New Algorithms for Social Recommendation

Joseph Christian G. Noel

A subthesis submitted in partial fulfillment of the degree of
Master of Computing (Honours) at
The Department of Computer Science
Australian National University

October 2011

Typeset in Computer Modern by TeX and LaTeX 2ε.

Except where otherwise indicated, this thesis is my own original work.

Joseph Christian G. Noel
22 October 2011

To Aurora Co

Acknowledgements

Thanks to my adviser, Scott Sanner.

Abstract

Automatic recommendations helps users see what items may be important them, and saves them the time and effort of looking for that item. The most basic recommender systems make recommendation using information about the users themselves as well as the items that are being recommended. Collaborative recommendation extends this by basing recommendations on correlations between other users with the same preferences. This paper deals with another extension to collaborative recommendation systems, that of social recommendation. Social recommendation makes use of a user's social information to improve the recommendations made to those users. Social information can come from the data in social networks like Facebook like the user's friends on the network or a log of the normalized sum of interactions between users. This paper looks at different methods of incorporating social information into recommender systems, in particular link recommendation on Facebook. Two sets of live user trial was done, using the LinkR Facebook application that was developed as part of this project Offline experiments were also performed on Facebook data that gathered by LinkR. Lastly, a user survey was conducted at the end of the first user trial and those results are also reported here. The results of the the first user trial, offline experiments, and the user survey were used to design new methods for social algorithms and a second live user trial was done with the new algorithms.

Contents

Chapter 1

Introduction

1.1 Description

Finding relevant information from the glut of data is one of the biggest challenges faced by users today. This is important not just for the users themselves, but also for companies that may wish to sell or provide a service to them. One way to help users find relevant information is through automatic recommender systems. Recommender systems seek to automatically discover what the user's preferences are.

1.1.1 Individual Recommendation

Individual recommendation models a user's preferences through information about the user alone. This information can be the user's profile details like age, sex, and occupation, as well as the user's history like previously bought or rated items.

1.1.2 Collaborative Recommendation

Collaborative recommendation models a user preferences not just through information about the user alone, but also through information about the other users. Collaborative recommendation algorithms examples are k-nearest neighbors and probabilistic matrix factorization.

1.1.3 Social Recommendation

In contrast to collaborative recommendation, which treats all users as equal for recommendation, social recommendation makes use of certain links to help calculate similarity between users. Additional information help with recommendation. It has been shown that users are more likely to have the same preference with their friends than with other random users.

These links could be connections between users in social networks like Facebook and MySpace, or some other measure of user interaction and similarity.

1.2 Facebook

Facebook is a social networking service that is currently the largest in the world. As of July 2011 it has 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.

Figure 1.1: A link posted by the author that has been liked by three other users.

This paper seeks to find out how best to recommend links to individual users such that there is a high likelihood of them "liking" it. We do this by creating a Facebook application that recommends links to users everyday and the users could give their feedback on the links, whether they liked it or disliked it.

1.3 LinkR

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

1. Collect data that have been shared by users and their friends on Facebook.
2. Recommend links to the users daily.
3. Collect feedback from the users on whether they liked or disliked the recommendations.

The main developer of LinkR is Khoi-Nguyen Tran, a PhD student at the Australian National University. The algorithms it uses at the backend for recommendation was developed as part of this paper.

1.4 Chapter Outline

In the following chapters we will show that social information can be an effective tool that can be used to improve recommender systems. In Chapter 2 I discuss the

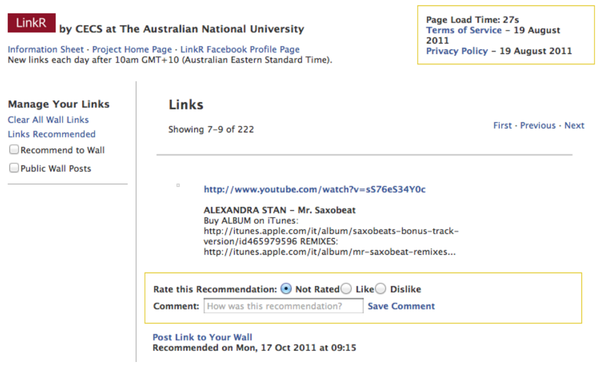

Figure 1.2: The LinkR application showing one of the recommendations.

background material regarding recommendation, the notations, and descriptions of existing algorithms.

In Chapter 3 I discuss the evaluation metrics and design issues that came up during the implementation of the LinkR application as well as during the evaluation of the different recommenders in general.

In Chapter 4 I discuss the results of the live online experiments using the first set of recommendations algorithms. The algorithms used for this trial were Nearest Neighbor, Support Vector Machines, Matchbox, and Social Matchbox. I also show the results of offline experiments using the same algorithm, and the results of a user survey conducted at the end of the live trials.

In Chapter 5 I discuss new social algorithms that build on the ones currently available in the literature. We ran a second live trial using four of these algorithms and report the results here.

In the last chapter I report on on the conclusions reached by our experiments and comment on the paths that further research in this area might take.

Chapter 2

Background

2.1 Notation

This paper uses boldface uppercase letters, like $\mathbf{U}$, to denote matrices. Boldface lowercase letters, like $\mathbf{v}$, are used to denote vectors. $\mathbf{U}_{ij}$ denotes the ith row and jth column of the matrix $\mathbf{U}$, and $\mathbf{v}_i$ denotes the ith element in the vector $\mathbf{v}$. λ and β are the regularization parameters. Additionally, our model for social recommendations needs the following:

- 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 $\mathbf{y} \in \mathbb{R}^J$. 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.
- 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 $P_{\mathbf{x},\mathbf{z},\mathbf{y}} \in \{-1 \text{ (disagree)}, 1 \text{ (agree)}\}$.
- 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:

  $$Int_{\mathbf{x},\mathbf{z}} = \frac{\#\text{ interactions between } \mathbf{x} \text{ and } \mathbf{z}}{\text{average } \#\text{ interactions between all user pairs}} \quad (2.1)$$

  $$S_{\mathbf{x},\mathbf{z}} = \ln\left(Int_{\mathbf{x},\mathbf{z}}\right) \quad (2.2)$$

  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 $S_{\mathbf{x},\mathbf{z}}$:

  $$S^+_{\mathbf{x},\mathbf{z}} = \ln\left(1 + Int_{\mathbf{x},\mathbf{z}}\right) \quad (2.3)$$

The matrix $\mathbf{R}$ is a sparse $N \times M$ matrix of user ratings on items. The problem of recommendation is filling out the empty elements of this matrix, and this can be looked at as a linear regression problem. There are two general ways that this has been done previously, Content-based Filtering (CBF) and Collaborative Filtering (CF). Content-based filtering makes recommendations based on correlations between the item features and the user's preferences on other items. In collaborative filtering, the system makes recommendations based on the correlation between other user's with similar preferences. Most traditional CBF methods learn in an explicit feature space, while most traditional CF methods learn in a latent feature space.

2.2 Collaborative Filtering Algorithms

2.2.1 K-Nearest Neighbor

The k-nearest neighbor algorithm is a method of pattern recognition that is based on the k closest training data in the feature space. There are two main variants of nearest neighbors for collaborative recommendation, user-based and item-based. Given a user u and an item i, let $N(u : i)$ be the set of nearest neighbors of u that have also given a rating for i, $N(i : u)$ be the set of nearest neighbors of i that have also been rated by u, $s_{uu'}$ the similarity rating between users u and u', and $s_{ii'}$ be the similarity rating for items i and i'. The predicted rating the user u gives item i in the user-based approach is then calculated as

$$r_{ui} = \frac{\sum_{v \in N(u;i)} s_{uv} r_{uv}}{\sum_{v \in N(u;i)} s_{uv}}$$

The item-based approach is calculated as

$$r_{ui} = \frac{\sum_{j \in N(i;u)} s_{ij} r_{uj}}{\sum_{j \in N(i;u)} s_{ij}}$$

The question of which approach to use depends on the dataset. 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.

2.2.2 Support Vector Machines

Support Vector Machines are a class of supervised learning classification algorithms that uses a hyperplane separating approach. During training, SVM builds a model by constructing a set of hyperplanes that separates one class of data from another class with the maximum margin possible. Data are classified by finding out on which side of a hyperplane they fall under.

For the experiments, SVM uses a fixed-length feature vector $\mathbf{f} \in \mathbb{R}^F$ derived from any $(\mathbf{x}, \mathbf{y}) \in D$, denoted as $\mathbf{f}_{\mathbf{x},\mathbf{y}}$. $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ 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 $\mathbf{y}$.

2.3 Matrix Factorization Models

2.3.1 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:

$$Obj = \sum_i \lambda_i Obj_i \tag{2.4}$$

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

Most target predictions are binary classification-based ($\{0,1\}$), therefore in the objectives a sigmoidal transform

$$\sigma(o) = \frac{1}{1+e^{-o}} \tag{2.5}$$

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

2.3.2 Matchbox Matrix Factorization

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:

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

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. The objective component for this model that we seek to minimize is:

$$\sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{x}^T U^T V y)^2 \tag{2.6}$$

2.3.3 L2 Regularization

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 objective components for the L2 regularizers are

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

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

2.3.4 Social Regularization

The social aspect of social recommendation is implemented as a regularizer on the user matrix. 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 have the same preferences for items.

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

2.3.5 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 that we may want to transform some of the regressor outputs $o[\cdot]$ using $\sigma(o[\cdot])$. This is convenient for our partial derivatives as

$$\frac{\partial}{\partial \mathbf{a}} \sigma(o[\cdot]) = \sigma(o[\cdot])(1 - \sigma(o[\cdot])) \frac{\partial}{\partial \mathbf{a}} o[\cdot]. \tag{2.7}$$

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.[1]

[1] 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

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

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

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

- **Matchbox Matrix Factorization**: Here we define alternating partial derivatives between U and V, holding one constant and taking the derivative w.r.t. the other:[2]

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

For the regularization objective components, the derivatives are:

- L_2 U **regularization**:

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

- L_2 V **regularization**:

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

(although they do not demonstrate a clear disadvantage either).

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

- **Social regularization**:

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

Hence, for any choice of primary objective and one or more regularizers, we simply add the derivatives for U and/or V according to (2.7).

Chapter 3

Evaluation of Social Recommendation Systems

3.1 Evaluation Metrics

We define True Positives (TP) to be the count of relevant items that were returned by the algorithm, False Positives (FP) to be the count of non-relevant items that were returned by the algorithm, True Negatives (TN) to be the count of non-relevant items that weren't returned by the algorithm, and False Negatives (FN) to be the non-relevant items that were returned by the algorithm.

Precision is a measure of what fraction of items returned by the algorithm were actually relevant.

$$Precision = \frac{TP}{TP + FP}$$

For some queries, results are returned as a ranked list. Therefore the position of an item in the list must also be evaluated, not just whether the item is in the returned list or not. A metric that does this is Average Precision, which computes the precision at every position in a ranked sequence of documents. If k is the rank in a sequence of retrieved documents, n is the number of retrieved documents, and $P(k)$ is the precision at cut-off k in the list. $rel(k)$ is an indicator function equalling 1 if the item at position k is a relevant document, and 0 otherwise. The average precision can be calculated as

$$AveP = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{\text{number of relevant documents}}$$

The main metric we use in this paper is the mean average precision (MAP) Since we make a recommendation for each user, these recommendations can be viewed as a separate query per user, and evaluate the average precision for each one. Getting the mean of all the average precisions gives us an effective metric for the entire recommendation system.

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$$

3.2 Design Choices

3.2.1 Facebook and LinkR data

Using the LinkR Facebook application developed for this project, we were able to gather data on 34,245 users and 407,887 links.[1] Data available on the users are:

- Basic user features: *gender*, *birthday*, *location*, and *hometown*.
- Mapping whether users **x** and **z** are friends.
- Interactions on Facebook between users **x** and **z**.

Data available on the links are:

- User who posted the link
- The user on whose wall the link was posted
- User's description of the link
- Link summary from the webpage
- Number of times the link has been liked
- Number of times the link has been shared
- Number of comments posted on the link
- List of users that have 'liked' the link.

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

- List of users who have clicked on the url.
- Optional "Like" or "Dislike" rating of the LinkR user on the link.

We also consider the users who posted the link to have implicitly liked it already. Outside of the "Dislike" ratings that we are able to get from the LinkR data, there is no other functionality within Facebook itself that allows users to explicitly define which link they do not like. Therefore, we need some way to infer disliked links during training. During training we consider links that were posted by the user's friends and which they have not likes as an evidence that they dislike a link. This is actually a big assumption as in a lot of cases given the nature of the Facebook news feed they may simply have not seen the link yet, and may actually like the link if they see it. Nevertheless, we find in our passive experiment and in live trial that this assumption is still useful.

[1] As of October 18, 2011, 12:15am

3.2.2 Training Data

Because of the sheer size of the Facebook data, it was impractical to run training and recommendations over the entire dataset. To keep the runtime of our experiments within reason, we used only the most recent 4 weeks of data for training the recommenders. This also helps alleviate some temporal aspects of the user's changing preferences, i.e., what the user liked last year may not be the same as what he or she likes this year. We also distinguish between the three types of link like/dislike data we can get from the dataset:

- ACTIVE: The explicit "Like" and "Dislike" rating that an application user gives on a recommended link. In addition to this, a click by a user on a recommended link also counts as a like by that user on that particular link. Only LinkR users have this data.

- PASSIVE: The like data given to us by Facebook through the Graph API, plus the inferred dislikes as detailed above.

- UNION: Combination of the ACTIVE and PASSIVE data.

3.2.3 Live Online Recommendations

For the recommendation 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 because we consider recency to be a big issue. Older links have a greater chance of being about things that are outdated already, or worse, the URL for the link may be broken and 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 just the right number.

For the live trials, Facebook users who installed the LinkR application were randomly assigned one of the 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. The reason for this is we wished to see whether the users would prefer recommended links from friends or from strangers. 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 was 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.

Figure 3.1: Screenshot a LinkR recommendation with the rating and feedback options.

3.2.4 Test Data

Our offline testing was used to tune the λ parameter values for the various regularizers and for deciding which recommendation algorithms to use in the live trials. Similar to our selection for training data, the test data used for our passive experiment also uses only the most recent 4 weeks of data. We distinguish the test data into the following classes:

- FB-USER-PASSIVE: The PASSIVE like/dislike data from all Facebook users in the dataset.
- APP-USER-PASSIVE: The PASSIVE like/dislike data from only the LinkR application users.
- APP-USER-ACTIVE-FRIENDS: The ACTIVE like/dislike data for the LinkR users, but only for friend recommended links.
- APP-USER-ACTIVE-NON-FRIENDS: The ACTIVE like/dislike data for the LinkR users, but only for non-friend recommended links.
- APP-USER-ACTIVE-ALL: The entire active like/dislike data for the LinkR users.

During passive experiments, we simply select which combination of training data and testing data to use. This helped us see which training-test data combination best reflected the results of the live trials Eventually, it was found that using UNION data for training and testing on APP-USER-ACTIVE-ALL best reflected the results of the live trials.

In cases where training and testing data overlap, i.e., training on PASSIVE and testing on APP-USER-PASSIVE, we get a random 20% subset of the training data per user for testing. These links are then removed from the training data to ensure that the set of links in the training data and set of links in the test data are disjoint.

Chapter 4

Comparison of Existing Recommender Systems

The recommendations used for this trial were

- *k*-Nearest Neighbor
- Support Vector Machines
- Matchbox: Matchbox MF + L2 Regularization
- Social Matchbox: Matchbox MF + Social Regularization + L2 Regularization

Social Matchbox uses the Social Regularizer to incorporate the social aspect of the data, while SVM incorporates a small amount of social information in the $\mathbf{f}_{\mathbf{x},\mathbf{z}}$ features it uses. Matchbox and Nearest Neighbors do not make use of any social information and are collaborative filtering recommenders.

4.1 Online Results

The first live user trial was run from August 1 to October 13. The algorithms were randomly distributed among the 106 users who installed the LinkR application. Each user was recommended 3 links everyday and they were able to rate the links on whether they 'Liked' or 'Disliked' it.

Social Matchbox and Support Vector Machines, the two algorithms that make use social information, garnered the most number of likes from the LinkR users, with Social Matchbox edging out SVM by just 4 likes. This suggests that using social information does indeed provide useful information that results in better recommendations.

When it comes to the number of dislikes, SVM garnered the most number of dislikes by a big margin. Social Matchbox and Matchbox came next, with Nearest neighbors having the least number of both likes and dislikes. Because Social Matchbox received the highest ratio of likes to dislikes among all the four algorithms, we considered Social Matchbox to be the best performing recommendation algorithm in this first trial.

Also, we looked at the algorithms with the results split between friend links and non-friend links recommendations. The majority of the strong results for Social Matchbox

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

Table 4.1: Number of Users Assigned per Algorithm.

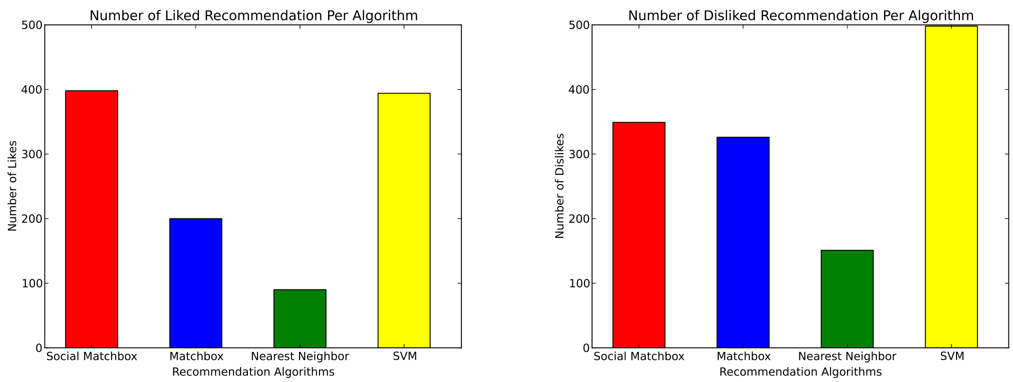

Figure 4.1: Results of online live trials

came when it was recommending friend links. When it recommended non-friend links, the number of likes was nearly half and the number of dislikes was double that of friends links. The rest of the algorithms also experienced the same lessening of likes and increase in dislikes when recommending non-friend links. Aside from liking and disliking a link just from the quality of links being recommended, it seems that users are also more likely to like a link simply because a friend had posted it and more likely to dislike it just because it comes from a stranger.

ost

4.2 Passive Results

The goal of the passive experiments was to see how best to reproduce the results of the live experiments offline. We tried out the different splitting combinations of training and testing data detailed in the last chapter, and found that training on the UNION dataset and and testing on the APP-USER-ACTIVE-ALL dataset best reflected the results of the online trials.

Additionally, when training on the UNION dataset, we can see the same general worsening of performance between the results of testing on APP-USER-ACTIVE-FRIENDS and APP-USER-ACTIVE-NON-FRIENDS.

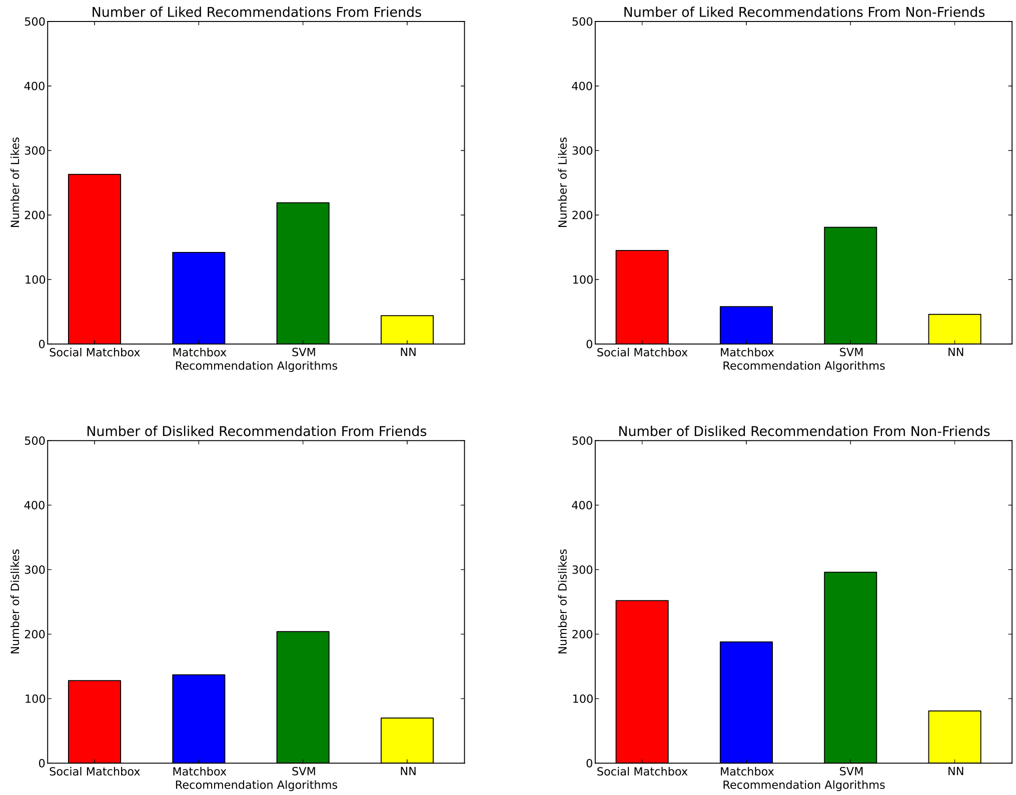

Figure 4.2: Likes and dislikes split between friends and non-friends

4.3 Survey Results

Near the end of the first trial, the LinkR users were invited to answer a survey regarding their experiences with the recommendations they were getting. They were asked a number of questions, with the following pertaining to the quality of the recommendations:

- Do you find that ANU LinkR recommends interesting links that you may not have otherwise seen?
- Do you feel that ANU LinkR has adapted to your preferences since you first started using it?
- How relevant are the daily recommended links?
- Overall, how satisfied are you with LinkR?

They gave their answers to each question as an integer rating with range $[1-5]$, with a higher value being better. Their answers were grouped together according to the recommendation algorithm that was assigned to them, and the averages per algorithm are below.

One more, we see that Social Matchbox achieved higher scores than the other recommendation algorithms, in all four questions. The results of the survey reflected the results in the online live trial and confirms that Social Matchbox was the best recommendation algorithm in the first trial.

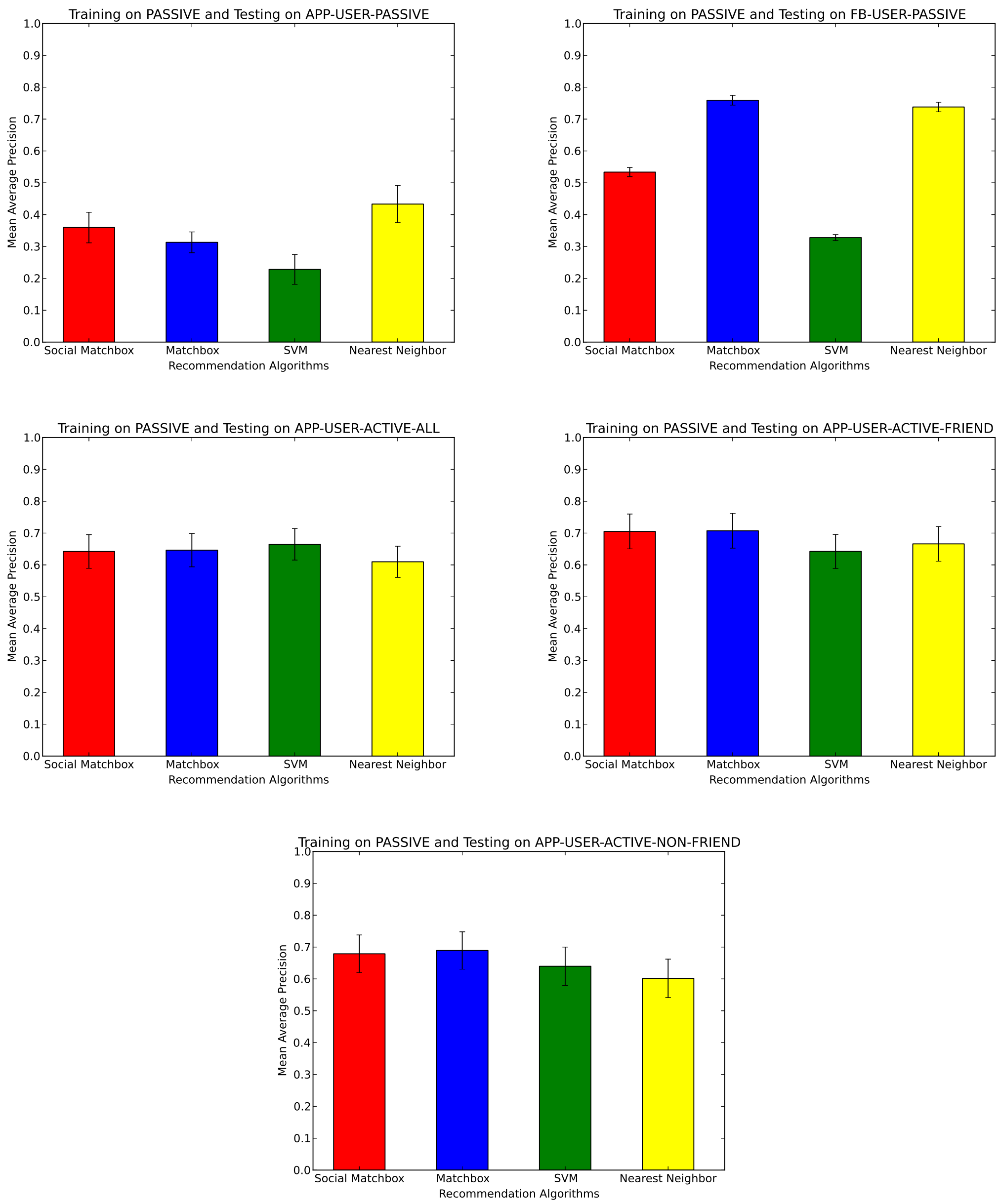

Figure 4.3: Results of training on Passive data

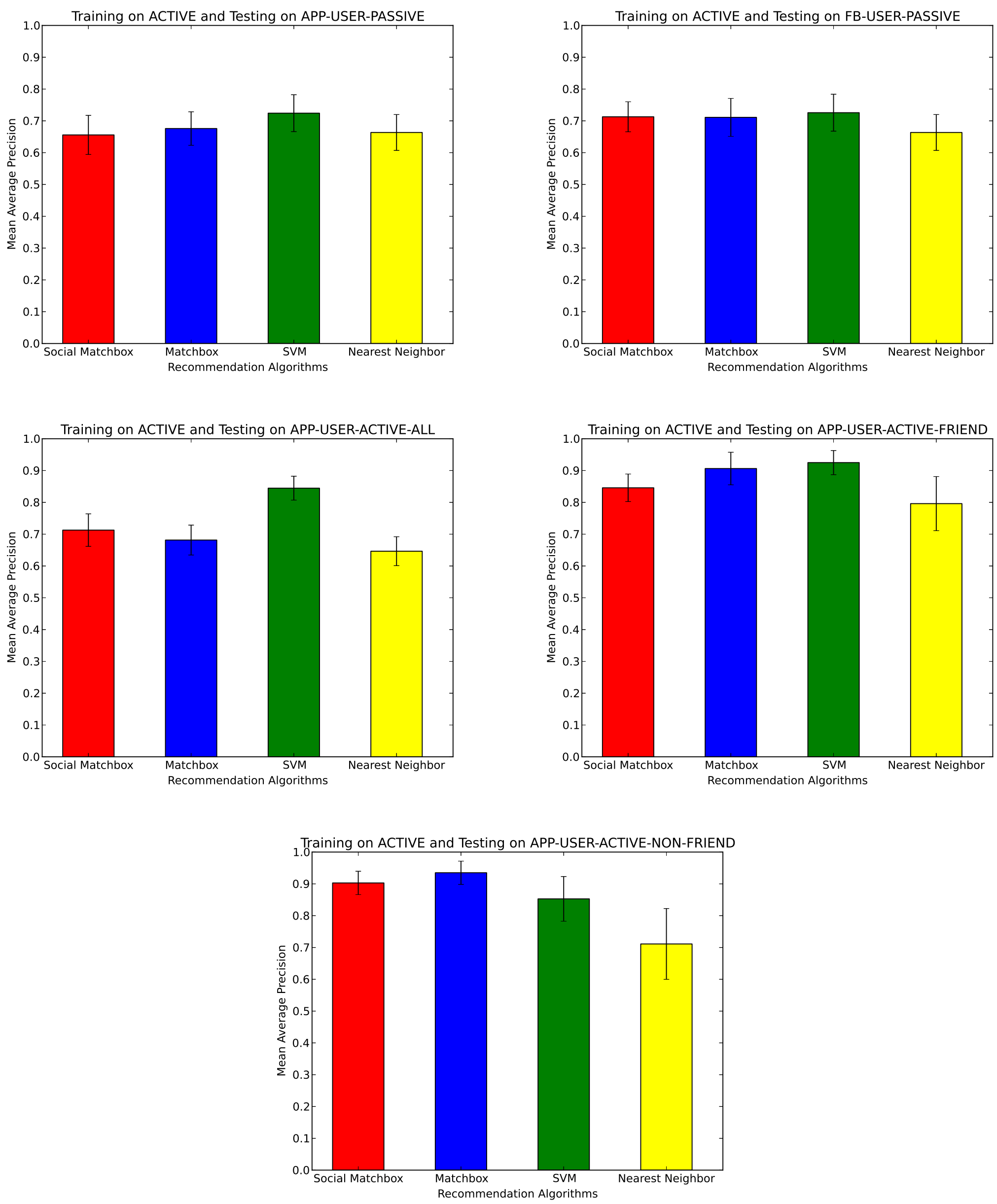

Figure 4.4: Results of training on Active data

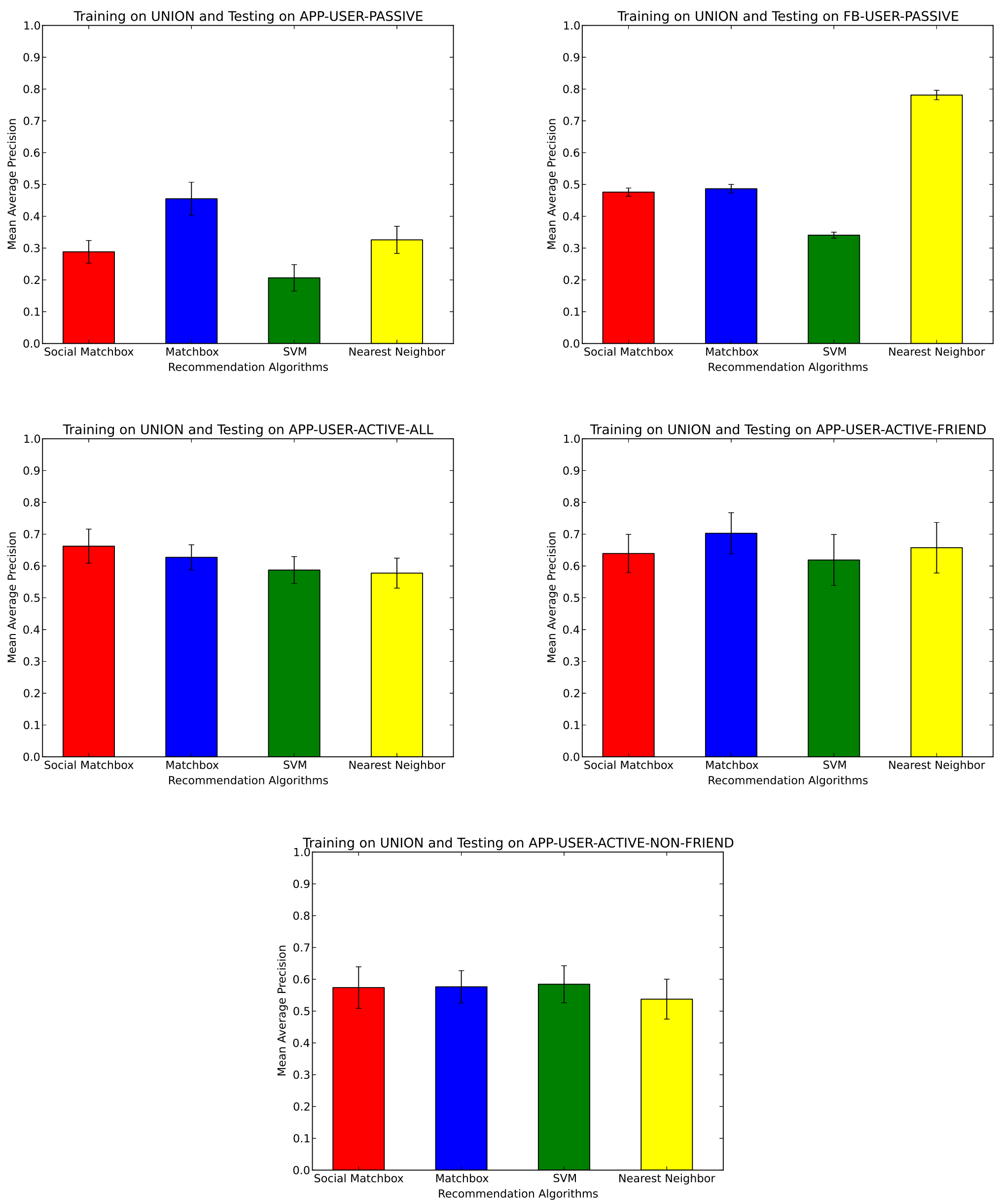

Figure 4.5: Results of training on Union data

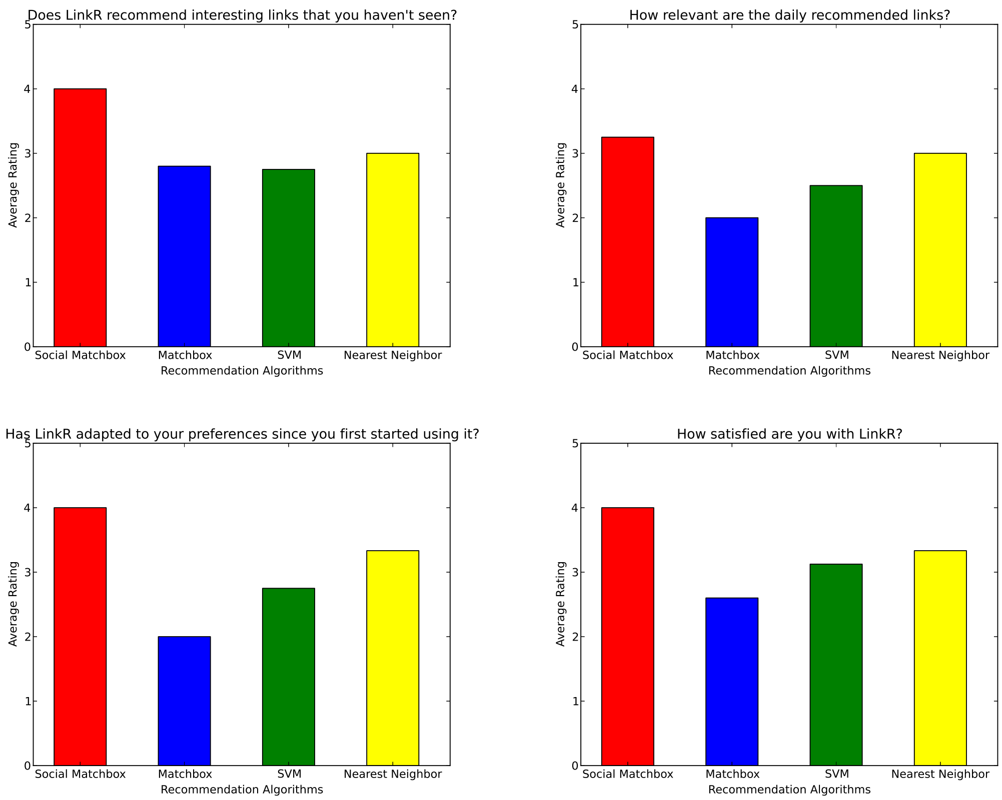

Figure 4.6: Results of user survey after the first trial

Chapter 5

New Algorithms for Social Recommendation

After the first trial, we made use of the results and what we learned to come up with new algorithms for social recommendation. Again, these new algorithms each form a component of a minimization objective Obj which is composed of sums of one or more objective components:

$$Obj = \sum_i \lambda_i Obj_i \tag{5.1}$$

Again, a sigmoidal transform

$$\sigma(o) = \frac{1}{1 + e^{-o}} \tag{5.2}$$

of regressor outputs $o \in \mathbb{R}$ is used to squash the outputs to the range $[0, 1]$. In places where the σ transform may be optionally included, this is written as $[\sigma]$.

5.1 Hybrid Objective

Because of the good performance of SVM, we decided to make use of its features $\mathbf{f}_{\mathbf{x},\mathbf{y}}$ to make another linear regressor. 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 component of linear CBF is therefore

$$\sum_{(\mathbf{x},\mathbf{y}) \in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{w}^T \mathbf{f}_{\mathbf{x},\mathbf{y}})^2 \tag{5.3}$$

However instead of using this linear CBF model by itself, we combine its predictions with the Matchbox matrix factorization prediction model $[\sigma]\mathbf{x}^T U^T V y$, to get a hybrid objective component. The full objective component for this hybrid model is

$$\sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{w}^T\mathbf{f}_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{x}^T U^T V y)^2 \tag{5.4}$$

5.2 New Social Regularizers

5.2.1 Social Spectral Regularization

$$\begin{aligned}
&\sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} \frac{1}{2} S^{+}_{\mathbf{x},\mathbf{z}} \|U\mathbf{x} - U\mathbf{z}\|_2^2 \\
&\quad = \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} \frac{1}{2} S^{+}_{\mathbf{x},\mathbf{z}} \|U(\mathbf{x} - \mathbf{z})\|_2^2 \\
&\quad = \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} \frac{1}{2} S^{+}_{\mathbf{x},\mathbf{z}} (\mathbf{x} - \mathbf{z})^T U^T U (\mathbf{x} - \mathbf{z})
\end{aligned} \tag{5.5}$$

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.

5.2.2 Social Co-preference Regularization

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 denoe a reweighted inner product. The objective component for social co-preference regularization along with its expanded form is

$$\begin{aligned}
&\sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2}(P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \langle U\mathbf{x}, U\mathbf{z}\rangle_{V\mathbf{y}})^2 \\
&\quad = \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2}(P_{\mathbf{x},\mathbf{z},\mathbf{y}} - \mathbf{x}^T U^T \operatorname{diag}(V\mathbf{y}) U\mathbf{z})^2
\end{aligned} \tag{5.6}$$

5.2.3 Social Co-preference Spectral Regularization

This is the same as the social co-preference regularization above, except that it uses the spectral regularizer format for learning the co-preferences.

We use $\|\cdot\|_{2,\bullet}$ to denote a re-weighted L_2 norm. The objective component for social co-preference spectral regularization along with its expanded form is

$$\begin{aligned}
&\sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2} P_{\mathbf{x},\mathbf{z},\mathbf{y}} \|U\mathbf{x} - U\mathbf{z}\|^2_{2,V\mathbf{y}} \\
&\quad = \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2} P_{\mathbf{x},\mathbf{z},\mathbf{y}} \|U(\mathbf{x} - \mathbf{z})\|^2_{2,V\mathbf{y}} \\
&\quad = \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in C} \frac{1}{2} P_{\mathbf{x},\mathbf{z},\mathbf{y}} (\mathbf{x} - \mathbf{z})^T U^T \operatorname{diag}(V\mathbf{y}) U(\mathbf{x} - \mathbf{z})
\end{aligned} \tag{5.7}$$

5.2.4 Derivatives

As before, we seek to optimize sums of the above objectives and will use gradient descent for this purpose. We again use the following useful abbreviations:

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

The derivatives for the linear CBF and hybrid objective functions, as well as the new social regularizers are

- **Explicit Linear CBF**:

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

- **Hybrid**:

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

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

$$= \sum_{(\mathbf{x},\mathbf{y})\in D} \delta_{\mathbf{x},\mathbf{y}} \frac{\partial}{\partial U} - [\sigma]\mathbf{x}^T U^T V \mathbf{y}$$

$$= - \sum_{(\mathbf{x},\mathbf{y})\in D} \delta_{\mathbf{x},\mathbf{y}} [\sigma(o^2_{\mathbf{x},\mathbf{y}})(1 - \sigma(o^2_{\mathbf{x},\mathbf{y}}))]\mathbf{t}\mathbf{x}^T$$

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

$$= \sum_{(\mathbf{x},\mathbf{y})\in D} \delta_{\mathbf{x},\mathbf{y}} \frac{\partial}{\partial V} - [\sigma]\mathbf{x}^T U^T V \mathbf{y}$$

$$= - \sum_{(\mathbf{x},\mathbf{y})\in D} \delta_{\mathbf{x},\mathbf{y}} [\sigma(o^2_{\mathbf{x},\mathbf{y}})(1 - \sigma(o^2_{\mathbf{x},\mathbf{y}}))]\mathbf{s}\mathbf{y}^T$$

- **Social spectral regularization**:

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

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

$$= \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends(\mathbf{x})} S^+_{\mathbf{x},\mathbf{z}} U(\mathbf{x}-\mathbf{z})(\mathbf{x}-\mathbf{z})^T$$

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

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

- **Social co-preference regularization**:

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

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

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

- **Social co-preference spectral regularization**:

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

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 according to (2.7).

5.3 Second Trial

For the second online trial, we chose four algorithms again to randomly split between the LinkR application users. The four recommendation algorithms are:

- Social Matchbox: Matchbox MF + Social Regularization + L2 Regularization
- Spectral Matchbox: Matchbox MF + Social Spectral Regularization + L2 Regularization
- Social Hybrid: Hybrid + Social Regularization + L2 Regularization
- Spectral Co-preference: Matchbox MF + Social Co-preference Spectral Regularization + L2 Regularization

Social Matchbox was included again as a baseline since it was the best performing algorithm in the first trial. The other algorithms were choses as a cross-section of the different regularizers. The limit of just using four algorithms for the trials was because of the small number of users who installed the LinkR application, we felt that four algorithms was the maximum we could use and still get concrete results.

5.3.1 Results

In the second trial, the Spectral Matchbox approach achieve a larger number of likes compared to the other algorithms. It had 10 more likes compared to the second place

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

Table 5.1: Number of Users Assigned per Algorithm.

algorithm, Spectral Co-preference. On top of this, Spectral Matchbox also had the second lowest number of dislikes among the four algorithms, and was the only algorithm to have more likes than dislikes. Because it had the best ratio of likes to dislikes, we consider Spectral Matchbox to be the best performing algorithm during this round of live trials.

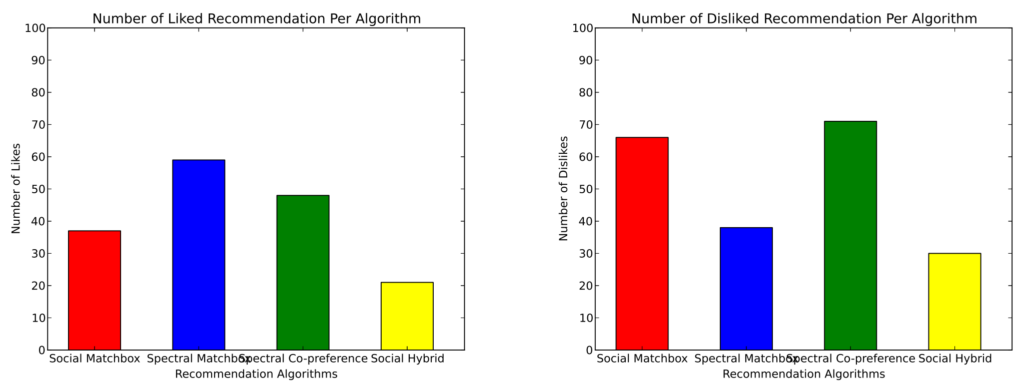

Figure 5.1: Results of online live trials

Splitting the results again between friend link recommendations and non-friend link recommendations, we again see that the recommendations worsened when it comes to recommending non-friend links. The number of likes decreased and the number of dislikes increased once again. Also, the differences were more drastic again with the two algorithms that uses the Social Regularizer, Social Matchbox and Social Hybrid.

5.4 Offline Results

When testing on all the combinations of the active data, the results between the algorithms on a test dataset are all within the standard error bars. The results of the live trials aren't reflected in the offline results, which highlights the difficulty of using the MAP metric for evaluating recommendation algorithms offline.

When training on the UNION dataset, we can see the same general worsening of performance between the results of testing on APP-USER-ACTIVE-FRIENDS and APP-USER-ACTIVE-NON-FRIENDS.

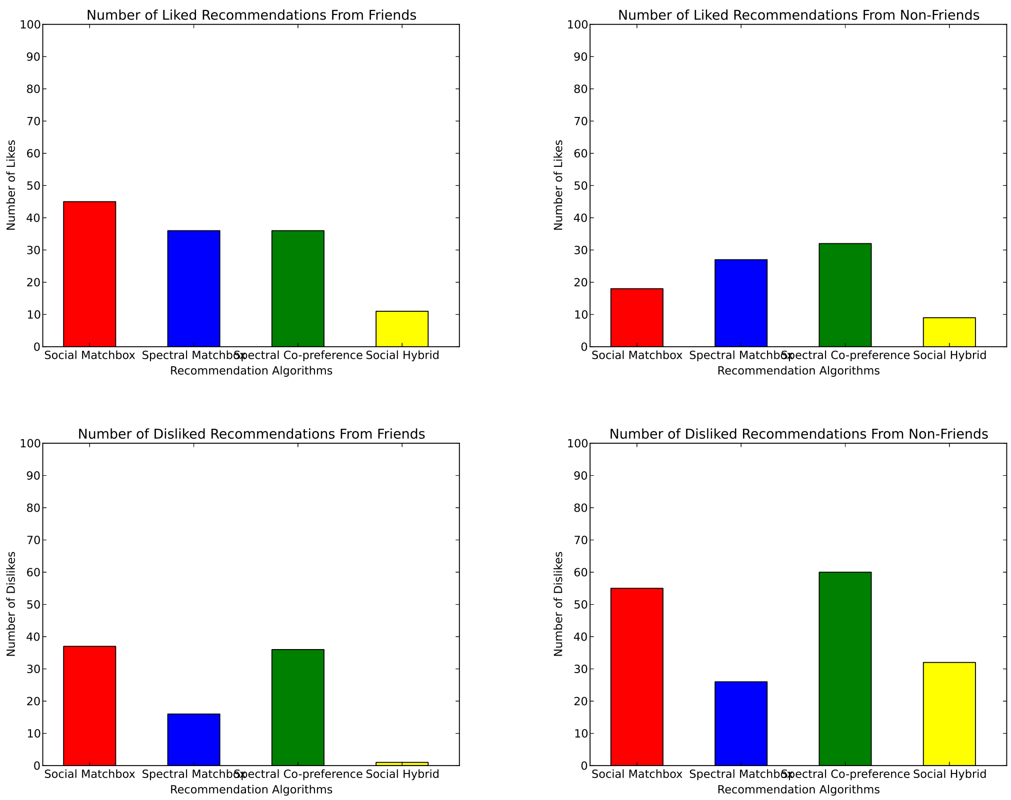

Figure 5.2: Likes and dislikes split between friends and non-friends

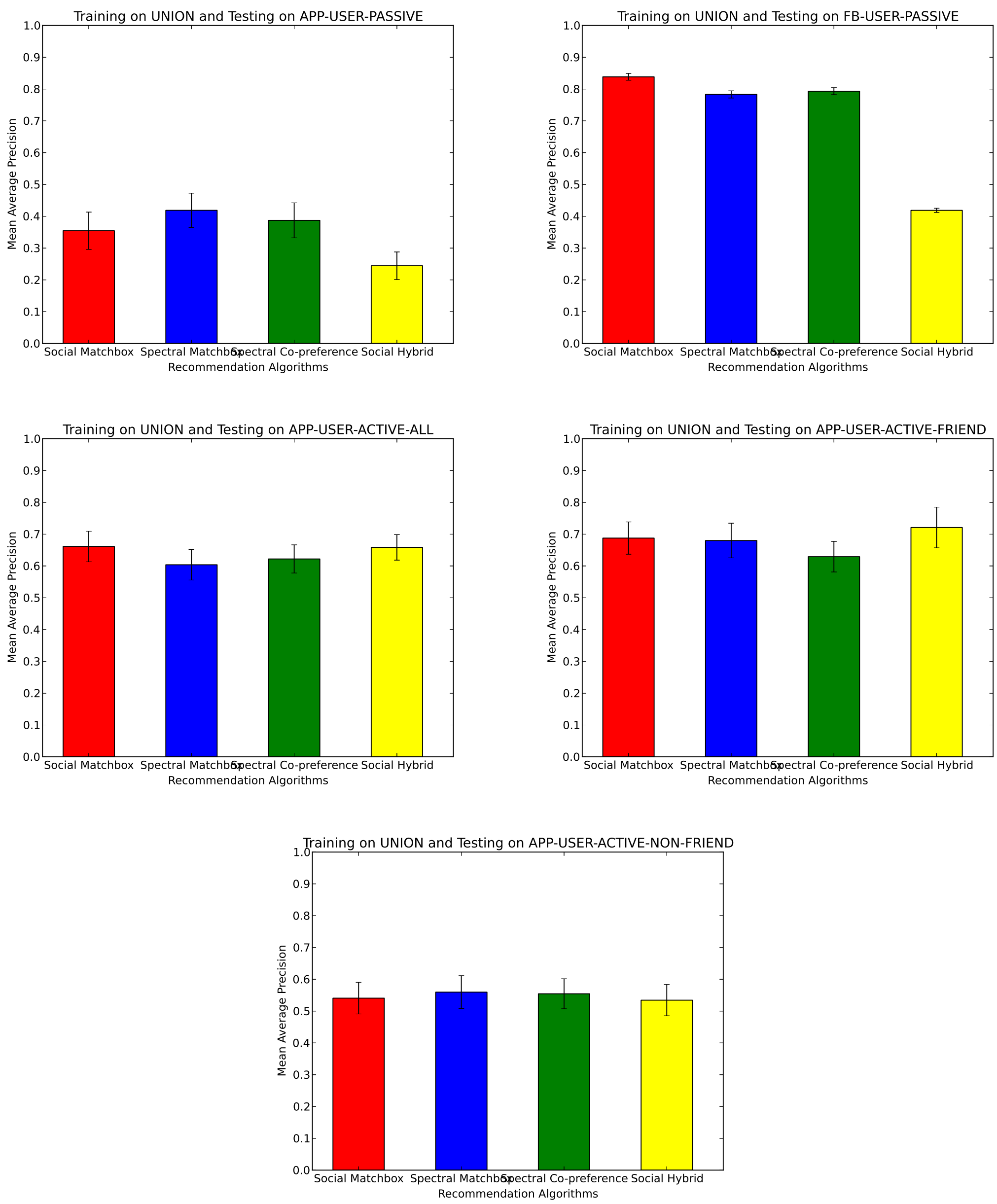

Figure 5.3: Results of training on Union data

Chapter 6

Conclusion

The Social Regularization constraint of Social Matchbox which has been detailed in other papers gives a very strong performance as a social recommender. It handily beat the other recommenders in the first trial and this was also reflected in offline experiments when training on the UNION dataset and testing on APP-USER-ACTIVE-ALL, the dataset combination that most closely resembles the training and recommendation data of the live trials. We also found that the Social Regularization constraint performed very differently when recommending friend links and when recommending non-friend links, and this was also reflected in the offline experiments.

The results of the second trial indicate that the Social Spectral Regularization constrain may be better for social regularization, the two algorithms that used it performed better than the two algorithms that used the Social Regularization constraint. However, this result can't be found in the offline experiments. What can be found in the offline experiments when training on the UNION data was same decrease in performance of the algorithms when recommending non-friend links compared to recommending friend links.

The results of the second trial highlight one difficulty encountered with evaluating the recommendation algorithms: which metric best correlates with human preferences. This paper uses the mean average precision as the main evaluation metric, but there may be other metrics that can better reflect the results of live user trials. Future work on social recommendation that may be crucial can be evaluation of different metric show closely they reflect user preferences.

The social recommenders discussed in this paper try to project the user preferences into the latent space, but it was found that other implementation issues greatly affect the user perceptions of the quality of the links being recommended. One complaint was that since the majority of the LinkR users were mainly English speakers, they automatically disliked links with non-English descriptions and those that pointed to non-English pages. A quick fix we did for this was to just stop recommending non-English links. In the future, having known languages in the user feature and the language of the link or its description in the link feature may result in a lot better and more accurate recommendation, especially for users knowledgeable in non-English languages.

With regards to link recommendations in LinkR, it was found that users were more likely to like recommended links when those links were originally posted by their friends, as opposed to links that were originally posted by non-friends. This was found in both the first and second user trials, and even in the offline experiments. This may be because users will trust a link more when it has been "vouched" for by a friend posting it, rather than just any link from a random stranger.

Bibliography

Birlutiu, A., Groot, P., and Heskes, T. 2010. Multi-task preference learning with an application to hearing aid personalization. *Neurocomputing 73*, 7-9, 1177–1185.

Bonilla, E. V., Chai, K. M. A., and Williams, C. K. I. 2008. Multi-task gaussian process prediction. In J. Platt, D. Koller, Y. Singer, and S. Roweis Eds., *Advances in Neural Information Processing Systems 20*, pp. 153–160. Cambridge, MA: MIT Press.

Boutilier, C. 2002. A POMDP formulation of preference elicitation problems. In *Proceedings of the 18th National Conference on Artificial Intelligence* (Menlo Park, CA, USA, 2002), pp. 239–246. American Association for Artificial Intelligence.

Bradley, R. A. and Terry, M. E. 1952. Rank analysis of incomplete block designs: The method of paired comparison. *Biometrika 39*, 324–345.

Cao, B., Sun, J.-T., Wu, J., Yang, Q., and Chen, Z. 2008. Learning bidirectional similarity for collaborative filtering. In *ECML-08* (2008).

Chajewska, U. and Koller, D. 2000. Utilities as random variables: Density estimation and structure discovery. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence* (2000), pp. 63–71. Morgan Kaufmann Publishers Inc.

Chajewska, U., Koller, D., and Parr, R. 2000. Making rational decisions using adaptive utility elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence* (2000), pp. 363–369. AAAI Press / The MIT Press.

Chu, W. and Ghahramani, Z. 2005a. Gaussian processes for ordinal regression. *Journal of Machine Learning Research 6*, 1019–1041.

Chu, W. and Ghahramani, Z. 2005b. Preference learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning* (New York, NY, USA, 2005), pp. 137–144. ACM.

Conitzer, V. 2009a. Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research 35*, 161–191.

Conitzer, V. 2009b. Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research 35*, 161–191.

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

Eric, B., Freitas, N. D., and Ghosh, A. 2008. Active preference learning with discrete choice data. In J. Platt, D. Koller, Y. Singer, and S. Roweis Eds., *Advances in Neural Information Processing Systems 20*, pp. 409–416. Cambridge, MA: MIT Press.

Guo, S. and Sanner, S. 2010. Real-time multiattribute Bayesian preference elicitation with pairwise comparison queries. In *Thirteenth International Conference on Artificial Intelligence and Statistics* (2010).

Howard, R. 1966. Information value theory. *IEEE Transactions on Systems Science and Cybernetics 2*, 1, 22–26.

Huang, J. and Guestrin, C. 2009. Riffled independence for ranked data. In *In Advances in Neural Information Processing Systems (NIPS)* (Vancouver, Canada, December 2009).

Jones, D. R. 2001. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization 21*, 4, 345–383.

Kamishima, T. 2003. Nantonac collaborative filtering: recommendation based on order responses. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2003), pp. 583–588. ACM.

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

Li, W.-J. and Yeung, D.-Y. 2009. Relation regularized matrix factorization. In *IJCAI-09* (2009).

Ma, King, and Lyu. 2009. Learning to recommend with social trust ensemble. In *SIGIR-09* (2009).

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

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

Quiñonero Candela, J. and Rasmussen, C. E. 2005. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research 6*, 1939–1959.

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

Stern, D. H., Herbrich, R., and Graepel, T. 2009. Matchbox: large scale online bayesian recommendations. In *WWW-09* (2009), pp. 111–120.

Viappiani, P. and Boutilier, C. 2009. Regret-based optimal recommendation sets in conversational recommender systems. In *Proceedings of the third ACM conference on Recommender systems* (New York City, NY, USA, October 2009), pp. 101–108. ACM.

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

Yu, K., Tresp, V., and Schwaighofer, A. 2005. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning* (New York, NY, USA, 2005), pp. 1012–1019. ACM.