

Affiliation Recommendation using Auxiliary Networks

Vishvas Vasuki
Zhengdong Lu

Nagarajan Natarajan
Inderjit Dhillon

Department of Computer Science and ICES, University of Texas at Austin.

ABSTRACT

Social network analysis has attracted increasing attention in recent years. In many social networks, besides friendship links amongst users, the phenomenon of users associating themselves with groups or communities is common. Thus, two networks exist simultaneously: the friendship network among users, and the affiliation network between users and groups. In this paper, we tackle the affiliation recommendation problem, where the task is to predict or suggest new affiliations between users and communities, given the current state of the friendship and affiliation networks. More generally, affiliations need not be community affiliations - they can be a user's taste, so affiliation recommendation algorithms have applications beyond community recommendation. In this paper, we show that information from the friendship network can indeed be fruitfully exploited in making affiliation recommendations. Using a simple way of combining these networks, we suggest two models of user-community affinity for the purpose of making affiliation recommendations: one based on graph proximity, and another using latent factors to model users and communities. We explore the two classes of affiliation recommendation algorithms suggested by these models. We evaluate these algorithms on two real world networks - Orkut and Youtube. In doing so, we motivate and propose a way of evaluating recommenders, by measuring how good the top 50 recommendations are for the average user, and demonstrate the importance of choosing the right evaluation strategy. The algorithms suggested by the graph proximity model turn out to be the most effective and efficient. This use of link prediction techniques for the purpose of affiliation recommendation is, to our knowledge, novel.

Categories and Subject Descriptors

H.2.8 [Database applications - Data mining]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

RecSys2010, September 26–30, 2010, Barcelona, Spain.

Copyright 2010 ACM 978-1-60558-906-0/10/09 ...\$10.00.

General Terms

Algorithms, Experimentation

1. INTRODUCTION

There has been an explosion in the number of online social networks and their active members. This wealth of information in the social networks has driven prolific work on the analysis of the networks, understanding the processes that explain the evolution of the networks, modeling the spread of behavior through the networks, predicting their future state and so on.

Users of a social network tend to affiliate with communities. In some social networks, the groups are identified more by the preferences of the members of the social network than by direct declaration: e.g. the genre of movies that a set of customers tend to patronize more often in Netflix. Online social networks like Facebook, Orkut and Live Journal are more interesting examples because the affiliation networks here are explicitly established by the members of the network. Thus, two networks exist simultaneously: the friendship network among users, and the affiliation network between users and groups.

The problem. Group formation and evolution in social networks[1], and co-evolution of social and affiliation networks[13] have been recently studied. One of the interesting challenges in social network analysis is the affiliation recommendation problem, where the task is to recommend communities to users. The fact that the social and affiliation networks “co-evolve” suggests that a better solution to the affiliation recommendation problem can be obtained if the friendship network is considered along with the affiliation network. This problem setting has applications beyond community recommendation. Affiliations, for example, can be interpreted in general as a user's taste for an item. Neither is it limited to social networks. For example, in biology, the friendship network can correspond to a network among genes, whereas the affiliation network can correspond to a network between genes and traits/ diseases, and the affiliation recommendation problem can be viewed as one of identifying genes affecting the expression of a disease.

Contributions. We consider how one can model the interplay between users and communities in both the networks simultaneously. An ideal unifying model would not only explain the current state of the networks, but also help in predicting future relationships among the nodes. Using a

simple way of combining these networks, we suggest and explore two ways of modeling the networks for the purpose of making affiliation recommendations. The graph proximity model is based on estimating the affinity between a user and a community by considering their proximity as nodes in a combined graph, while the latent factors model is based on the proposition that community affiliations arise from interactions of *user* and *group factors*. Each of these network models suggests affiliation recommendation algorithms.

We evaluate these algorithms on social networks from Orkut consisting of 9,123 users and 75,546 communities, and Youtube consisting of 16,575 users and 21,326 communities. We propose a way of evaluating affiliation recommendations, by measuring how good the top 50 recommendations per user are, and demonstrate the importance of designing the right evaluation strategy. Of the algorithms proposed, those suggested by the graph proximity model turn out to be the most effective and efficient. This use of link prediction techniques for the purpose of affiliation recommendation is, to our knowledge, novel. We show that information in the friendship network can be used effectively for affiliation recommendation. We also observe that the benefit we derive from the social network in affiliation recommendation is strongly contingent on how the problem is modeled and what algorithms are used.

Overview. We now provide a brief overview of the organization of the paper. In Section 2, we consider a network formed by merging the friendship and affiliation networks and introduce two models of the behaviour of nodes in this network: the graph proximity model, and the latent factors model, and explore recommendation algorithms that arise from these models. In Section 3, we consider how the proposed models and algorithms relate to prior work. Then, in Section 4, we describe our chosen evaluation strategy, and using experiments on real world networks, we evaluate the effectiveness of various algorithms in affiliation recommendation in practical scenarios. Finally, in Section 5, we conclude with a summary of our findings, and a discussion of future lines of research in this direction.

2. MODELS

In this section, we first establish the notation used, and then pose the affiliation recommendation problem as a *ranking* problem. Then we describe a natural way of combining the friendship and affiliation networks into a single graph. In the subsections that follow, we describe affiliation recommendation approaches based on the graph proximity model and those based on the latent factors model.

Notation. We use N_u to denote the total number of users in the affiliation network and N_g to denote the total number of communities in the affiliation network. $\mathbf{A} \in \mathbb{R}^{N_u \times N_g}$ denotes the user \times group adjacency matrix of affiliation network \mathcal{A} . $\mathbf{S} \in \mathbb{R}^{N_u \times N_u}$ denotes the user \times user adjacency matrix of friendship network \mathcal{S} . Other notation will be introduced as needed.

As a ranking problem. The task is to recommend affiliations to a given user. This problem can be posed as a problem of ranking various affiliations in the order of the user’s interest in joining them. The methods we describe

here to solve this problem rely on assigning scores to various affiliations in order to rank them. The task of an affiliation recommendation algorithm can be viewed as one of generating an $N_u \times N_g$ score matrix.

2.1 Prediction on the Combined Graph

Consider the adjacency matrices \mathbf{A} and \mathbf{S} . For now, we assume \mathbf{S} to be symmetric (or equivalently \mathcal{S} to be undirected), although a non-symmetric extension to our model can be easily obtained. Clearly, \mathbf{S} corresponds to an undirected graph among users and $\begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix}$ corresponds to an undirected bipartite graph between users and groups. Despite the heterogeneity of the two types of “links”, it is rather natural to consider a graph \mathcal{C} between all the users and groups, with the combined adjacency matrix

$$\mathbf{C} = \begin{bmatrix} \lambda \mathbf{S} & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix},$$

where the heterogeneity of two types of links is reduced to a single parameter $\lambda \geq 0$, that controls the ratio of the weight of friendship to the weight of group membership. Clearly when $\lambda = 0$, the user-user friendship ceases to play a role in this joint graph, and it simply degenerates to the bipartite affiliation graph.

As in the case of prediction with a single graph, the prediction on \mathcal{C} can be carried out from the following two perspectives:

- **Graph Proximity Model:** We assume that the entire graph is known, and that the prediction of new or unobserved links is based on an estimated proximity in \mathbf{C} between nodes.
- **Latent Factors Model:** We model the graph as a matrix, some of whose zero entries are actually ones, while modelling all entries as the product of latent factors.

We will elaborate on the two perspectives in the following two subsections.

2.1.1 Graph Proximity Model

As described earlier, the affiliation network can be modelled by a graph. The graph proximity model assumes that the probability of there arising a link between two nodes in a graph is based on an estimated proximity between the two nodes. The proximity of two vertices can be calculated as the weighted sum of the number of paths connecting the two with varying lengths. This is the underlying mechanism of many link prediction models in the context of social network analysis. Consider the widely-used Katz measure[8] on the friendship network \mathcal{S} . The proximity is given by

$$\text{Katz}(\mathbf{S}; \beta) = \beta \mathbf{S} + \beta^2 \mathbf{S}^2 + \beta^3 \mathbf{S}^3 + \dots$$

where the weights of paths decay exponentially with the length.

A simple extension of Katz to the bipartite graph \mathcal{A} is

$$\begin{aligned} \text{Katz}(\mathbf{A}; \beta) &= (\beta \mathbf{A} \mathbf{A}^T + \beta^2 (\mathbf{A} \mathbf{A}^T)^2 + \beta^3 (\mathbf{A} \mathbf{A}^T)^3 + \dots) \mathbf{A} \\ &= \beta \mathbf{A} \mathbf{A}^T \mathbf{A} + \beta^2 (\mathbf{A} \mathbf{A}^T)^2 \mathbf{A} + \dots \end{aligned} \quad (1)$$

where in the co-occurrence matrix $\mathbf{A} \mathbf{A}^T$, two users i and j are considered connected if i and j belong to at least one

same group, i.e. $(\mathbf{A}\mathbf{A}^T)_{ij} > 0$. In this case, we consider the paths of the following types

$$\text{user } i \xrightarrow{\mathbf{A}\mathbf{A}^T} j \cdots \xrightarrow{\mathbf{A}\mathbf{A}^T} k \xrightarrow{\mathbf{A}} \text{to group } n$$

The intuition in considering $\mathbf{A}\mathbf{A}^T\mathbf{A}$ is if user i shares some community with user j , it is likely that i will join some other community j belongs to. The higher order terms can be interpreted in a similar way.

Consider the following Katz-measure proximity matrix on the combined graph \mathcal{C}

$$\text{katz}(\mathcal{C}; \beta) = \beta\mathbf{C} + \beta^2\mathbf{C}^2 + \beta^3\mathbf{C}^3 + \cdots$$

The user-community block of the above matrix is simply

$$\text{katz}(\mathcal{C}; \beta)_{12} = \beta\mathbf{A} + \beta^2\lambda\mathbf{S}\mathbf{A} + \beta^3(\lambda^2\mathbf{S}^2\mathbf{A} + \mathbf{A}\mathbf{A}^T\mathbf{A}) + \cdots \quad (2)$$

Clearly (2) generalizes the normal Katz on the bipartite graph \mathbf{A} by also considering some paths like

$$\text{user } i \xrightarrow{\mathbf{S}} j \xrightarrow{\mathbf{A}} \text{group } n \text{ (in } \mathbf{C}^2)$$

and

$$\text{user } i \xrightarrow{\mathbf{S}} j \xrightarrow{\mathbf{A}\mathbf{A}^T} k \xrightarrow{\mathbf{A}} j \text{ group } n \text{ (in } \mathbf{C}^4).$$

$\text{katz}(\mathcal{C}; \beta)_{12}$ can then be used as the score matrix.

Since the computation of the Katz matrices as described above is expensive, we consider a truncated Katz matrix,

$$t\text{Katz}(T, k, \beta) = \sum_{i=1}^k \beta^i T^i.$$

For computing the Katz matrices for the graphs described by \mathbf{A} or \mathbf{C} , a conservative estimate of the computational cost is $O(N_u \times nnz)$, where nnz is the number of non-zeros in $(\mathbf{A}\mathbf{A}^T)^k$.

2.1.2 Latent Factors Model

We now describe the latent factors model, consider the optimization problem it tries to solve, and examine some of its properties. In this model, the zeros in the adjacency matrix of the affiliation network, \mathbf{A} may be viewed as being unobserved entries with a huge prior belief in favor of them being actually zero. Every user i and community j are assumed to have a low dimensional representations u_i, g_j . The affinity of a user i to a community j is assumed to correspond to $u_i^T g_j$. In other words, users and communities with a high inner product are assumed to be connected to each other; that is to say:

$$\mathbf{A}_{i,j} \approx u_i^T g_j.$$

In matrix form, we express this as

$$\mathbf{A} \approx \mathbf{U}^T \mathbf{G},$$

where $\mathbf{U} \in \mathbb{R}^{d \times N_u}$ is the matrix of user factors and $\mathbf{G} \in \mathbb{R}^{d \times N_g}$ is the matrix of group factors. Note that $\text{rank}(\mathbf{U}) \leq d$, $\text{rank}(\mathbf{G}) \leq d$, $d \ll N_u, N_g$.

To get factors which account for both the observed entries in \mathbf{A} as well as the interactions in \mathbf{S} we consider the following generalized combined network,

$$\mathbf{C}'(\lambda, \mathbf{D}) = \begin{bmatrix} \lambda\mathbf{S} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{D} \end{bmatrix}, \quad (3)$$

where \mathbf{D} is the *derived* similarity between groups which is *not* observed. Note that $\mathbf{C} = \mathbf{C}'(\lambda, 0)$. Also, we approximate \mathbf{C}' as an interaction of user and group factors of the

individual graphs. i.e.

$$\mathbf{C}'(\lambda, \mathbf{D}) \approx \begin{bmatrix} \mathbf{U}^T \\ \mathbf{G}^T \end{bmatrix} [\mathbf{U} \quad \mathbf{G}].$$

The same set of user factors drive both the friendship and affiliation network creations. We therefore want to find the user factors \mathbf{U} and the group factors \mathbf{G} such that the reconstruction error on the observed entries in \mathbf{C}' ,

$$\|\mathbf{U}^T \mathbf{U} - \lambda\mathbf{S}\|^2 + 2\|\mathbf{U}^T \mathbf{G} - \mathbf{A}\|^2 + \|\mathbf{G}^T \mathbf{G} - \mathbf{D}\|^2 \quad (4)$$

is minimized.

Role of λ . Intuitively, in equation (4), λ controls the contribution of \mathbf{S} in deciding the user factors. The larger the λ , the learned factor model describes the friendship network \mathcal{S} better, and correspondingly the affiliation network is described less well.

Choice of \mathbf{D} . The derived similarity between the communities \mathbf{D} in the combined matrix \mathbf{C}' can be approximated using proximity between communities in the graph corresponding to \mathbf{C}' . It follows that a potential choice is $\mathbf{D} = \mathbf{A}^T \mathbf{A}$, which is simply the number of users which any two communities share. One may also consider weighing the contribution of \mathbf{D} with λ , which is the weight factor learnt for \mathbf{S} . We will see later that the experiments suggest that the choice of \mathbf{D} is not very important, and that the information from \mathbf{D} , when derived from \mathbf{A} , is redundant.

2.2 Solution to the optimization problem

The analytical solution to the minimizing $\{\mathbf{U}, \mathbf{G}\}$ of the objective function in (4) is given by the SVD of \mathbf{C}' . Specifically,

$$\text{argmin}_{\mathbf{U}, \mathbf{G}} \{ \|\mathbf{U}^T \mathbf{U} - \lambda\mathbf{S}\|^2 + 2\|\mathbf{U}^T \mathbf{G} - \mathbf{A}\|^2 + \|\mathbf{G}^T \mathbf{G} - \mathbf{D}\|^2 \} = \text{SVD}(\mathbf{C}', d) \quad (5)$$

where $\text{SVD}(\mathbf{C}', d)$ denotes the rank- d singular value decomposition of matrix \mathbf{C}' , i.e. $\mathbf{C}' \approx \mathbf{U}\mathbf{\Sigma}\mathbf{G}^T$, where $\mathbf{\Sigma}$ is the $d \times d$ matrix of singular values, \mathbf{U} is the matrix of d leading left singular vectors, and \mathbf{G} is the matrix of d leading right singular vectors. We can interpret the low rank approximation $\mathbf{U}\mathbf{\Sigma}\mathbf{G}^T$ as the score matrix.

3. RELATED WORK

Increasing attention to recommendation systems in general can be attributed primarily to commercial enterprises like Netflix and Amazon, where making good recommendations for the customers is important for business. A huge body of literature studies the problem of group recommendation, where the problem is to recommend items or products to a group of users in a friendship network. Affiliation recommendation for users of a friendship network, however, is relatively new and less studied.

3.1 Joint matrix factorization models

We now examine the relationship of the latent factors model proposed in this paper to a variety of other models proposed recently. Bader and Chew[2], in the context of information retrieval, tackle the problem of applying LSA to multi-lingual corpora. In such corpora, one has access to term-similarity information along with term-document matrices corresponding to various languages. In order to

derive low dimensional term and document factors which account for information from both these sources, they form a joint matrix similar to our combined adjacency matrix \mathbf{C} and compute its SVD. However, unlike this work, [2] does not deal with the item recommendation problem, and it does not view this joint matrix as arising out of a pair of networks.

We will now consider two other joint matrix factorization models: One class of models uses a probabilistic collaborative filtering to approach the problem, whereas another tackles the problem of combining information from multiple sources from the perspective of joint matrix factorization.

3.1.1 Probabilistic Collaborative Filtering Models

Collaborative filtering is a natural way to approach the affiliation recommendation problem. Typically collaborative filtering is applied to user-item preference problems. This is based on the simple idea that users with similar tastes behave similarly.

This approach has recently been applied to the affiliation recommendation problem by Chen et al[4]. The authors examined the use of Latent Dirichlet Allocation (LDA) in affiliation recommendation. The LDA approach does not use information from the friendship network among users. So, here we briefly examine the relationship between the latent factors model we propose and this LDA based approach, while ignoring the friendship network aspects of our model.

Consider the objective (4) we are trying to minimize. In the proposed model, if we ignore the constraint that the user factors \mathbf{U} do not result in too large a deviation from \mathbf{S} , we are essentially trying to find a low rank approximation to \mathbf{A} in terms of the Frobenius norm. The solution which minimizes that objective is given by the SVD of \mathbf{A} . This is the Latent Semantic Analysis approach (LSA), which has long been exploited for similar problems in the area of information retrieval. pLSA, or probabilistic LSA [6], instead proposes a statistical model for the process generating \mathbf{A} and then learns the model parameters which are most likely to have generated the observations in \mathbf{A} . These parameters can then be used in finding a low rank approximation to \mathbf{A} , in terms of the KL divergence. It can thus be viewed as the probabilistic version of LSA. LDA, where Dirichlet priors are added to pLSA’s generative model, can be viewed as the Bayesian version of pLSA. Thus, the LDA based approach to the affiliation recommendation problem may be viewed as trying to find a low rank approximation to \mathbf{A} , albeit from a probabilistic, Bayesian perspective, while ignoring information from \mathbf{S} .

Combinatorial collaborative filtering[5] is another work in the same vein. Unlike the LDA based approach, however, the probabilistic model of user-community relationships used in this work utilizes information not only from \mathbf{A} , but also from text descriptions of various communities. Next, we examine a couple of closely related matrix factorization models.

3.1.2 Linked Matrix Factorization Models

Tang et al have proposed Linked Matrix Factorization (LMF) [12] as a way of combining information from multiple graphs on the same set of entities, in order to make more accurate inferences. However, they tackle a different problem, namely, clustering. The link between their network model and ours is established by the objective functions that we optimize. The LMF model tries to simultaneously find

a low rank approximation for the adjacency matrix of each network, using matrix factorization. Each such matrix factorization has a source-specific factor matrix, $\Lambda^{(m)}$, and a factor matrix, \mathbf{P} , that is shared by all the sources. The objective function of LMF is effectively to minimize the quantity,

$$\sum_{i=1}^M \|\mathbf{A}^{(m)} - \mathbf{P}\Lambda^{(m)}\mathbf{P}^T\|_F^2$$

Comparing this to (4), we see that \mathbf{U} , which represents the user factors, is shared by the two sources of information, i.e., \mathbf{A} and \mathbf{S} . However, an important distinction is that we have two graphs which share only one set of entities in common, whereas in LMF, each source of information is a network on exactly the same set of users.

Singh and Gordon have proposed a model for relational learning called Collective Matrix Factorization[11]. They suggest a generalized framework for inferring relations, given a set of entities and observed relations among them. This model factors multiple source matrices simultaneously, and uses common factors for approximation whenever the same entity participates in multiple relations. It allows different loss functions for each matrix approximation, and combines the information from multiple relations using weights which reflect the relative importance for each relation. This essentially generalizes the idea of Linked Matrix Factorization. The Latent Factors Model proposed in this paper uses the parameter λ to determine the contribution of the friendship network \mathbf{S} in generating the user factors.

The above mentioned papers use optimization techniques based on the alternating least squares approach, we use SVD which efficiently solves the optimization problem posed by the latent factors model.

3.2 Co-evolution of social and affiliation networks

Researchers have studied the effects of friendship ties on affiliations in other contexts, like the growth and evolution of social networks[1], and spread of influence through a social network[7, 3]. They tend to model the dependence of a user joining a group on the number of friends the user already has in the group. Zheleva et al[13] proposed a unified model for the generation of social and affiliation networks, and observed that the social network is one of the factors that influences the evolution of affiliation network. Our idea that friendship network combined with the affiliation network can be exploited in making affiliation recommendations is inspired by this line of research.

4. EXPERIMENTAL EVALUATION

We first introduce the datasets on which we conduct our experiments, and describe the experiment setup. We then describe our methodology for comparing the performance of various algorithms at the affiliation recommendation task. Finally, we discuss the results of experiments which apply the algorithms based on both the graph proximity model and the latent factors model in performing affiliation recommendation.

4.1 Data

We use two popular online social networks *Orkut* and *Youtube*, both operated by Google, for our experiments. The

users of both the social networks explicitly identify themselves as belonging to some *communities* or *groups*. Thus, for each of the networks we have adjacency matrix \mathbf{A} that identifies the memberships of the users in the groups and adjacency matrix \mathbf{S} that identifies friendships among users. For our experiments, we used data gathered by Mislove et al [9]. We compare the predictive ability of the algorithms using sub-networks extracted from these large networks¹. The statistics for these networks are presented in Table 1.

Feature	Orkut	Youtube
N_u	9123	16575
N_g	75546	21326
Average number of groups per user	55.8	10.5
Minimum number of groups per user	4	4
Mode of number of groups per user	6	4
Average number of users per group	6.7	8.1
Mode of number of users per group	2	1
Minimum number of users per group	1	1
Average number of friends per user	46	11.7
Mode of number of friends per user	2	1

Table 1: Details about the datasets used in experiments.

4.2 Experiment setup

For every user u , let $\mathbf{E}_u = \{(u, v) \mid \mathbf{A}_{u,v} = 1\}$ denote the affiliations of u , as observed in a given affiliation network \mathbf{A} . Invariably in all the experiments, we set aside a subset of these affiliations $\mathbf{E}_u^{(t)} \subset \mathbf{E}_u$ as test data (we use $|\mathbf{E}_u^{(t)}| = 30\%|\mathbf{E}_u|$). The remaining affiliations $\mathbf{E}_u^{(tr)} = \mathbf{E}_u \setminus \mathbf{E}_u^{(t)}$ are used as training data for the recommendation algorithms.

All of our recommendation algorithms require “learning” parameters for some model of the affiliation process, and hence for the purposes of learning the parameters, we use a set of *validation* edges $\mathbf{E}_u^{(v)} \subset \mathbf{E}_u^{(tr)}$ (we use $|\mathbf{E}_u^{(v)}| = 30\%|\mathbf{E}_u^{(tr)}|$). During the validation process, we compare different model parameters based on the number of correct edges among $25N_u$ recommendations² made using the model.

4.2.1 Evaluation method

We now describe our methodology for evaluating the performance of an affiliation recommendation algorithm. We first introduce notions of interest, such as precision, sensitivity, specificity, ROC and AUC. We then describe the way in which we evaluate the performance of a recommendation algorithm based on its top 50 recommendations to the average user. We then demonstrate the importance of choosing the right evaluation method for the community recommendation task by showing that using a different, but less appropriate, evaluation strategy yields different results.

Three commonly used measures of quality of solutions in information retrieval and classification tasks are *Precision*, *Recall* or *Sensitivity* and *Specificity*. *Precision* measures the exactness or fidelity of the prediction while *Sensitivity* measures the completeness of the prediction. Suppose that the

¹We extracted a small connected component from the large network for the propose of experimentation.

²This is chosen because, in Section 4.2.1, we argue that a predictive model should be judged based on the quality of its top few recommendations.

recommendation algorithm makes n recommendations to a user. Then, *Precision* is defined as the ratio of the number of correctly identified positives (true positives) to n , and *Sensitivity* is the ratio of the number of correctly identified positives to the total number of positives i.e. $|\mathbf{E}_u^{(t)}|$. *Specificity*, on the other hand, measures the ability of the recommender to exclude uninteresting affiliations from the recommendations it makes. It is defined as the fraction of such “negative affiliations” correctly excluded from the recommendation. All three of these performance measures range from 0 to 1.

Often, one is interested in evaluating the performance of a recommendation algorithm not for a single value of the number of recommendations n , but for the entire range of n . For a given recommendation algorithm and a user, sensitivity is a non-decreasing function of n , while specificity is a monotonically non-increasing function of n . The relationship between the increase in sensitivity, as n increases, with the decrease in specificity is of interest in comparing the quality of recommendations. For a good recommendation, as n increases, sensitivity increases without a big drop in specificity. The Receiver Operating Characteristics (ROC) curve, which is a plot of sensitivity vs (1 - specificity) for all values of n , is a common way of comparing the performance of classification algorithms over the entire range of n (or equivalently cutoff scores). Area under the ROC curve, or AUC, is then used as a way to compare different classification algorithms: the greater the AUC, the better the algorithm’s sensitivity vs 1-specificity tradeoff.

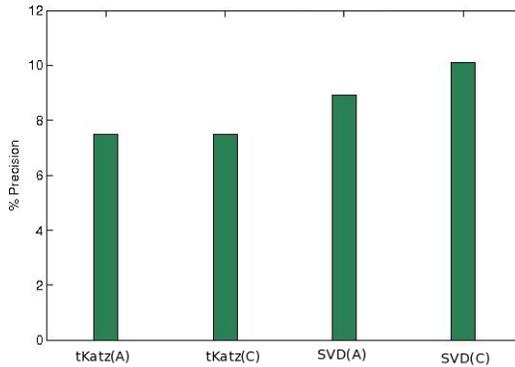
Consider a social network website, like Orkut or Facebook; or a vendor like Netflix which sells movies. They would be interested in making, let us say, five pages of recommendations to their users, but not much more than that: certainly not a hundred. Also, irrespective of whether a user participates in five communities or seventy, the social networking website would probably want to make roughly the same number of recommendations per user. So, we choose to evaluate the recommendation algorithms we propose based on their top 50 recommendations. We do this by examining a slice of the ROC curve formed by measuring the sensitivity and specificity the recommendation algorithm achieves for an average user at regular intervals between $n = 1$ and $n = 50$. To do this, for a given n , we compute the sensitivity and specificity for every user in the network, and take the mean of these values to be the average sensitivity and average specificity. We then plot the average sensitivity vs 1 - average specificity curve, as in Figure 2. Note that comparisons made using this method are statistically robust, as the sensitivities and specificities of recommendation algorithms are averaged over, for example, 9500 users in Orkut and 16000 users in Youtube.

Let $k(n_u)$ be the number of “good recommendations” made by a recommendation algorithm for a user u , when it makes n_u recommendations to that user. Then, the above “per user” sensitivity measure calculates $N_u^{-1} \sum_u \frac{k(n_u)}{|\mathbf{E}_u^{(t)}|}$. In our experiments, we will use $\forall u : n_u = 50$.

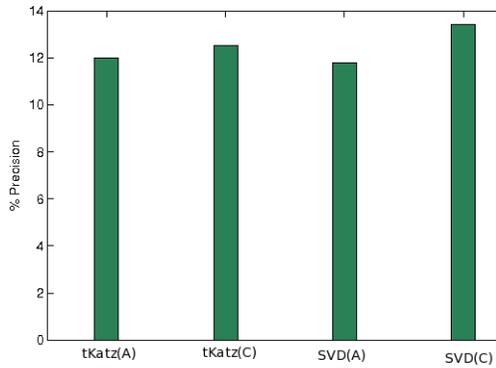
Contrast this with finding the “global” sensitivity $\frac{k'(n)}{\sum_u |\mathbf{E}_u^{(t)}|}$, where $k'(n)$ denotes the number of “good recommendations” made by a given recommendation algorithm while making n predictions in total. For a fixed n , this “global” sensi-

tivity is proportional to precision, and is a commonly used measure of performance of link prediction algorithms in the context of social network analysis. Note that, in this case, while $n = \sum_u n_u$, there is no guarantee that, for two given users u and v , $n_u = n_v$; indeed the recommendation algorithm, when asked to make n recommendations, may not make any recommendations at all for some users. Therefore, this measure of goodness of a recommendation algorithm is not equivalent to the “per user” sensitivity described earlier.

Also, judging identical algorithms on identical datasets, using this alternate evaluation method can yield very different rankings of recommendation methods, as illustrated by comparing Figure 1 with Figure 2. Hence, the choice of an appropriate method for evaluating affiliation recommendations is an important one.



(a) Orkut dataset

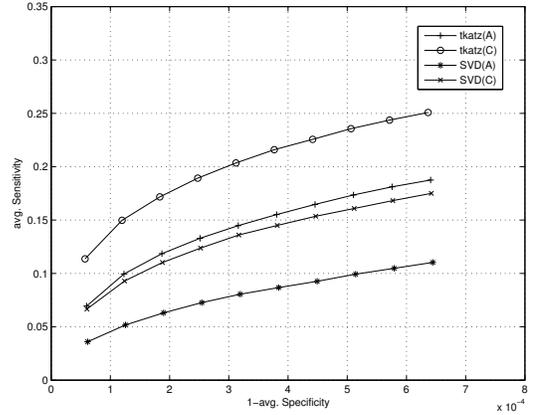


(b) Youtube dataset

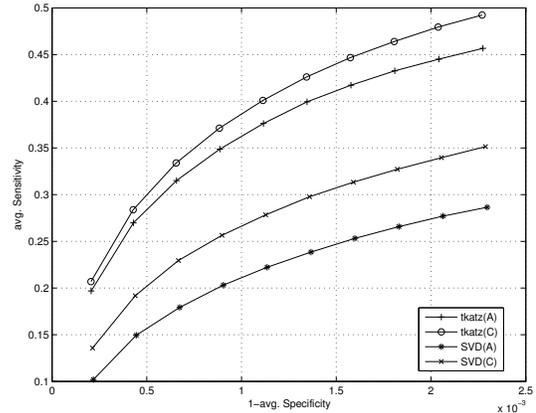
Figure 1: Comparison of recommendation algorithms using “global sensitivity” yields results different from Figure 2 while making $\sum_u E_u$ recommendations in total (See Section 4.2.1).

4.3 Results and Discussion

In this section, we report and analyse the performance of the various recommendation algorithms, based on the graph proximity model and latent factors model discussed in Section 2. We compare the performance of the graph proximity based methods with latent factors based methods based on the average sensitivity and average specificity metrics intro-



(a) Orkut dataset



(b) Youtube dataset

Figure 2: Comparison of recommendation algorithms based on graph proximity and latent factors models, as described in Section 4.2.1. The leading slice of the ROC curve is shown. The graph proximity based predictors consistently outperform latent factors based predictors in the two datasets. See Section 4.3 for discussion.

duced earlier, for a given number of recommendations in the range [5,50] (in steps of 5).

The results are reported in Figure 2. Consider the performance of the recommendation algorithms on the Orkut dataset in Figure 2 (a). SVD(A) gives the lowest performance of all the methods. SVD(C) performs better than SVD(A), which is expected given that it uses information from the friendship network \mathbf{S} in addition to the information from affiliation network \mathbf{A} . For the average user, the graph proximity model based methods significantly outperform latent factors based methods as observed from the figure. In particular, we see that tKatz(C) performs much better than tKatz(A), which in turn outperforms latent factor based methods. We see that the information in the friendship network indeed proves highly beneficial to making affiliation recommendations and graph proximity based methods exploit this information the most.

Another interesting comparison of latent factor methods based on the choice of \mathbf{D} in constructing the combined net-

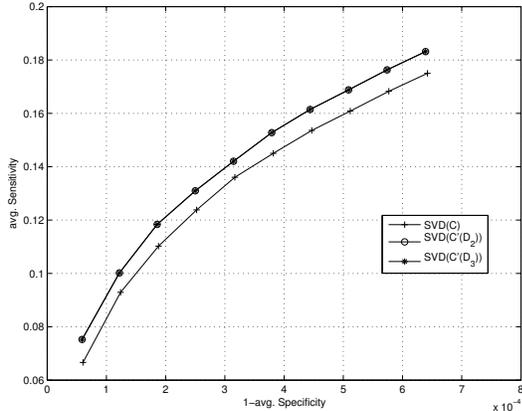


Figure 3: Comparison of latent factors based algorithms for various choices of \mathbf{D} , for the Orkut dataset. See Section 4.3 for discussion.

work \mathbf{C}' given in (3), is presented in Figure 3. We observe that the choice of \mathbf{D} does not appear to make any significant difference in the performance of the recommendation algorithm. In the plot, \mathbf{D}_2 denotes using $\mathbf{A}^T\mathbf{A}$ for \mathbf{D} in \mathbf{C}' while \mathbf{D}_3 denotes using $\lambda\mathbf{A}^T\mathbf{A}$, where λ is also the weight associated with \mathbf{S} in the combined graph. Though, in case of the Orkut dataset, we see that $\text{SVD}(\mathbf{C}', \mathbf{D})$ performs slightly better than $\text{SVD}(\mathbf{C})$, the choice of $\mathbf{D} \neq 0$ does not affect the performance. In case of Youtube (plots not shown), our experiments indicate that \mathbf{D} is not useful at all. The potential choices for \mathbf{D} do not perform any better than $\mathbf{D} = 0$.

The summary of performances of the algorithms on the Youtube dataset is shown in Figure 2 (b). We observe that the case for Youtube is similar to that of Orkut, in that graph proximity based algorithms significantly outperform latent factors based algorithms. In particular, $\text{tKatz}(\mathbf{C})$ is highly successful compared to the other methods.

The best parameters learnt by the various algorithms are presented in Table 2. Note that the best parameter $\beta = 10^{-12}$ implies that the calculated tKatz measure was effectively using the common neighbors method. In other words, users and communities connected by path lengths 5 or more are not useful in making affiliation recommendations.

Algorithm	Orkut	Youtube
$\text{SVD}(\mathbf{C})$	$d = 50, \lambda = 0.8$	$d = 90, \lambda = 1$
$\text{SVD}(\mathbf{C}'(\lambda\mathbf{A}^T\mathbf{A}))$	$d = 60, \lambda = 0.6$	$d = 70, \lambda = 1$
$\text{tKatz}(\mathbf{A})$	$\beta = 10^{-12}$	$\beta = 10^{-12}$
$\text{tKatz}(\mathbf{C})$	$\beta = 0.01, \lambda = 0.2$	$\beta = 0.1, \lambda = 0.4$

Table 2: Best parameters learned by the recommendation algorithms using validation.

We see that the recommendation algorithms perform consistently across the two datasets, and the evaluations are robust as the specificities and sensitivities are averaged over 9000 users in Orkut and 16000 users in Youtube.

5. CONCLUSION AND FUTURE WORK

In this paper, we have tackled the affiliation recommendation problem, where the task is to recommend new affilia-

tions to a given user, given the current state of the friendship and affiliation networks. We show that information from the friendship network can indeed be fruitfully exploited in making affiliation recommendations. This auxiliary source of information was hitherto not used in making community recommendations.

Using a simple way of combining these networks, we suggested two ways of modeling the networks for the purpose of making affiliation recommendations (Section 2). The first of these approached the problem from the graph proximity viewpoint, whereas the second modelled the interactions of users and groups in the two networks using latent factors derived from optimizing towards a joint matrix factorization objective. We studied the algorithms suggested by these models on real world networks (Section 4). We motivated and proposed a way of evaluating recommenders, by measuring how good the top 50 recommendations are, and demonstrated the importance of choosing the right evaluation strategy. Algorithms suggested by the graph proximity model turn out to be the most effective, based on experiments on large real world data-sets. These results show that the application of techniques from social network link prediction in affiliation and item recommendation is a promising one.

5.1 Future Work

There is the intriguing possibility of using an affiliation network for link prediction in the friendship network. Discovering techniques and models which do this effectively seems to be a challenging research avenue. Our early experiments at doing this indicate that this is a much harder problem. The reasons for this are not yet clear, and this question seems fertile for further exploration.

Within the ambit of the affiliation recommendation problem itself, one may research the ways of fruitfully using even more sources of information. For example, Chen et al[5] use information from textual description of communities along with the affiliation network to make affiliation recommendations. It might be useful to consider the social network together with this auxiliary information. Also, predictors based on latent factors model and the graph proximity model may be suited for different types of users, and creating a meta-predictor which combines predictions from both classes of predictors is another attractive research direction. Scalability is a challenge in using predictors based on the graph proximity models on massive datasets - both in terms of memory and in terms of computational cost[10], and developing efficient predictors based on the graph proximity model will be part of future work.

6. ACKNOWLEDGEMENTS

We thank Prateek Jain and Berkant Savas for helpful discussions. We also thank Alan Mislove [9] for providing access to Orkut and Youtube datasets. Zhengdong Lu is supported by an ICES postdoctoral fellowship at UT Austin. This research was supported by NSF grants CCF-0916309 and IIS-0713142.

7. REFERENCES

- [1] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In

- KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 44–54, New York, NY, USA, 2006. ACM.
- [2] Brett W. Bader and Peter A. Chew. Enhancing multilingual latent semantic analysis with term alignment information. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 49–56, Morristown, NJ, USA, 2008. Association for Computational Linguistics.
- [3] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 199–208, New York, NY, USA, 2009. ACM.
- [4] Wen-Yen Chen, Jon-Chyuan Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y. Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. In *WWW '09: Proceedings of the 18th International Conference on World wide web*, pages 681–690, New York, NY, USA, 2009. ACM.
- [5] Wen-Yen Chen, Dong Zhang, and Edward Y. Chang. Combinational collaborative filtering for personalized community recommendation. In *KDD '08: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 115–123, New York, NY, USA, 2008. ACM.
- [6] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual International ACM SIGIR Conference on Research and development in Information Retrieval*, pages 50–57, New York, NY, USA, 1999. ACM.
- [7] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, New York, NY, USA, 2003. ACM.
- [8] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 556–559, New York, NY, USA, 2003. ACM.
- [9] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pages 29–42, New York, NY, USA, 2007. ACM.
- [10] B. Savas and I. S. Dhillon. Fast and accurate low rank approximation of massive graphs. Technical report TR–10–18, Department of Computer Science, The University of Texas at Austin, 2010.
- [11] Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. In *KDD '08: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 650–658, New York, NY, USA, 2008. ACM.
- [12] Wei Tang, Zhengdong Lu, and Inderjit S. Dhillon. Clustering with multiple graphs. In *ICDM '09: Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, pages 1016–1021, Washington, DC, USA, 2009. IEEE Computer Society.
- [13] Elena Zheleva, Hossam Sharara, and Lise Getoor. Co-evolution of social and affiliation networks. In *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1007–1016, New York, NY, USA, 2009. ACM.