New Social Collaborative Filtering Algorithms for Link Recommendation on Facebook

Anonymous
Unknown

ABSTRACT

This paper examines the problem of designing efficient, scalable, and accurate social *collaborative filtering* (CF) algorithms for personalized link recommendation on Facebook. Unlike standard CF algorithms using relatively simple user and item features (possibly just the user ID and link ID), link recommendation on social networks like Facebook poses the more complex problem of learning user preferences from a rich and complex set of user profile and interaction information. Most existing *social CF* (SCF) methods have extended traditional CF *matrix factorization* (MF) approaches, but have overlooked important aspects specific to the social setting; specifically, existing SCF MF methods (a) do not permit the use of item or link features in learning user similarity based on observed interactions, (b) do not permit directly modeling user-user information diffusion according to the social graph structure, and (c) cannot learn that that two users may only have overlapping interests in specific areas. This paper proposes a unified SCF optimization framework that addresses (a)–(c) and compares these novel algorithms with a variety of existing baslines. Evaluation is carried out via live user trials in a custom-developed Facebook App involving data collected over three months from over 100 App users and their nearly 30,000 friends. Not only do we show that our novel proposals to address (a)–(c) outperform existing approaches, but we also identify which offline ranking and classigication evaluation metrics correlate most with human judgment of algorithm performance. Overall, this paper represents a critical step forward in extending SCF recommendation algorithms to fully exploit the rich content and structure of social networks like Facebook.

Categories and Subject Descriptors

H.3.3 [**AREA**]: SUBAREA

General Terms

TERM

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) aim to address this problem by leveraging the preferences of a user population under the assumption that similar users will have similar preferences. These principles underlie the recommendation algorithms powering websites like Amazon and Netflix.[1]

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, educational and work history, preferences, etc.) and a rich set of user interactions with their friends (direction comments/posts, clicks of like, 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 to fully exploit the social network content in recommendation algorithms?

1.1 Objectives

This paper examines the problem of designing efficient, scalable, and accurate *social CF* (SCF) algorithms for *personalized link recommendation on Facebook* – quite simply the task of recommending personalized links to users that might interest them. [**PROVIDE FACEBOOK PICTURE AND BRIEF DISCUSSION OF PICTURE HERE. DISCUSS MODES OF INTERACTION – POST / LIKE / CLICK.**] *User interest* can be determined via many methods including *indirect feedback* in the

[1]On Amazon, this is directly evident with statements displayed of the form "users who looked at item X ended up purchasing item Y 90% of the time". While the exact inner workings of Netflix are not published, the best performing recommendation algorithm in the popular Netflix prize competition [**CITATION NEEDED**] used an ensemble of CF methods.

form of link clicks and *direct feedback* in the form of explicit link ratings or other evidence that a user *liked* a link (e.g., explicitly clicking "like").

Many existing SCF approaches extend *matrix factorization* (MF) techniques for CF [9] and have proved quite powerful in their ability to accurately model user preferences even when only a unique ID is available for both the user and item being recommended. The power of such methods stems from their ability to project users and items into latent vector spaces of reduced dimensionality where they are effectively grouped by similarity. Indeed, we will show in Section 5 that existing social extensions of MF are quite powerful and outperform a variety of other commonly used SCF approaches.

Given the strong performance of existing MF approaches to SCF, we aim to comparatively evaluate them and further improve on their performance where possible. To do this, we first identify a number of problems of existing SCF MF methods that we make our objective to address in this paper:

(a) **Non-feature-based user similarity:** Existing SCF MF methods do not permit the use of item or link features in learning user similarity based on observed interactions.

(b) **Model direct user-user information diffusion:** Existing SCF MF methods do not permit directly modeling user-user information diffusion according to the social graph structure.

(c) **Restricted common interests:** Existing SCF MF methods cannot learn that that two users may only have overlapping interests in specific areas.

This paper addresses all of these problems with novel contributions in an efficient, scalable, and unified latent factorization component framework for SCF. We present results of our algorithms on live trials in a custom-developed Facebook App involving data collected over three months from over 100 App users and their nearly 30,000 friends. These results show that a number of extensions proposed to resolve (a)–(c) outperform all previously existing algorithms.

In addition, given that live online user evaluation trials are time-consuming, requiring many users and often an evaluation period of at least one month, we have one last important objective to address in this paper:

(d) **Idenfying passive evaluation paradigms that correlate with actively elicited human judgments.** The benefits of doing this are many-fold. When designing new SCF algorithms, there are myriad design choices to be made, for which actual performance evaluation is the only way to validate the correct choice. Furthermore, simple parameter tuning is crucial for best performance and SCF algorithms are often highly sensitive to well-tuned parameters. Thus for the purpose of algorithm design and tuning, it is crucial to have methods and metrics that can be evaluated immediately on passive data (i.e., a passive data set of user likes) that are shown to correlate with human judgments in order to avoid the time-consuming process of evaluating the algorithms in live human trials.

Next we outline our specific paper contributions to address the above problem objectives in-depth.

1.2 Contributions

In the preceding section, we outlined three deficiencies of existing MF approaches for SCF. Now we discuss our specific contributions in this paper to address these three deficiencies:

(a) **User-feature social regularization:** One can encode prior knowledge into the learning process using a technique known as *regularization*. In the case of social MF, we often want to regularize the learned latent representations of users to enforce that users who interact heavily often have similar preferences, and hence similar latent representations.

Thus to address the deficiency noted in *non-feature-based user similarity*, we build on ideas used in Matchbox [10] to incorporate user features into the social regularization objective for SCF. There are two commonly used methods for social regularization in SCF — in Section 6 we extend both to handle user features and determine that the *spectral* regularization extension performs best.

(b) **Hybrid social collaborative filtering:** While MF methods prove to be excellent at projecting user and items into latent spaces, they suffer from the caveat that they cannot model joint features over user and items (they can only work with independent user features and independent item features). This is problematic when it comes to the issue of *modeling direct user-user information diffusion* — in short, the task of learning how often information flows from one specific user to another specific user.

The remedy for this turns out to be quite simple — we need only introduce an objective component in addition to the standard MF objective that serves as a simple linear regressor for such information diffusion observations. Because the resulting objective is a combination of latent MF and linear regression objectives, we refer to it simply as *hybrid SCF*. In Section 6, we evaluate this approach and show that it outperforms standard SCF.

(c) **Copreference regularization:** Existing SCF methods that employ social regularization make a somewhat coarse assumption that if two users interact heavily (or even worse, are simply friends) that their latent representations must match as closely as possible. Considering that friends have different reasons for their friendships — co-worders, schoolmates, common hobby — it is reasonable to expect that two people (friends or not) may only share *restricted common interests*: coworkers may both enjoy technical content related to work, but differ otherwise; schoolmates may like to hear news about other schoolmates, but differ otherwise; people who share an interest in a common hobby are obviously interested in that hobby, but should not necessarily share common interests elsewhere.

To this end, we propose a finer-grained approach to regularizing users by restricting their latent user representation to be similar (or different) only in subspaces relevant to the items mutually liked/disliked (or disagreed upon – one user likes and the other dislikes). Because this method of regularization requires evidence of preferences between two users for the same item, we refer to it as regularizing based on *copreferences*. In Section 6, we evaluate this extension to standard SCF and show that it improves performance.

The previous contributions all relate to algorithmic and machine learning aspects of SCF algorithms. However, in a different dimension and as discussed in the previous section, we also have to know how to *evaluate* these algorithms both from active user feedback (ratings of new recommendations) and passive user content (simply a catologue of previously rated links for a user). Thus as our final contribution, we perform the following extensive comparative evaluation:

(d) **Comparative evaluation of active and passive metrics that align with user judgments:** In Section 4, we propose a number of training and testing regimes and a number of evaluation metrics for both ranking and classification paradigms. In both Sections 5 and 6 we compare the performance of these metrics with the given algorithms and raw data in order to determine which regimes and metrics correlate closely with human judgment of performance in each setting.

1.3 Outline

The remaining chapters in this paper are organized as follows:

- **Section 2:** We first define notation used throughout the paper and then proceed to review both standard collaborative filtering approaches, specific MF approaches, and their social extensions.
- **Section 3:** We discuss the engineering and UI considerations behind the Facebook App that was developed for this paper.
- **Section 4:** We discuss our evaluation methodology for both offline and online (live user trial) experimentation. Our goal here is to evaluate a variety of performance objectives, both qualitative and quantitative, in order to evaluate the user experience with each recommendation algorithm and to determine which online evaluations correlate with which offline evaluations.
- **Section 5:** We empirically investigate existing SCF methods in our Facebook App and evaluation framework. Our objective here is to carry out a fair comparison and understand the settings in which each algorithm works — and most importantly for research progress — where these algorithms can be improved.
- **Section 6:** We begin by discussing novel algorithms that we propose along the lines of our contributions outlined in Section 1.2. Then proceed to evaluate them in our Facebook App and evaluation framework to understand whether these improve over the baselines, as well as to understand if there are any obvious deficiencies in the new approaches.
- **Section 7:** We summarize our conclusions from this work and outline directions for future research.

All combined, 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.

2. BACKGROUND

2.1 Social Collaborative Filtering

[When SCF is introduced, insert discussion below... note that change of notation from methods introduced later... these do not use user and item features.]

There are essentially two general classes of MF methods applied to SCF that we discuss below. The first class 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 i and k based on evidence $S_{i,k}$ of interaction between i and k. We call these methods:

- **Social regularization [11, 2]** (Obj_{rs}):

$$\sum_i \sum_{k \in friends(i)} \frac{1}{2}(S_{i,k} - \langle U_i, U_k \rangle)^2$$

- **Social spectral regularization [7, 4]** (Obj_{rss}):

$$\sum_i \sum_{k \in friends(i)} \frac{1}{2} S^+_{i,k} \|U_i - U_k\|_2^2$$

The *SoRec* system [6] proposes a slight twist on social spectral regularization in that it learns a third interactions matrix Z (same dimensionality as U and V) and uses $U_i^T Z_k$ to predict user-user interaction preferences in the same way that standard CF uses $U_i^T V_j$ to predict ratings. It also uses a sigmoidal transform on the predictions.

- **SoRec regularization [6]** (Obj_{rss}):

$$\sum_i \sum_{k \in friends(i)} \frac{1}{2}(S_{i,k} - \sigma(\langle U_i, Z_k \rangle))^2$$

The second class of SCF MF approaches represented by the *Social Trust Ensemble* can be termed as a *weighted average* approach since it simply composes a prediction for item j from a weighted average of a user i's predictions and their friends (k) predictions.

- **Social Trust Ensemble [5] (Non-spectral)** (Obj_{pmcf}):

$$\sum_{(i,j) \in D} \frac{1}{2}(R_{i,j} - \sigma(U_i^T V_j + \sum_k U_i^T V_k))^2$$

2.2 Tensor Factorization Methods

Tensor factorization (TF) methods can be used to learn latent models of interaction of 2 dimensions and higher. A dimension 2 TF method is simply standard MF. An example of a dimension 3 TF method is given by [8] where recommendation of user-specific tags for an item are modeled with tags, user, and items each in one dimension. To date, TF methods have not been used for social recommendation, however, we draw on the idea of addition dimensions of latent learning in our copreference regularization method.

3. FACEBOOK APP ENGINEERING

4. EVALUATION METHODOLOGY

5. BASELINE COMPARISION

6. NOVEL ALGORITHMS FOR SOCIAL RECOMMENDATION

7. CONCLUSIONS

8. BACKGROUND

We define collaborative filtering (CF) as the task of predicting whether a user will like (or dislike) an item by using that user's preferences as well as those of other users. CF can be done with or without explicit user or item features as in [10], hence subsuming traditional content-based filtering (CBF) according to our definition.

We loosely define social CF (SCF) as the task of CF augmented with additional social network information such as the following that are available on social networking sites such as Facebook:

- 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 and links).
- Content of interactions between users (often text and links).
- Evidence of other interactions between users (being tagged in photos).
- Publicly available preferences (likes/dislikes of posts and links).
- Publicly available 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, nonetheless we refer to any CF method that *can be applied* in a social setting *as* SCF.

8.1 Notation

In this work, we outline a number of potential SCF optimization objectives. First, however, we must outline mathematical notation common to the SCF setting and models explored in this work:

- N users, each having an I-element feature vector $\mathbf{x} \in \mathbb{R}^I$ (alternately if a second user is needed, $\mathbf{z} \in \mathbb{R}^I$).
- M items, each having a J-element feature vector $\mathbf{y} \in \mathbb{R}^J$.
- A (non-exhaustive) data set D of user preferences of the form $D = \{(\mathbf{x}, \mathbf{y}) \to R_{\mathbf{x},\mathbf{y}}\}$ where class $R_{\mathbf{x},\mathbf{y}} \in \{0 \text{ (dislike)}, 1 \text{ (like)}\}$.
- A (non-exhaustive) data set C of co-preferences derived from D of the form $C = \{(\mathbf{x}, \mathbf{z}, \mathbf{y}) \to P_{\mathbf{x},\mathbf{z},\mathbf{y}}\}$ where class $P_{\mathbf{x},\mathbf{z},\mathbf{y}} \in \{-1 \text{ (disagree)}, 1 \text{ (agree)}\}$.

Note that feature vectors for users and items can consist of any real-valued features as well as $\{0, 1\}$ features like user and item IDs.

Most traditional CBF methods learn in an explicit feature space, while most traditional CF methods learn in a latent feature space (out of necessity by using only user and item ID features). Since our definition of (S)CF subsumes both, we define both explicit and implicit features:

- *Explicit*: We assume that a fixed-length feature vector $\mathbf{f} \in \mathbb{R}^F$ can be derived for any $(\mathbf{x}, \mathbf{y}) \in D$, denoted as $\mathbf{f}_{\mathbf{x},\mathbf{y}}$. In the SCF setting, $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ may include features that are non-zero only for specific items and/or users, e.g., a $\{0, 1\}$ indicator feature that user $\mathbf{x}$ and user $\mathbf{z}$ have both liked item $\mathbf{y}$. Using $\langle \cdot, \cdot \rangle$ to denote an inner product, we define a weight vector $\mathbf{w} \in \mathbb{R}^F$ such that $\langle \mathbf{w}, \mathbf{f}_{\mathbf{x},\mathbf{y}} \rangle = \mathbf{w}^T \mathbf{f}_{\mathbf{x},\mathbf{y}}$ forms a linear regressor.
- *Implicit*: As done in standard CF methods, we assume that a matrix U allows us to project users $\mathbf{x}$ (and $\mathbf{z}$) into a latent space of dimensionality K; likewise we assume that a matrix V allows us to project items $\mathbf{y}$ into a latent space also of dimensionality K. Formally we define U and V as follows:

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

  Now 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 latent bilinear regressor.

There are many ways to incorporate indirect social information into SCF methods of preference. Here we opt 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:

$$Int_{\mathbf{x},\mathbf{z}} = \frac{\#\text{ interactions between } \mathbf{x} \text{ and } \mathbf{z}}{\text{average } \#\text{ interactions between all user pairs}} \tag{1}$$

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

For purposes of this definition, an *interaction* is any single event showing evidence that users $\mathbf{x}$ and $\mathbf{z}$ have interacted, e.g., a message exchange or being tagged in a photo together.

In addition, we can define $S^+_{\mathbf{x},\mathbf{z}}$, a *non-negative* variant of $S_{\mathbf{x},\mathbf{z}}$:

$$S^+_{\mathbf{x},\mathbf{z}} = \ln\left(1 + Int_{\mathbf{x},\mathbf{z}}\right) \tag{3}$$

8.2 Social Collaborative Filtering

8.3 Objective components

We take a composable approach to collaborative filtering (CF) systems where a (social) CF minimization objective *Obj* is composed of sums of one or more objective components:

$$Obj = \sum_i \lambda_i Obj_i \tag{4}$$

Because each objective may be weighted differently, we include a weighting term $\lambda_i \in \mathbb{R}$ for each component that should be optimized via cross-validation.

We note that most target predictions are binary classification-based ($\{0,1\}$) so that in our objectives we might want to use a sigmoidal transform

$$\sigma(o) = \frac{1}{1+e^{-o}} \tag{5}$$

of regressor outputs $o \in \mathbb{R}$ to squash it to the range $[0,1]$. In places where the σ transform may be optionally included, we write $[\sigma]$.

Now we define potential primary objective components:

- **Explicit Linear CBF** (Obj_{pcbf}):

$$\sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{w}^T\mathbf{f}_{\mathbf{x},\mathbf{y}})^2 \tag{6}$$

- **Matchbox [10] CF+CBF** (Obj_{pmcf}):

$$\sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{x}^T U^T V y)^2 \tag{7}$$

- **Hybrid** (Obj_{phy}):

$$\sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{w}^T\mathbf{f}_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{x}^T U^T V y)^2 \tag{8}$$

In the above, our free parameters for learning are U, V, and $\mathbf{w}$. It is important to regularize these parameters to prevent overfitting in the presence of sparse data; for this purpose there are a variety of choices ranging from the well-known L_2 regularizer that models a prior of 0 on the parameters to more SCF-specific forms of regularization that constrain rows of U and V to be similar based on various observations in the SCF data:

- **L_2 w regularization** (Obj_{rw}):

$$\frac{1}{2}\|\mathbf{w}\|_2^2 = \frac{1}{2}\mathbf{w}^T\mathbf{w} \tag{9}$$

- **L_2 U regularization** (Obj_{ru}):

$$\frac{1}{2}\|U\|_{\text{Fro}}^2 = \frac{1}{2}\operatorname{tr}(U^T U) \tag{10}$$

- **L_2 V regularization** (Obj_{rv}):

$$\frac{1}{2}\|V\|_{\text{Fro}}^2 = \frac{1}{2}\operatorname{tr}(V^T V) \tag{11}$$

- **Social regularization** (Obj_{rs}):

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

- **Social spectral regularization** (Obj_{rss}):

$$\begin{aligned}&\sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} \frac{1}{2}S^+_{\mathbf{x},\mathbf{z}}\|U\mathbf{x} - U\mathbf{z}\|_2^2 \\ &= \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} \frac{1}{2}S^+_{\mathbf{x},\mathbf{z}}\|U(\mathbf{x} - \mathbf{z})\|_2^2 \\ &= \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} \frac{1}{2}S^+_{\mathbf{x},\mathbf{z}}(\mathbf{x} - \mathbf{z})^T U^T U(\mathbf{x} - \mathbf{z}) \end{aligned}\tag{13}$$

  Note: standard spectral regularization assumes $S^+_{\mathbf{x},\mathbf{z}} \in [0,1]$; however we may also want to try $S_{\mathbf{x},\mathbf{z}}$ since a negative value actively encourages the latent spaces to oppose each other, which may be desired.

The motivation behind the next two objectives is to constrain users $\mathbf{x}$ and $\mathbf{z}$ who have similar (opposing) preferences to be similar (opposite) in the same latent latent space relevant to item $\mathbf{y}$. This captures the crucial aspect — missing from other SCF methods — that while two users may not be globally similar (opposite), there may be sub-areas of their interests where they are similar (opposite). For example, two friends may have similar interests concerning music, but different interests concerning politics. The following regularization objectives aim to learn such selective co-preferences:

- **Social co-preference regularization** (Obj_{rsc}) – this requires a reweighted inner product $\langle\cdot,\cdot\rangle_\bullet$ expanded into its definition below:

$$\begin{aligned}&\sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2}(P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \langle U\mathbf{x}, U\mathbf{z}\rangle_{V\mathbf{y}})^2 \\ &= \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2}(P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \mathbf{x}^T U^T \operatorname{diag}(V\mathbf{y})U\mathbf{z})^2 \end{aligned}\tag{14}$$

  Note 1: computationally, it could be very expensive to compute this for all pairs, we might consider ways to restrict it, e.g., only considering *App users* for $\mathbf{x}$ or only considering *friends* for $\mathbf{x}$ and $\mathbf{z}$.

  Note 2: we should also try setting $P_{\mathbf{x},\mathbf{z},\mathbf{y}} = \text{(disagree)} = 0$.

- **Social co-preference spectral regularization** (Obj_{rscs}) – this requires a re-weighted L_2 norm $\|\cdot\|_{2,\bullet}$ expanded into its definition below:

$$\begin{aligned}&\sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2}P_{\mathbf{x},\mathbf{z},\mathbf{y}}\|U\mathbf{x} - U\mathbf{z}\|_{2,V\mathbf{y}}^2 \\ &= \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2}P_{\mathbf{x},\mathbf{z},\mathbf{y}}\|U(\mathbf{x} - \mathbf{z})\|_{2,V\mathbf{y}}^2 \\ &= \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2}P_{\mathbf{x},\mathbf{z},\mathbf{y}}(\mathbf{x} - \mathbf{z})^T U^T \operatorname{diag}(V\mathbf{y})U(\mathbf{x} - \mathbf{z}) \end{aligned}\tag{15}$$

  Note: see notes 1 and 2 for the previous case that also apply here.

8.4 Derivatives

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

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

$$\begin{aligned}\frac{\partial}{\partial \mathbf{a}} Obj &= \frac{\partial}{\partial \mathbf{a}} \sum_i \lambda_i Obj_i \\ &= \sum_i \lambda_i \frac{\partial}{\partial \mathbf{a}} Obj_i\end{aligned}$$

Previously we noted that in the objective components of Section 8.3, we may want to transform some of the regressor outputs $o[\cdot]$ using $\sigma(o[\cdot])$. This is convenient for our partial derivatives as

$$\frac{\partial}{\partial \mathbf{a}} \sigma(o[\cdot]) = \sigma(o[\cdot])(1-\sigma(o[\cdot]))\frac{\partial}{\partial \mathbf{a}} o[\cdot]. \tag{16}$$

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.[2]

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

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

Now we proceed to derivatives for the previously defined primary objective components:

- **Explicit Linear CBF** (Obj_{pcbf}):

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

- **Matchbox [10] CF+CBF** (Obj_{pmcf}): Here we define alternating partial derivatives between U and V, holding one constant and taking the derivative w.r.t. the other:[3]

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

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

We note that these derivatives use outer products $\mathbf{t}\mathbf{x}^T$ and $\mathbf{s}\mathbf{y}^T$.

- **Hybrid** (Obj_{phy}):

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

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

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

Now we proceed to derivatives for the previously defined regularization objectives:

[2] We note that our experiments using the sigmoidal transform in objectives with $[0,1]$ predictions do not generally demonstrate a clear advantage vs. the omission of this transform as originally written (although they do not demonstrate a clear disadvantage either).

[3] We will use this method of alternation for all objective components that involve bilinear terms.

- **L_2 w regularization** (Obj_{rw}):

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}} Obj_{rw} &= \frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} \\ &= \mathbf{w}\end{aligned}$$

- **L_2 U regularization** (Obj_{ru}):

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

- **L_2 V regularization** (Obj_{rv}):

$$\begin{aligned}\frac{\partial}{\partial V} Obj_{rv} &= \frac{\partial}{\partial V} \frac{1}{2} \operatorname{tr}(V^T V) \\ &= V\end{aligned}$$

- **Social regularization** (Obj_{rs}):

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

- **Social spectral regularization** (Obj_{rss}):

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

Before we proceed to the final derivatives, we define one additional vector abbreviation:

$$\mathbf{r} = U\mathbf{z} \qquad \mathbf{r}_k = (U\mathbf{z})_k;\ k = 1 \ldots K.$$

- **Social co-preference regularization** (Obj_{rsc}):

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

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

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

- **Social co-preference spectral regularization** (Obj_{rscs}):

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

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

Hence, for any choice of primary objective and one or more regularizers, we simply add the derivatives for each of $\mathbf{w}$, U, and V (if present) according to (16).

8.5 Algorithms

Here we outline simple baseline algorithms evaluated:

- *GP*: Most globally popular links – user-independent
- *FLL*: Most liked links among user friends – user-centric (FLL)
- *FUW*: Friend uniform weighting – sample links posted by friends, weighting friends uniformly
- *FIW*: Friend interaction weighting – sample links posted by friends, weighting friends according to number of interactions
- *NN*: Nearest neighbor – similar to Bell and Koren's Netflix work

Here we outline the SCF learning algorithms evaluated in the first 1-month Facebook trial in terms of the primary and regularization objectives used:

- *CBF*: $Obj_{pcbf} + \lambda_{rw} Obj_{rw}$ – but trained with hinge loss (SVM) rather than L_2 loss
- *CF*: $Obj_{pcbf} + \lambda_{ru} Obj_{ru} + \lambda_{rv} Obj_{rv}$ – standard Matchbox-style CF model

- *SCF*: $Obj_{pcbf} + \lambda_{ru} Obj_{ru} + \lambda_{rv} Obj_{rv} + \lambda_{rs} Obj_{rs}$ – social CF (similar to that used in many papers)

Here we outline the SCF learning algorithms to be evaluated for inclusion in the 2nd-month Facebook trial in terms of the primary and regularization objectives used:

- *HSCF*: $Obj_{phy} + \lambda_{rw} Obj_{rw} + \lambda_{ru} Obj_{ru} + \lambda_{rv} Obj_{rv} + \lambda_{rs} Obj_{rs}$ – hybrid social CF

- *SSCF*: $Obj_{pcbf} + \lambda_{ru} Obj_{ru} + \lambda_{rv} Obj_{rv} + \lambda_{rss} Obj_{rss}$ – social spectral CF

- *SCCF*: $Obj_{pcbf} + \lambda_{ru} Obj_{ru} + \lambda_{rv} Obj_{rv} + \lambda_{rsc} Obj_{rsc}$ – social co-preference CF

- *SCCF*: $Obj_{pcbf} + \lambda_{ru} Obj_{ru} + \lambda_{rv} Obj_{rv} + \lambda_{rscs} Obj_{rscs}$

- (hybrid variants of the above only if HSCF outperforms SCF)

- (might try combining social and co-preference regularization)

In these models, the predictor for evaluation purposes is always formed from the predictor in the primary objective.

8.6 Related work

There is a massive amount of related work on SCF [10, 5, 11, 3, 8, 6, 7, 4, 1, 2] embodying some of the ideas above, however there are a few aspects covered here, not covered in this related work:

1. Existing SCF methods *cannot* capture some of the basic features that are used in standard CBF systems due to the inherent independent factorization between user and items (e.g., how much one user follows another) — this is the motivation behind the *hybrid* objectives.

2. All methods *except* for Matchbox [10] ignore the issue of user and item features. We extend the Matchbox approach above in our SCF methods.

3. *None* of the methods that propose social regularization [5, 7, 4, 11, 3, 2] incorporate user features into this regularization (as done above).

4. Tensor-based factorizations such as [8] use a full $K \times K \times K$ tensor for collaborative filtering w.r.t. tag prediction for users and items. While our co-preference regularization models above were motivated by tensor approaches, we instead take an item-reweighted approach to the standard inner products to (a) avoid introducing yet more parameters and (b) as a way to introduce additional regularization in a way that supports the standard Matchbox [10] CF model where prediction at run-time is made for a (user,item) pair, not for triples of (user,item,tag) as assumed in the tensor models.

9. EVALUATION

9.1 Train and test framework

- Data is (user, item) pairs [time must be ignored due to the fact that Facebook does not record timestamps for "likes"]

- If test data drawn from subset of train data then: randomly select x% of data for $x \in [10, 30]$ (nominally 20%) for testing – ensure that train/test (user,item) sets *do not* overlap
  else if train/test drawn from disjoint candidate sets: select all test data available

- Eventually will want to cross-validate (repeatedly train/test) but for now stderrs over user means is OK

Restrictions for training set of (user,item) pairs:

- (Active) Actively recommended LinkR like/dislike data (must limit to App users)

- (Passive) Passively liked/posted data (i.e., non-LinkR) – infer dislikes as you are currently doing (but don't use any Active LinkR info)

- (Union) Union of Active and Passive

Restrictions for testing set of (user,item) pairs:

- (FB-User-Passive) All Facebook users in data, all available passive links for data set (infer dislikes as currently doing)

- (App-User-Passive) App users only, all available passive links for data set (infer dislikes as currently doing)

- (App-User-Active-All) App users only, all available active friend & non-friend links for data set

- (App-User-Active-Friend) App users only, all available active friend links for data set

- (App-User-Active-Non-friend) App users only, all available active non-friend links for data set

Note 1: for App-User-Active-?, discard users who don't have at least one like and dislike.

Note 2: in case where training is on Active data and testing on Passive data (or vice versa), the train/test data will be drawn from disjoint candidate sets. In all other cases, it is possible to build the train/test set by splitting the same candidate set. See notes above on how to choose size of test set.

9.2 Evaluation metrics

- Ranking view: mean average precision (MAP)... result lists per user can be determined in different ways (see below).

- Binary classification view: area under the curve (AUC) on App-User-Active-?

- (might consider other ranking metrics like DCG, MRR)

Note – no need to compute for now: Recall@k, F-score@k [a recommender systems researcher pointed out to me that Recall@k (and hence F-score@k) don't make as much sense and are usually *not* cited in the literature... so let's ignore]

When determining candidate lists for MAP, there are two reasonable choices:

- (Same) List of all links available to be ranked in test set – same for all users
- (Spec) In the special case of App-User-Active-?, can build a specialized list of links per *App* user... just rank their *explicit likes/dislikes*

Thus, overall evaluation choices are a cross-product:

$$\{\text{metric}\} \times [\{\text{list candidate set}\}] \times \{\text{train}\} \times \{\text{test}\}$$

.

9.3 Evaluation configurations

It would be good to have scripts to generate any of the following results:

- $\{$AUC$\}\times\{$Passive,Active,Union$\}\times\{$App-User-Active-?$\}$
- $\{$MAP$\}\times\{$Same,Spec$\}\times\{$Passive,Active,Union$\}\times\{$App-User-Active-?$\}$
- $\{$MAP$\} \times \{$Same$\} \times \{$Passive,Active,Union$\}$ $\times\{$FB-User-Passive, App-User-Passive, App-User-Active-?$\}$

10. REFERENCES

[1] Bin Cao, Jian-Tao Sun, Jianmin Wu, Qiang Yang, and Zheng Chen. Learning bidirectional similarity for collaborative filtering. In *ECML-08*, 2008.

[2] Peng Cui, Fei Wang, Shaowei Liu, Mingdong Ou, and Shiqiang Yang. Who should share what? item-level social influence prediction for users and posts ranking. In *International ACM SIGIR Conference (SIGIR)*, 2011.

[3] Li, Yeung, and Zhang. Generalized latent factor models for social network analysis. In *IJCAI-11*, 2011.

[4] Wu-Jun Li and Dit-Yan Yeung. Relation regularized matrix factorization. In *IJCAI-09*, 2009.

[5] Ma, King, and Lyu. Learning to recommend with social trust ensemble. In *SIGIR-09*, 2009.

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

[7] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *WSDM-11*, 2011.

[8] S. Rendle, L. B. Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *KDD-09*, 2009.

[9] Ruslan Salkhunitdov and **???** Probabilistic matrix factorization. In *???*, ???

[10] David H. Stern, Ralf Herbrich, and Thore Graepel. Matchbox: large scale online bayesian recommendations. In *WWW-09*, pages 111–120, 2009.

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