New Objective Functions for Social Collaborative Filtering

Joseph Noel
ANU; Canberra, AU
jinonoel@gmail.com

Scott Sanner
NICTA & ANU; Canberra, AU
first.last@nicta.com.au

Khoi-Nguyen Tran
ANU; Canberra, AU
first.last@anu.edu.au

Peter Christen
ANU; Canberra, AU
first.last@anu.edu.au

Lexing Xie
ANU; Canberra, AU
first.last@anu.edu.au

Edwin Bonilla
NICTA & ANU; Canberra, AU
first.last@nicta.com.au

ABSTRACT

This paper examines the problem of social *collaborative filtering* (CF) algorithms 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; specifically, existing matrix factorization methods (a) do not exploit user features in all aspects of learning, (b) do not permit directly modeling user-to-user information diffusion, and (c) use objectives that treat users as globally (dis)similar even though they may only be (dis)similar in specific latent areas of interest. This paper proposes a unified framework for social CF matrix factorization that addresses (a)–(c) by introducing novel objective functions for training. We demonstrate that optimizing these 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) [14] 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 [9, 10, 17, 6, 11, 8] extend *matrix factorization* (MF) techniques such as [15] 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.

Given the strong performance of existing MF approaches to SCF, we aim to further improve on their performance in this paper. To do this, we have first identified a number of major deficiencies of existing SCF MF objective functions:

(a) **Feature-based user similarity learning:** Existing SCF MF objectives do not fully exploit user features when learning user similarity based on observed interactions. For example, one might enforce that two users are similar when their gender matches, but existing SCF MF objectives do not allow *learning* such a property in cases where the data supports it.

(b) **Direct learning of user-to-user information diffusion:** Existing SCF MF objectives do not permit directly modeling user-to-user information diffusion ac-

cording 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.

(c) **Learning restricted interests:** Existing SCF MF objectives treat users as globally (dis)similar although they may only be (dis)similar in specific latent areas of 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. Existing SCF MF objective functions do not leverage *co-preference (dis)agreement* (i.e., whether two users agreed or disagreed in their rating of the *same* item) to encourage *learning* restricted latent areas where two users are (dis)similar.

This paper addresses (a)–(c) by proposing novel objective functions in a unified latent factorization framework for SCF. We present results of our proposed recommendation algorithms in online human trials of a custom-developed Facebook App for link recommendation that involves data collected over three months from over 100 active App users and their 34,000+ friends. These results show that the extensions proposed to resolve (a)–(c) outperform a wide range of previously existing SCF algorithms.

To this end, the remaining sections of the paper are organized as follows:

- **Section 2 – Definitions and Background:** We define mathematical notation used throughout the paper along with a discussion of related work and all baseline comparison methods we use in this paper.

- **Section 3 – New Objective Functions for SCF:** We propose novel extensions to address (a)–(c) within a unified mathematical framework. Gradients for optimizing these functions are provided in Appendix A.

- **Section 4 – Evaluation Framework:** We discuss the details of our Facebook App for link recommendation as well as our evaluation methodology for online (live user trial) experimentation.

- **Section 5 – Empirical Results:** In our first online trial, we identify an MF-based SCF method as the best-performing baseline algorithm. This directly motivated our decision to extend MF-based methods in Section 3 and hence we compare these new objectives to the best-performing SCF MF baseline from the first trial. After showing where each extension improves on the baseline, we propose a number of additional experiments to better understand trends and patterns in our social recommendation setting such as the use of explicit like/dislike feedback vs. link click information, the relationship between like popularity and user preference, and importantly, why Steve Jobs passing almost ruined our second trial results and what insight we can gain from such extreme events.

- **Section 6 – Conclusions:** We summarize our conclusions from the empirical evaluations and outline future research directions stemming from the current work.

All combined, this paper represents a critical step forward in powerful SCF recommendation algorithms based on latent factorization methods and their ability to fully exploit the breadth of information available on social networks.

2. DEFINITIONS AND BACKGROUND

Collaborative filtering (CF) [14] 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) simply 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 attempt to 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 $\mathbf{x} \in \{1 \dots N\}$.

- 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 $\mathbf{y} \in \{1 \dots M\}$.

- 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 $C = \{(\mathbf{x}, \mathbf{z}, \mathbf{y}) \to P_{\mathbf{x},\mathbf{z},\mathbf{y}}\}$ 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 4.2.3.

In addition, we can 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 in this paper. 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 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), *item features* such as popularity (number of times shared on Facebook) and as a crude, but effective *information diffusion* [3] model, binary features for *each* friend $\mathbf{z} \in friends_{\mathbf{x}}$ indicating if $\mathbf{z}$ liked (or disliked) $\mathbf{y}$.

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 items is far fewer than the number of users, it has been found that the item-based approach usually provides better predictions as well as being

more efficient in computations [2]; this holds for the evaluation in this paper as well so we focus on the item-based approach here.

Given a user $\mathbf{x}$ and an item $\mathbf{y}$, let $N(\mathbf{y} : \mathbf{x})$ be the set of *item* nearest neighbors of $\mathbf{y}$ that have also been rated by $\mathbf{x}$ and let $S_{\mathbf{y},\mathbf{y'}}$ be the cosine similarity (i.e., normalized dot product) between two vectors of item ratings for $\mathbf{y}$ and $\mathbf{y'}$ with an index for each user that has rated both items. 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

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

2.3.2 *Matrix Factorization (MF) Models*

Another common approach to CF attempts to factorize an (incomplete) matrix of ratings $R_{\mathbf{x},\mathbf{y}}$ (users $\mathbf{x}$ in columns, items $\mathbf{y}$ in rows) into a product $R \approx U^T V$ of real-valued rank-K matrices, where U and V defined as follows:

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

In this section, we consider the case where we *do not* have user and item features and thus we simply use $\mathbf{x}$ and $\mathbf{y}$ as indices to pick out the respective rows and columns of U and V so that $U_{\mathbf{x}}^T V_{\mathbf{y}}$ acts as a measure of affinity between user $\mathbf{x}$ and item $\mathbf{y}$ in respective K-dimensional latent spaces $U_{\mathbf{x}}^T$ and $V_{\mathbf{y}}$.

However, there remains the question of how we learn the matrix factorization components U and V in order to optimally carry out this user and item projection, especially since R is an ***. The answer is simple: we need only define the objective we wish to maximize as a function of U and V and then use gradient descent to optimize them. Formally, we can optimize the following MF objective:

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

Taking gradients w.r.t. U and V here (holding one constant while taking the derivative w.r.t. the other), we can easily define an alternating gradient descent approach to approximately optimize these objectives and hence determine good projections U and V that (locally) minimize the reconstruction error of the observed responses $R_{\mathbf{x},\mathbf{y}}$ [15].

2.3.3 *Social Collaborative Filtering (SCF)*

Our evaluation of previously defined CF methods in Section ?? shows that MF methods generally work best for the SCF application evaluated in this paper (Section 4). Hence, we focus on SCF extensions of MF methods and review prior work here.

There are essentially two general classes of MF methods applied to SCF that we discuss below. All of the social MF methods defined to date *do not* make use of user or item features and hence $\mathbf{x}$ and $\mathbf{z}$ below should be treated as user indices as defined previously for the non-feature case.

The first class of social MF methods can be termed as *social regularization* approaches in that they somehow constrain the latent projection represented by U. There are two social regularization methods that directly constrain U for user $\mathbf{x}$ and $\mathbf{z}$ based on evidence $S_{\mathbf{x},\mathbf{z}}$ of interaction between

x and **z**. The first class of methods are simply termed *social regularization* [17, 6]

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

while the second class of methods are termed *Social spectral regularization* [11, 8]

$$\sum_{i} \sum_{\mathbf{z} \in friends_{\mathbf{x}}} \frac{1}{2} S^+_{\mathbf{x},\mathbf{z}} \|U_{\mathbf{x}} - U_{\mathbf{z}}\|_2^2 \qquad (6)$$

We refer to the latter as *spectral* regularization methods since they are identical to the objectives used in spectral clustering [12].

The *SoRec* system [10] 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 $\sigma(o) = \frac{1}{1+e^{-o}}$:

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

The second class of SCF MF approaches represented by the single examplar of the *Social Trust Ensemble* (STE) [9] can be termed as a *weighted average* approach since this approach simply composes a prediction for item **y** from a weighted average of a user **x**'s predictions *as well as* their friends (**z**) predictions (as evidenced by the additional $\sum_{\mathbf{z}}$ in the objective below):

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

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

In this section we introduce a unified matrix factorization framework for expressing all MF objectives evaluated in this paper, including newly proposed objectives to address observed deficiencies of SCF MF methods as outlined in Section 1

3.1 Mathematical Preliminaries

3.1.1 Composable Objectives

We take a composable approach to (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.

Most target predictions are binary classification-based ($\{0,1\}$), 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 $[0,1]$. In places 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. Nonetheless we include the possibility of a σ transform here where appropriate since it may prove useful in other settings.

3.2 Existing Objective Functions

3.2.1 Matchbox Matrix Factorization (Obj_{pmcf})

If we *do* have user and item features, we can respectively represent the latent projections of user and item as $(U\mathbf{x})_{1...K}$ and $(V\mathbf{y})_{1...K}$ 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 **x** and item **y**. Now we are faced with what objective to optimize in this feature-based MF case; fortunately, the answer comes in the form of the basic objective function used in Matchbox [16] — although Matchbox used Bayesian optimization methods, we can easily express its objective in the following log likelihood form:

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

3.2.2 L_2 Regularization of U and V (Obj_{ru}, Obj_{rv})

To help in generalization, it is important to regularize the free parameters U and V to prevent overfitting in the presence of sparse data. This can be done with the L_2 regularizer that models a prior of 0 on the parameters. The regularization component for U is

$$Obj_{ru} = \frac{1}{2}\|U\|_{\text{Fro}}^2 = \frac{1}{2}\text{tr}(U^T U) \qquad (11)$$

We can state the identical case substituting V for U.

In the same manner as U and V, it is important to regularize the free parameter **w** to prevent overfitting in the presence of sparse data. This can again be done with the L_2 regularizer that models a prior of 0 on the parameters. The objective and gradient for the L_2 regularizer for **w** is:

$$Obj_{rw} = \frac{1}{2}\|\mathbf{w}\|_2^2 = \frac{1}{2}\mathbf{w}^T \mathbf{w} \qquad (12)$$

$$ \qquad (13)$$

3.3 New Objective Functions

3.3.1 Social Regularization (Obj_{rs})

The social aspect of SCF is implemented as a regularizer on the user matrix U. What this objective component does is constrain users with a high similarity rating to have the same values in the latent feature space. This models the assumption that users who are similar socially should also have similar preferences for items.

This method is an extension of existing SCF techniques [17, 6] described in Section 2.3.2 that constrain the latent space to enforce users to have similar preferences latent representations when they interact heavily. Like Matchbox which extends regular matrix factorization methods by making use of user and link features, our extension to the Social Regularization method incorporates user features to learn similarities between users in the latent space.

$$Obj_{rs} = \sum_{\mathbf{x}} \sum_{\mathbf{z} \in friends(\mathbf{x})} \frac{1}{2}(S_{\mathbf{x},\mathbf{z}} - \langle U\mathbf{x}, U\mathbf{z} \rangle)^2 \qquad (14)$$

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

3.3.2 Social Spectral Regularization (Obj_{rss})

As we did with the Social Regularization method in Section 3.3.1, we build on ideas used in Matchbox [16] to extend social spectral regularization [11, 8] by incorporating user features into the objective. The objective function for our extension to social spectral regularization is:

$$Obj_{rss} = \sum_{\mathbf{x}} \sum_{\mathbf{z} \in friends(\mathbf{x})} \frac{1}{2} S_{\mathbf{x},\mathbf{z}}^+ \|U\mathbf{x} - U\mathbf{z}\|_2^2 \qquad (15)$$

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

3.3.3 Hybrid Objective (Obj_{phy})

As specified in Chapter 1, one weakness of MF methods is that they cannot model joint features over user and items, and hence the cannot model direct user-user information diffusion. Information diffusion models the unidirectional flow of links from one user to another (i.e., one user likes/shares what another user posts). We believe that this information will be useful for SCF, and is lacking in current SCF methods.

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

We make use of the $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ features detailed in Section 4.2 to make the linear regressor. $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ models user-user information diffusion because it is a joint feature between users and the links. It allows us to learn information diffusion models whether certain users always likes content from another specific user.

Using $\langle \cdot, \cdot \rangle$ to denote an inner product, we define a weight vector $\mathbf{w} \in \mathbb{R}^F$ such that $\langle \mathbf{w}, \mathbf{f}_{\mathbf{x},\mathbf{y}} \rangle = \mathbf{w}^T \mathbf{f}_{\mathbf{x},\mathbf{y}}$ is the prediction of the system. The objective of the linear regression component is therefore

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

We combine the output of the linear regression objective with the Matchbox output $[\sigma]\mathbf{x}^T U^T V y$, to get a hybrid objective component. The full objective function for this hybrid model is

$$\sum_{(\mathbf{x},\mathbf{y}) \in D} \frac{1}{2} (R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{w}^T \mathbf{f}_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{x}^T U^T V y)^2 \qquad (16)$$

3.3.4 Social 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 music, but different interests concerning politics. The social co-preference regularizers aim 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 tailor the latent space similarities or dissimilarities between users to specific sets of items. This fixes the issue detailed in the previous paragraph by allowing users $\mathbf{x}$ and $\mathbf{z}$ to be similar or opposite in the same latent latent space relevant only to item $\mathbf{y}$.

The objective component for social co-preference regularization along with its expanded form is

$$Obj_{rsc} = \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in C} \frac{1}{2} (P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \langle U\mathbf{x}, U\mathbf{z} \rangle_{V\mathbf{y}})^2 \qquad (17)$$

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

We might also define a co-preference spectral regularization approach, but our experiments have indicated this does not work as well as the non-spectral objective above, so we omit it here.

4. EVALUATION METHODOLOGY

In this chapter we first discuss our Facebook Link Recommendation (LinkR) application and then proceed to discuss how it can be evaluated using general principles of evaluation used in the machine learning and information retrieval fields.

4.1 Facebook

Facebook is a social networking service that is currently the largest in the world. As of July 2011 it had more that 750 million active users. Users in Facebook create a profile and establish "friend" connections between users to establish their social network. Each user has a "Wall" where they and their friends can make posts to. These posts can be links, photos, status updates, etc. Items that have been posted by a user can be "liked", shared, or commented upon by other users. An example of a link post on a Wall that had been liked by others was provided previously in Figure **??**.

This thesis seeks to find out how best to recommend links to individual users such that there is a high likelihood that they will "like" their recommended links. We do this by creating a Facebook application (i.e., '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. We discuss this application in detail next.

4.1.1 LinkR

Facebook allows applications to be developed that can be installed by their users. As part of this thesis project, the LinkR Facebook application was developed. The functionalities of the LinkR application are as follows:

1. Collect data that have been shared by users and their friends on Facebook.

2. Recommend (three) links to the users daily.

3. Collect feedback from the users on whether they liked or disliked the recommendations.

Figure 1 shows the Facebook LinkR App as it appears to users.

4.2 Dataset

Using the LinkR Facebook App developed for this project, we were able to gather data on 34,860 users and 437,023 links.

Figure 1: The Facebook LinkR App showing one of the recommendations as it appears to users of the system. Users have the option of liking or disliking a recommendation as well as providing explicit feedback commentary.

4.2.1 User Data

Date that are collected and used for the user features are as follows:

- Gender: male or female

- Birthday: year

- $location_{id}$: an integer ID corresponding to the user's specific present location (city and country)

- $hometown_{id}$: an integer ID corresponding to the user's specific home town (city and country)

- $F_{\mathbf{x},\mathbf{z}} \in \{0,1\}$: indicator of whether users $\mathbf{x}$ and $\mathbf{z}$ are friends.

- $Int_{\mathbf{x},\mathbf{z}} \in \mathbb{N}$: interactions on Facebook between users $\mathbf{x}$ and $\mathbf{z}$ as defined in Section **??**.

4.2.2 Link Data

Data that are used for the link features are:

- id of the user who posted the link.

- id of the user on whose wall the link was posted.

- Text description of the link from the user who posted it.

- Text link summary from the metatags on the target link webpage.

- Number of times the link has been liked.

- Number of times the link has been shared.

- Number of comments posted on the link.

- $F'_{\mathbf{x},\mathbf{y}} \in \{0,1\}$: indicator of whether user $\mathbf{x}$ has liked item $\mathbf{y}$.

Additionally, links that have been recommended by the LinkR application have the following extra features:

- id's of users who have clicked on the link url.

- Optional "Like" or "Dislike" rating of the LinkR user on the link.

4.2.3 Interaction Data

The interactions between users that we count (equally weighted) to define $Int_{\mathbf{x},\mathbf{z}}$ are:

1. Being friends.

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

3. Liking an item (link, photo, video, photo, or message) on a user's wall.

4. Commenting on an item (link, photo, video, photo, or message) on a user's wall.

5. Being tagged together in the same photo.

6. Being tagged together in the same video.

7. Two users tagging themselves as attending the same school.

8. Two users tagging themselves as attending the same class in school.

9. Two users tagging themselves as playing sports together.

10. Two users tagging themselves as working together for the same company.

11. Two users tagging themselves as working together on the same project for the same company.

4.2.4 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 liked. We use only the links from the last two weeks since an informal user study has indicated a preference for recent links. Furthermore, older links have a greater chance of being outdated and are also likely to represent broken links that are not working anymore. We have settled on recommending three links per day to the LinkR users and according to the survey done at the end of the first trial, three links per day seems to be the generally preferred number of daily recommendations.

For the live trials, Facebook users who installed the LinkR application were *randomly assigned one of four algorithms in each of the two trials.* Users were not informed which algorithm was assigned to them to remove any bias. We distinguish our recommended links into two major classes, links that were posted by the LinkR user's friends and links that were posted by users other than the LinkR user's friends.

Algorithm	Users
Social Matchbox	26
Matchbox	26
SVM	28
Nearest Neighbor	28

Table 1: Number of Users Assigned per Algorithm.

The LinkR users were encouraged to rate the links that were recommended to them, and even provide feedback comments on the specific links. 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. Based on the user feedback, we filtered out non-English links and links without any descriptions from the recommendations to prevent user annoyance.

At the end of the first trial, we conducted a user survey with the LinkR users to find out how satisfied they were with the recommendations they were getting.

5. EMPIRICAL RESULTS

In this chapter we discuss the first set of four SCF algorithms that was implemented for the LinkR application and then show how each algorithm performed during the live user trial, how satisfied the users were with links being recommended to them through LinkR, and the results of offline passive experiments with the algorithms.

All code available. `http://code.google.com/p/social-recommendation/` Research received ethics approval from the ANU, but we are prohibited from attempting to release the present data (in raw or anonymized form).

5.1 First Trial

The CF and SCF algorithms used for the first user trial were:

1. *k*-**Nearest Neighbor (KNN)**: We use the user-based approach as described in Section 2.3.1.

2. **Support Vector Machines (SVM)**: We use the the SVM implementation described in Section **??** using the features described in Section 4.2.

3. **Matchbox (Mbox)**: Matchbox MF + L2 U Regularization + L2 V Regularization

4. **Social Matchbox (Soc. Mbox)**: Matchbox MF + Social Regularization + L2 Regularization

Social Matchbox uses the Social Regularization method to incorporate the social information of the FB data. SVM incorporates social information in the $\mathbf{f_{x,y}}$ features that it uses. Matchbox and Nearest Neighbors do not make use of any social information.

The first live user trial was run from August 25 to October 13. The algorithms were randomly distributed among the 106 users who installed the LinkR application. The distribution of the algorithms to the users are show in Table 1

Each user was recommended three links everyday and they were able to rate the links on whether they 'Liked' or 'Disliked' it. Results shown in Figure 2 are the percentage of Like ratings and the percentage of Dislike ratings per algorithm stacked on top of each other with the Like ratings on top.

5.1.1 Online Results

As shown in Figure 2, Social Matchbox was the best performing algorithm in the first trial and in fact was the only algorithm to get receive more like ratings than dislike ratings. This would suggest that using social information does indeed provide useful information that resulted in better link recommendations from LinkR.

We also look at the algorithms with the results split between friend links and non-friend links recommendations. Again, the results shown in Figure **??** are the percentage of Like ratings and the percentage of Dislike ratings per algorithm stacked on top of each other with the Like ratings on top. As shown in Figure **??**, all four algorithms experienced a significant performance drop in the ratio of Likes to Dislikes when it came to recommending non-friend links. This suggests that aside from Liking or Disliking a link solely from the quality of the link being recommended, users are also more likely to Like a link simply because a friend had posted it and more likely to Dislike it because it was posted by a stranger.

5.1.2 Summary

At the end of the first trial, we have observed the following:

- Social Matchbox was the best performing algorithm in the live user trial in terms of percentage of likes.

- Social Matchbox received the highest user evaluation scores in the user survey at the end of the user trial.

- Of the various combinations of training and testing data in the offline passive experiment, we found that training on the UNION subset and testing on the APP-USER-ACTIVE-ALL subset best correlated with the results of the live user trial and the user survey. Training on the UNION dataset had advantages compared to training on the other data subsets, namely that it had the large amount of information of the PASSIVE data and the explicit dislikes information of the ACTIVE data.

5.2 Second Trial

For the second online trial, we chose four algorithms again to randomly split between the LinkR application users. Social Matchbox was included again as a baseline since it was the best performing algorithm in the first trial. The distribution count of the algorithms to the users is shown in Table 2

The four SCF algorithms are:

- **Social Matchbox (Soc. Mbox)** : Matchbox MF + Social Regularization + L2 U Regularization + L2 V Regularization

- **Spectral Matchbox (Spec. Mbox)**: Matchbox MF + Social Spectral Regularization + L2 U Regularization + L2 V Regularization

- **Social Hybrid (Soc. Hybrid)**: Hybrid + Social Regularization + L2 U Regularization + L2 V Regularization + L2 $\mathbf{w}$ Regularization

- **Spectral Co-preference (Spec. CP)**: Matchbox MF + Social Co-preference Spectral Regularization + L2 U Regularization + L2 V Regularization
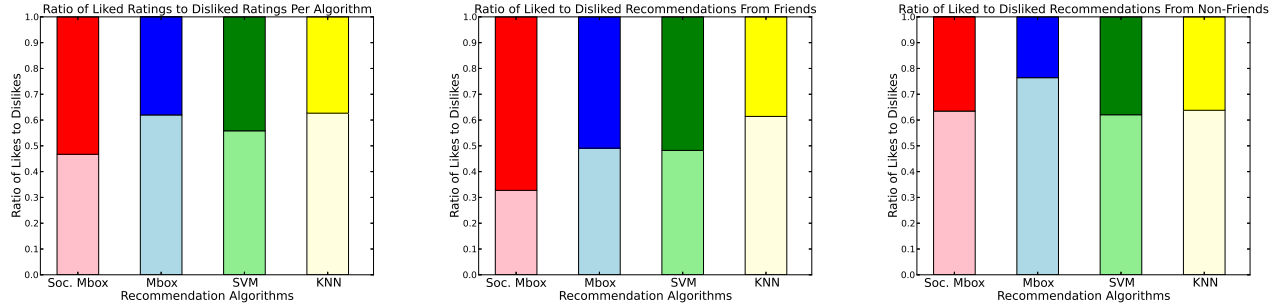
Figure 2: Results of the online live trials. The percentage of Liked ratings are stacked on top of the percentage of Disliked ratings per algorithm. Social Matchbox was found to be the best performing of the four algorithms evaluated in the first trial. *** Results of the online live trials, split between friends and non-friends. The percentage of Liked ratings are stacked on top of the percentage of Disliked ratings per algorithm. There is a significant drop in performance between recommending friend links and recommending non-friend links.

Algorithm	Users
Social Matchbox	26
Spectral Matchbox	25
Spectral Co-preference	27
Social Hybrid	25

Table 2: Number of Users Assigned per Algorithm.

5.2.1 Online Results

The online experiments were switched to the new algorithms on October 13, 2011. For the online results reported here, since the second live trial is still currently ongoing, we took a snapshot of the data as it was on October 22, 2011. The algorithms were randomly distributed among the 103 users who still had the LinkR application installed. The distribution of the algorithms to the users are show in Table 2

Results shown in Figure 3 are the percentage of Like ratings and the percentage of Dislike ratings per algorithm stacked on top of each other with the Like ratings on top.

First thing we note in Figure 3 is the decrease in performance for Social Matchbox, and in fact for all SCF algorithms in general. Except for Spectral Matchbox, they all received more Dislike ratings than Like ratings. What we noticed is that of the recommendations being made in the week that we switched over to the new algorithms, the majority of the links were about Steve Jobs, who had died the week previously. We believe that the redundancy and lack of variety of the links being recommended caused an increase in the Dislike ratings being given by users on the recommended links. Taking out the skewed results that follows an unusual event such as this, the relative algorithm performance was better.

We again split the results again between friend link recommendations and non-friend link recommendations, with the results shown in Figure 3 being the percentage of Like ratings and the percentage of Dislike ratings per algorithm stacked on top of each other with the Like ratings on top. As shown Figure 3, all four algorithms experienced significant performance drop in the number of likes when it came to recommending non-friend links. This reflects the results of the first trial.

We note the following observations from the results shown in Figures 3:

- Compared to the other algorithms, Spectral Matchbox achieved the best ratio of likes to dislikes as seen, as seen in Figure 3. Combined with the results for Spectral Co-preference, Spectral social regularization in general appears to be a better way to socially regularize compared to social regularization. This comparison holds even when the results are split between friend links recommendations and non-friend links recommendations, as seen in Figure 3.

- When looking at just the friend link recommendations in Figure 3, Social Hybrid was the best performing algorithm. This result comes from the user-user information diffusion among its friends that Social Hybrid learns, which could not be learned by the other SCF algorithms. Learning information diffusion thus helps when it comes to building better SCF algorithms.

- Spectral Co-preference didn't do well on friend link recommendations, however it did better on the non-friend link recommendations. When it comes to recommending friend links, friend interaction information coming through social regularizer seems more important than implicit co-likes information provided by the co-preference regularizer. When there is no social interaction information such as with non-friend links, co-preference methods with its implicit co-likes information appear much better than just vanilla collaborative filtering at projecting users into subspaces of common interest.

5.2.2 Summary

We summarize the observations made during the second trial:

- The social spectral regularization methods generally performed better in the live user trials, even when the results were split between friend link recommendations and non-friend link recommendations.

- Learning information diffusion models helps in SCF, as evidenced by the strong performance of Social Hybrid when recommending friend links.
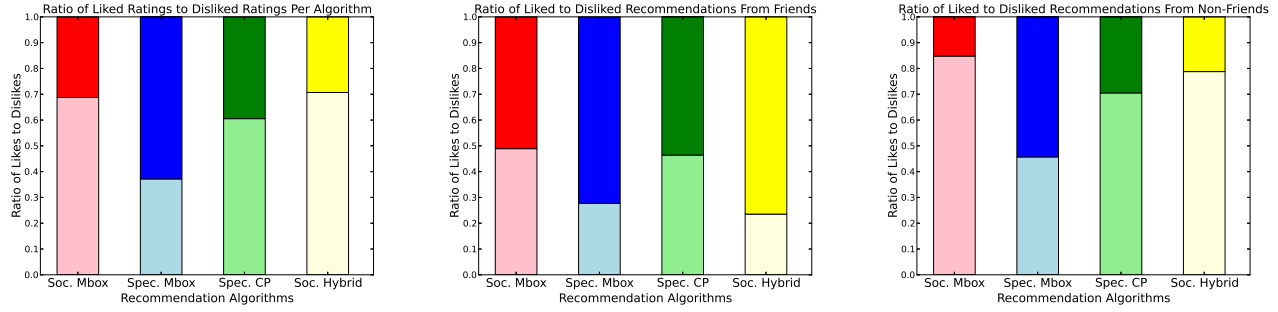
Figure 3: Results of online live trials. The percentage of Liked ratings are stacked on top of the percentage of Disliked ratings per algorithm. Spectral Matchbox achieved the highest ratio of likes to dislikes among the four algorithms. Spectral social regularization in general appears to be a better way to socially regularize compared to social regularization. *** Results of the online live trials, split between friends and non-friends. As in the first trial, there is a significant drop in performance between recommending friend links and recommending non-friend links.

- When there is no social interaction information, learning implicit co-likes information is better than using plain CF methods.

- The better performance of the social spectral regularization methods in the live trials were not reflected in the offline experiments. Perhaps there is a better metric than MAP that correlates with human preferences.

5.3 Algorithm Independent Observations

5.3.1 Click evidence

5.3.2 Impact of Popularity

5.3.3 Impact of Interactions

5.3.4 Impact of Steve Jobs

5.3.5 Impact of Genre

6. CONCLUSIONS

6.1 Summary

In this paper, we evaluated existing algorithms and proposed new algorithms for social collaborative filtering via the task of link recommendation on Facebook.

In Chapter 2, we outlined three main deficiencies in current social collaborative filtering (SCF) techniques and proposed new techniques in Chapters 4 and 5 to solve them; we review them here as follows:

(a) **Non-feature-based user similarity:** We extended existing social regularization and *social spectral regularization* methods to incorporate *user features* to learn user-user similarities in the latent space.

(b) **Model direct user-user information diffusion:** We defined a new hybrid SCF method where we *combined* the *collaborative filtering (CF) matrix factorization (MF) objective* used by Matchbox [16] with a *linear content-based filtering (CBF) objective* used to model direct user-user information diffusion in the social network.

(c) **Restricted common interests:** We defined 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.

Having evaluated existing baselines (with minor extensions) in Chapter 4 and evaluating these new algorithms in Chapter 5 in live online user trials with over 100 Facebook App users and data for over 30,000 unique Facebook users, we summarize the main results of the paper:

- Our extensions to social regularization in Chapter 4 and Chapter 5 proved to be very effective methods for SCF, and outperformed all other algorithms in the first Facebook user evaluation trial. This was reflected in the live user trial, user survey, and offline passive experimental results. However, our socially regularized SCF MF extension in Chapter 4 could still be improved by changing to a spectral approach, and this further extension in Chapter 5 generally outperformed the SCF socially regularized extension of Chapter 4. The take-home point is that a very useful form of social regularization appears to be *social spectral regularization* as proposed in Chapter 5.

- Learning direct user-user information diffusion models (i.e., how much one user likes links posted by another) can result in improved SCF algorithms in comparison to standard MF methods. To this end, our *social hybrid* algorithm which uses this information outperformed all other SCF methods when recommending *friend links* as shown in Chapter 5.

- Friend interaction information coming from social regularization is more useful than the implicit co-likes information of co-preference regularization. However, when there is no social interaction information available (as in the case of recommending *non-friend links*), learning this implicit co-likes information outperforms plain CF methods as evidenced by the relative success of the *co-preference regularization* algorithm as demonstrated in Chapter 5.

- Knowing what offline metrics for measuring SCF performance correlate with human preferences from live trials is crucial for efficient evaluation of SCF algorithms and also quite useful for offline algorithm tuning. In general, as most strongly evidenced in Chapter 4, it appears that evaluating *mean average precision* (MAP) when training on data that includes both passively inferred dislikes and explicit negative preference feedback (i.e., explicit dislikes indicated via the Facebook App), but evaluating MAP only with explicit feedback (no inferred dislikes) seems to be a train/test evaluation approach and evaluation metric that correlates with human survey feedback.

Hence, this paper has made a number of substantial contributions to SCF recommendation systems and has helped advance methods for SCF system evaluation.

6.2 Future Work

This work just represents the tip of the iceberg in different improvements that SCF can make over more traditional non-social CF methods. Here we identify a number of additional future extensions that can potentially further improve the proposed algorithms in this paper:

- We used only a subset of the possible feature data in Facebook and in the links themselves. Extending the social recommenders to handle more of the rich information that is available may result in better performance for the SCF algorithms. One critical feature that would have been useful is including a *genre* feature in the links (e.g., indicating whether the link represented a blog, news, video, etc.) to provide a fine-grained model of which types of links that users prefers to receive. This additional information would have likely prevented a number of observed dislikes from users regarding specific genres of links, e.g., those that do not listen to music much and hence do not care about links to music videos — even if these links are otherwise liked by friends and very popular.

- Enforcing diversity in the recommended links would prevent redundant links about the same topic being recommended again and again. This is especially useful when an unusual event happens like the death of Steve Jobs and the ensuing massive amount of Steve Jobs related links that flooded Facebook. While users may like to see a few links on the topic, their interest in similar links decreases over time and diversity in recommendations could help address this saturation effect.

- Another future direction this work can go to is to incorporate active learning in the algorithms. This would ensure that the SCF algorithm did not exploit the learned preferences too much and made an active effort to discover better link preferences that are available.

- Probably the biggest assumption we have made in our implementations is how we inferred the implicit dislikes of users in the Facebook data. A better and method of inferring implicit dislikes will give a big boost to the SCF algorithms. As evidenced by the results of training on ACTIVE and UNION data, having a more accurate list of likes and dislikes greatly improves the performance of the SCF algorithms.

- Finally, there may be a better metric than MAP for offline evaluation that more accurately correlates with live human preference and research should continue to evaluate a variety of existing SCF evaluation metrics in order to identify what offline evaluation metrics correlate with human judgments of algorithm performance.

While there are many exciting extensions of this work possible as outlined above, this paper represents a critical 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 to achieve state-of-the-art link recommendation.

7. ADDITIONAL AUTHORS

Additional authors: Ehsan Abbasnejad (ANU & NICTA; Canberra, AU; email: `first.last@nicta.com.au`).

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

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

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

[12] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems NIPS 14*, 2001.

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

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

[15] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

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

[17] 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.   DERIVATIVES

A.1   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:

$$\frac{\partial}{\partial \mathbf{a}} Obj = \frac{\partial}{\partial \mathbf{a}} \sum_i \lambda_i Obj_i = \sum_i \lambda_i \frac{\partial}{\partial \mathbf{a}} Obj_i \qquad (18)$$

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

$$\frac{\partial}{\partial \mathbf{a}} \sigma(o[\cdot]) = \sigma(o[\cdot])(1 - \sigma(o[\cdot])) \frac{\partial}{\partial \mathbf{a}} o[\cdot]. \qquad (19)$$

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.

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

$$\mathbf{s} = U\mathbf{x} \qquad \mathbf{s}_k = (U\mathbf{x})_k; \ k = 1 \ldots K$$
$$\mathbf{t} = V\mathbf{y} \qquad \mathbf{t}_k = (V\mathbf{y})_k; \ k = 1 \ldots K$$
$$\mathbf{r} = U\mathbf{z} \qquad \mathbf{r}_k = (U\mathbf{z})_k; \ k = 1 \ldots K$$

All matrix derivatives used for the objectives below can be verified in [13].

A.2   Derivatives for SCF Objectives

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

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

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

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

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

We can state the identical case substituting V for U.

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

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

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

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

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

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

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