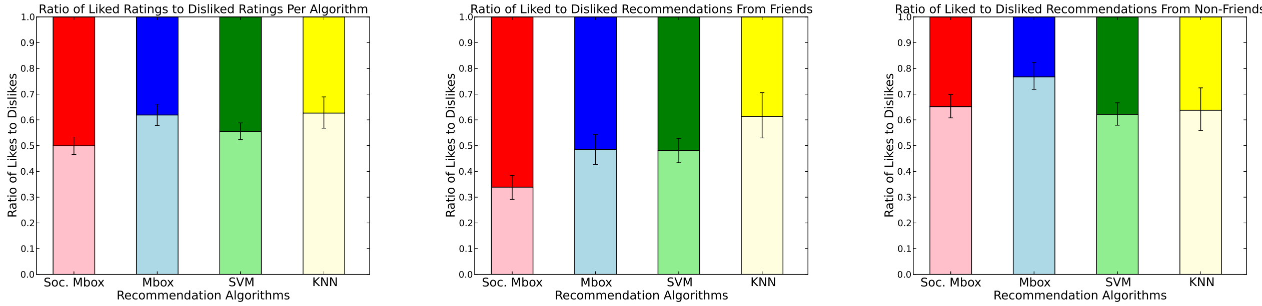

Figure 2: Stacked bar graphs of online results for the first user trial. The fraction of likes is displayed above the fraction of dislikes. (left) all links, (center) friend links, (right) non-friend links. The 95% confidence interval on all results is $< \pm 0.02$ so all differences are signficant except for MBox and SVM in the center and the virtual tie between three algorithms on the right.

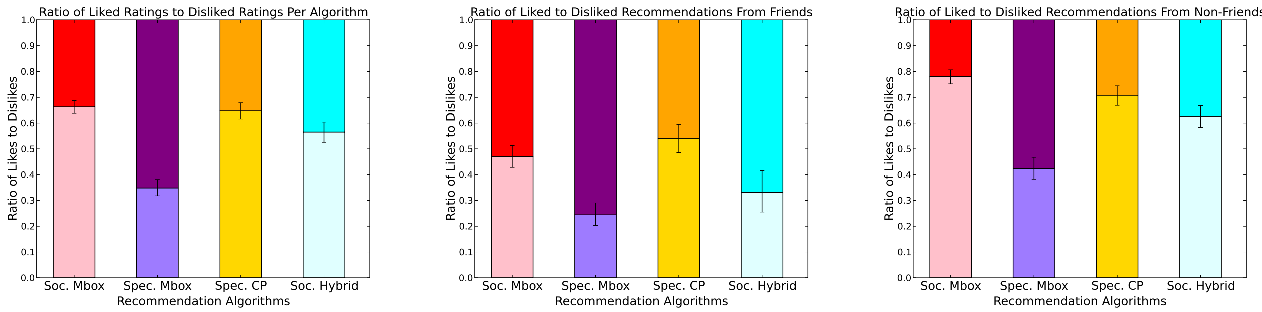

Figure 3: Stacked bar graphs of online results for the second user trial. The fraction of likes is displayed above the fraction of dislikes. (left) all links, (center) friend links, (right) non-friend links. The 95% confidence interval on all results is $< \pm 0.026$ so all differences except Spec. Mbox and Soc. Hybrid in the center graph are significant.

5.3 User Behavior Analysis

Here we briefly analyze user behavior during both trials of the Facebook LinkR App that can be helpful in building future SCF systems.

5.3.1 *Click evidence*

In Figure 4, we observe the ratings of links that users clicked on. The most important thing we notice in Figure 4 (left) is that even though users clicked on a link, they were somewhat likely to rate it as a 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 Figure 4 (center) and (right) for non-friend links, we observe that whether a description was present had very little impact on whether a link was liked or not, so we cannot infer that all 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 very weak indicator of likes and explicit feedback should be used if at all possible.

5.3.2 *Impact of Popularity*

In Figure 5 we analyze the impact of global link popularity (in terms of total shares on Facebook) on how much Facebook LinkR App users liked a link. The trend is clear for both friend (left) and non-friend (right) links: users tend to like the most popular (top quartile) links the least, or at least as much as the least popular (bottom quartile) links. In general users tended to prefer links that were somewhat popular (second highest quartile). From this we can infer that link popularity should not be used too heavily in determining link recommendations since clearly the most popular links are not liked the most on average.

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) learns user similarity from features using social spectral regularization, (b) models direct user-user information diffusion by extending Matchbox [17], and (c) models restricted common interests with social co-preference regularization.

We evaluated existing baseline (variants) and then evaluated these new algorithms in Section 5 in live online user trials with over 100 Facebook App users and data for over 30,000 unique Facebook users showing the settings in which

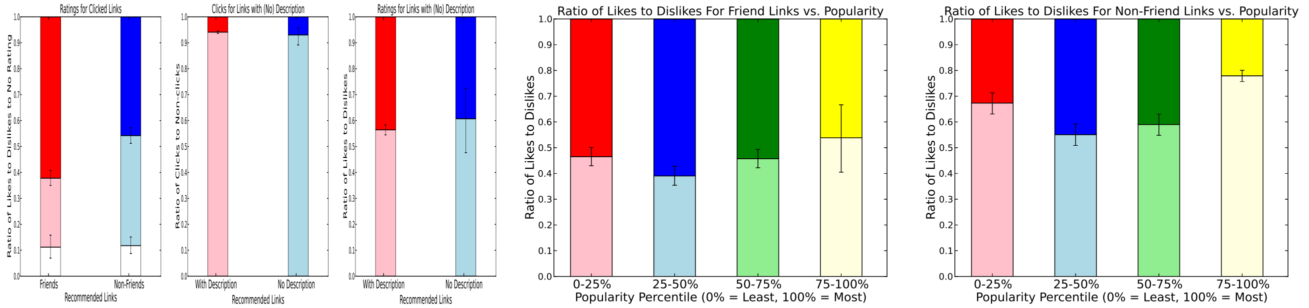

Figure 4: Stacked bar graphs of online results for the first user trial. The fraction of likes (or clicks) is displayed above the fraction of dislikes (or non-clicks) – and above the fraction of not-rated links for the left-most figure. (left) ratings for clicked links, (center) click rates if text description not present, (right) percentage liked if click description not present. Stacked bar graphs of online results for the first user trial. The fraction of likes is displayed above the fraction of dislikes. Shown are the ratings vs. quartile of popularity for (left) friend and (right) non-friend links. Stacked bar graphs of online results for the first user trial. The fraction of likes is displayed above the fraction of dislikes. Shown are the ratings vs. quartile of popularity for (left) friend and (right) non-friend links.

Figure 5: ***

our newly proposed objective functions helped the most in comparison to baselines. As such, this paper represents one concrete step forward in SCF algorithms based on top-performing MF methods and their ability to fully exploit the breadth of information available on social networks.

Our work opened up many new possibilities for further improving SCF algorithms and systems. Future work can include: incorporating content *genre* feature to provide a fine-grained model about user preference among different types of links; enforcing diversity among recommended links to prevent redundancy; and devising active learning strategies to better explore the social recommendation space.

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 *Advances in Neural Information Processing Systems 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 *Advances in Neural Information Processing Systems*, volume 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] 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 two useful vectors:

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

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

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

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

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

$$\frac{\partial}{\partial \mathbf{w}} Obj_{rw} = \frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} = \mathbf{w}$$

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

$$\begin{aligned} \frac{\partial}{\partial U} Obj_{rss} &= \frac{\partial}{\partial U} \sum_{\mathbf{x}} \sum_{\mathbf{z} \in friends(\mathbf{x})} \frac{1}{2} S^+_{\mathbf{x},\mathbf{z}} (\mathbf{x}-\mathbf{z})^T U^T U (\mathbf{x}-\mathbf{z}) \\ &= \sum_{\mathbf{x}} \sum_{\mathbf{z} \in friends(\mathbf{x})} S^+_{\mathbf{x},\mathbf{z}} U (\mathbf{x}-\mathbf{z})(\mathbf{x}-\mathbf{z})^T \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} Obj_{phy} &= \frac{\partial}{\partial \mathbf{w}} \sum_{(\mathbf{x},\mathbf{y}) \in D} \frac{1}{2} \Big(\underbrace{R_{\mathbf{x},\mathbf{y}} - [\sigma] \overbrace{\mathbf{w}^T \mathbf{f}_{\mathbf{x},\mathbf{y}}}^{o^1_{\mathbf{x},\mathbf{y}}} - [\sigma] \mathbf{x}^T U^T V \mathbf{y}}_{\delta_{\mathbf{x},\mathbf{y}}} \Big)^2 \\ &= - \sum_{(\mathbf{x},\mathbf{y}) \in D} \delta_{\mathbf{x},\mathbf{y}} [\sigma(o^1_{\mathbf{x},\mathbf{y}})(1-\sigma(o^1_{\mathbf{x},\mathbf{y}}))] \mathbf{f}_{\mathbf{x},\mathbf{y}} \end{aligned}$$

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

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

$$\begin{aligned} \frac{\partial}{\partial U} Obj_{rsc} &= \frac{\partial}{\partial U} \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in C} \frac{1}{2} \Big(\underbrace{P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \mathbf{x}^T U^T \operatorname{diag}(V\mathbf{y}) U \mathbf{z}}_{\delta_{\mathbf{x},\mathbf{z},\mathbf{y}}} \Big)^2 \\ &= - \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in C} \delta_{\mathbf{x},\mathbf{z},\mathbf{y}} \operatorname{diag}(V\mathbf{y}) U (\mathbf{x}\mathbf{z}^T + \mathbf{z}\mathbf{x}^T) \end{aligned}$$

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

$$\begin{aligned} \frac{\partial}{\partial V} Obj_{rsc} &= \frac{\partial}{\partial V} \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in C} \frac{1}{2} (P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \mathbf{x}^T U^T \operatorname{diag}(V\mathbf{y}) U \mathbf{z})^2 \\ &= \frac{\partial}{\partial V} \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in C} \frac{1}{2} \Big(\underbrace{P_{\mathbf{x},\mathbf{z},\mathbf{y}} - (\overbrace{U\mathbf{x}}^{\mathbf{s}} \circ \overbrace{U\mathbf{z}}^{\mathbf{r}})^T V \mathbf{y}}_{\delta_{\mathbf{x},\mathbf{z},\mathbf{y}}} \Big)^2 \\ &= - \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in C} \delta_{\mathbf{x},\mathbf{z},\mathbf{y}} (\mathbf{s} \circ \mathbf{r}) \mathbf{y}^T \end{aligned}$$