Tumblr Blog Recommendation with Boosted Inductive Matrix Completion

Donghyuk Shin
Dept. of Computer Science
University of Texas at Austin
dshin@cs.utexas.edu

Suleyman Cetintas
Advertising Sciences Group
Yahoo Labs, Sunnyvale, CA
cetintas@yahoo-inc.com

Kuang-Chih Lee
Advertising Sciences Group
Yahoo Labs, Sunnyvale, CA
kclee@yahoo-inc.com

Inderjit S. Dhillon
Dept. of Computer Science
University of Texas at Austin
inderjit@cs.utexas.edu

ABSTRACT

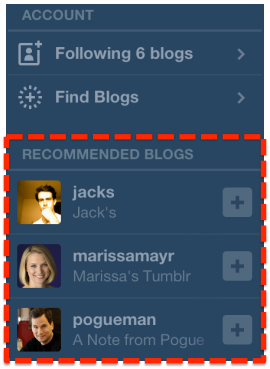
Popular microblogging sites such as Tumblr have attracted hundreds of millions of users as a content sharing platform, where users can create rich content in the form of posts that are shared with other users who follow them. Due to the sheer amount of posts created on such services, an important task is to make quality recommendations of blogs for users to follow. Apart from traditional recommender system settings where the follower graph is the main data source, additional side-information of users and blogs such as user activity (e.g., like and reblog) and rich content (e.g., text and images) are also available to be exploited for enhanced recommendation performance. In this paper, we propose a novel boosted inductive matrix completion method (BIMC) for blog recommendation. BIMC is an additive low-rank model for user-blog preferences consisting of two components; one component captures the low-rank structure of follow relationships and the other captures the latent structure using side-information. Our model formulation combines the power of the recently proposed inductive matrix completion (IMC) model (for side-information) together with a standard matrix completion (MC) model (for low-rank structure). Furthermore, we utilize recently developed deep learning techniques to obtain semantically rich feature representations of text and images that are incorporated in BIMC. Experiments on a large-scale real-world dataset from Tumblr illustrate the effectiveness of the proposed BIMC method.

Categories and Subject Descriptors

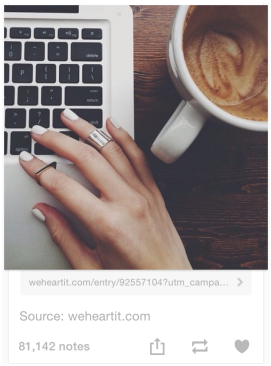
H.2.8 [**Database Management**]: Database Applications—*Data Mining*; H.3.3 [**Information Search and Retrieval**]: Information Filtering

CIKM'15, October 19–23, 2015, Melbourne, VIC, Australia.

DOI: http://dx.doi.org/10.1145/2806416.2806578.



(a) Blog recommendation module in Tumblr



(b) Example post with high note count.

Figure 1: The blog recommendation module (a) and an example post (b) with high note count (i.e., like and reblog count) in Tumblr.

Keywords

Blog Recommendation, Inductive Matrix Completion, Deep Learning Features

1. INTRODUCTION

Microblogging services have emerged as a leading content sharing and communication platform combining both traditional blogging and social networking characteristics. Tumblr[1] is one of the most popular microblogging services with more than 230 million users, where users can create and share posts with the followers of their blogs. Conversely, users consume shared content by following blogs of interest, which has become an overwhelming task due to the sheer number of options. Thus, one of the core problems in microblogging sites is predicting whether a user will follow a blog or not. Improved blog recommendations would not only lead to higher user engagement by assisting users to discover interesting content, but also attract more appealing followers for sponsored or advertisers blogs. Figure 1(a) shows the blog recommendation module in Tumblr.

The problem of recommending blogs differs from traditional collaborative filtering settings, such as the Netflix rating prediction problem [3], in two main aspects. First, in-

[1] www.tumblr.com

teractions between users and blogs are *binary* in the form of follows and there is no explicit rating information available about user preferences. The "follow" information can be represented as a unidirectional unweighted graph and popular proximity measures based on the structural properties of the graph can then be applied to the problem [36]. Secondly, an important but beneficial difference is that blog recommendation inherently entails rich *side information* in addition to the conventional user-item matrix (i.e., follower graph). There are two main categories of side information: (1) *user generated content* such as images, tags and text (e.g., Figure 1(b)) and (2) *user activity* including likes and reblogs. In the case of Tumblr, incorporating image features is crucial as majority of posts contain photos. Text data is also rich in Tumblr, since posts have no limitation in length, compared to other microblogging sites such as Twitter[2]. While such user generated content characterizes various blogs, user activity is a more direct and informative signal of user preference as users can explicitly express their interests by liking and reblogging a post. This implies that users who liked or reblogged the same posts are likely to follow similar blogs. In fact, as shown in many existing studies, such side information not only improves recommendation quality, but also alleviates sparsity issues in the user-item matrix [26, 14, 31].

On the other hand, rigorous approaches for incorporating side-information in a recommender system setting are lacking. Consider the standard matrix completion (MC), one of the most widely used and theoretically well-studied method for recommendation tasks, for which there have been several rigorous guarantees established in the recent past [20, 5, 17, 7]. However, MC is exposed to data sparsity issues and restricted to the transductive setting, i.e., predictions can only be made for existing users/items, as it only considers observations from the user-item matrix. More recently, the inductive matrix completion (IMC) was proposed and theoretically analyzed by [16] motivated by settings where side information of users/items is available in the form of feature vectors. However, IMC assumes that observed entries are fully explained by such features, which is not always the case especially with noisy features that do not support the user-item matrix. Furthermore, IMC cannot make meaningful recommendations for users or items without any features, which is often the case in Tumblr (see Section 3).

To this end, we propose a novel Boosted Inductive Matrix Completion (BIMC) model for blog recommendation that combines the power of an inductive matrix completion model together with a standard matrix completion model via boosting. Specifically, BIMC first applies the MC model to smooth the input matrix and reduce the noise level by low-rank approximation, and then further models the residual of the approximation with the IMC model. That is, BIMC captures both the low-rank structure of follow relationships as well as the latent structure using side-information of users and items in an additive manner capturing entries in the follower graph where MC fails to learn.

By incorporating user/blog features, BIMC is also capable of making recommendations in the *inductive* setting, i.e., make predictions for users or blogs *not seen at training time*, which includes cold-start cases[3]. This is particularly important for Tumblr as users and blogs often have very few or no links in the follower graph as shown in Section 3. Experiments on large-scale real-world proprietary data from Tumblr show that our proposed BIMC significantly outperforms MC, IMC and several other standard methods for the blog recommendation task.

Lastly, an important issue is how to effectively represent the three side-information sources (image, text and activity) as features. Recently, deep learning approaches have emerged as a powerful class of models that understand semantic content of images, giving state-of-the-art performance on image recognition tasks [24, 9, 19]. This is also the case for text data, where vectorial representations of words capturing semantic relations between them are learned from neural networks [27, 35]. Encouraged by these results, we employ deep learning features for both images and tags/text as a useful and robust representation of users and blogs. For activity features, we represent likes and reblogs as a weighted graph similar to the follower graph and we compute principal components of the activity graph as features. To our knowledge, we are the first to consider image as well as activity features. Furthermore, adopting deep learned features for recommender systems is still unexplored.

Our contributions are summarized as follows:

- We propose a Boosted Inductive Matrix Completion based blog recommendation system that combines the power of an inductive matrix completion model together with a standard matrix completion model.
- We represent users and blogs with an extensive set of side information sources such as the user activity, text/tags, and images; and extract a comprehensive set of features using state-of-the-art deep learning methods.
- We show that the proposed BIMC model effectively combines heterogeneous user and blog features from multiple sources for more accurate recommendations.
- We conduct extensive experiments as well as detailed analysis on large-scale real-world data from Tumblr, and demonstrate the superiority of the proposed BIMC method over several state-of-the-art baselines.

The rest of the paper is organized as follows. In Section 2, we review some closely related work. Then we analyze the Tumblr data and study some of its important characteristics in Section 3. Next we present our proposed blog recommendation method in Section 4 and give details of user and blog feature extraction in Section 5. Experimental results are given in Section 6 followed by conclusions in Section 7.

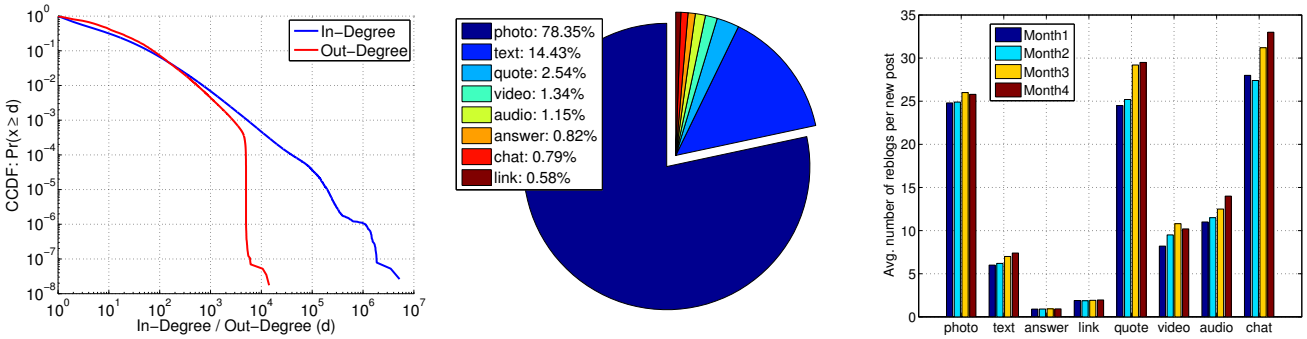
2. RELATED WORK

In general, various sources of information additional to the traditional user-item matrix can boost recommendation performance. Recommender systems with side information is by no means new and numerous methods have been proposed based on the type of side information they utilize, such as user generated content [22, 38, 13, 28, 25, 2], user/item profile or attribute [1, 4], social network [18, 26] and context information [29]. A recent comprehensive survey of the state-of-the-art methods can be found in [31].

One of the main approaches that extend MC with side information is the Collective Matrix Completion (CMC) model [33, 4], where the goal is to jointly recover a collection of matrices with shared low-rank structure. In [38], the user-item matrix and the user-user similarity matrix based on tags information are jointly factorized to facilitate better recommendations. Recent work on CMC provides consis-

[2] `www.twitter.com`; posts (or tweets) are restricted to 140 characters.

[3] Cold-start refers to users or items without any known entries in the user-item matrix.



(a) In/out degree distribution of the follower graph.

(b) Distribution of different post types.

(c) Average number of reblogs per new post for each post type.

Figure 2: Statistics of Tumblr data.

tency guarantees under certain assumptions [11], which can be restrictive due to imposing a common structure. Recommender systems with social networks are mostly based on the latent factor model with additional constraints in the objective such as latent factors being similar between connected users [18, 26]. Another approach is the regression-based latent factor model proposed by [1], where attribute information is integrated into the model. However, the proposed method does not scale well to large datasets. Graph-based methods have also been extended to incorporate side information. For example, [22] constructs a multi-partite graph with social and tag information, which does not scale well with additional side-information or when features are represented as a dense matrix. Lastly, user generated content, such as reviews and comments, have been exploited by analyzing sentiment information [28, 25]. In most cases, methods are either specialized for a particular source of information or do not scale well with large number of features and lack theoretical guarantees.

Closely related to Tumblr blog recommendation is the Who-To-Follow system in Twitter [12]. Previous approaches for followee recommendation include a probabilistic model based on probabilistic latent semantic analysis proposed by [21]. In [37], a community-based approach is proposed, where matrix factorization is applied independently to each of the discovered communities. However, both methods do not consider any other explicit user/blog features. In [13], follower/followee as well as content (tweets) information is used to represent users in a similarity-based collaborative filtering method. Similarly, [2] first identifies a list of candidate followees, which are the 2-hop neighbors in the follower graph, and then refines the list using content-based profiles of users. Graph-based methods that use proximity measures between nodes have also been applied to followee recommendation [36]. One major drawback is that these methods can not efficiently deal with the inductive setting. Furthermore, none of the existing methods consider images nor user activity information, which is also available in Twitter.

There has been limited work on employing deep learning methods for recommender systems. One exception is the music recommendation method proposed by [30]. In [30], the traditional matrix factorization is combined with a deep convolutional neural network to learn a function that maps music content features to corresponding latent factors. Another exception is the work by [10], where a recurrent neural network is trained to capture semantics of text documents that is used in a content-based recommender system. Both studies have shown deep learning as a promising approach for recommender systems.

3. DATASET CHARACTERISTICS

In this section, we analyze some important characteristics of different aspects of the Tumblr data[4]. As a social network service, Tumblr users can follow blogs of interest without mutual confirmation similar to Twitter, but different from Facebook[5]. The follow information can be represented as a directed bipartite graph where nodes correspond to users and blogs and an edge from node i to j represents user i following blog j. We use a snapshot of the follower graph sampled from June 2014, which consists of 76.86 million nodes with 2.27 billion edges. We find similar characteristics as in [6] including the in/out degree distributions shown in Figure 2(a). The in-degree follows a power-law distribution, while the out-degree does not and shows a sharp drop when the out-degree is around 5,000, which is the maximum number of blogs a user can follow in Tumblr. About 50% of nodes are without any followers (i.e., 0 in-degree) and the maximum in-degree is 5.22 million, while about 25% of nodes are not following any blogs (i.e., 0 out-degree) and the maximum out-degree is 14,208.

As a microblogging platform, Tumblr provides useful tools close to that of traditional blogging sites for creating longer, richer and higher quality content. Specifically, it allows users to create 8 different types of posts: *photo*, *text*, *answer*, *link*, *quote*, *video*, *audio* and *chat*. Furthermore, posts in Tumblr have no limitation in length unlike other microblogging sites such as Twitter, which is restricted to 140 characters per post. It also supports the use of *tags* for each post, which are separate from the post content. Lastly, users can *like* a post or re-broadcast the post to its own followers by *reblogging*. While these two activities have different intentions to the user, both directly reflect the user's interest which should be utilized for better recommendation quality.

We processed 5 months of Tumblr data, where each month contains about 1.5 TB of sampled records of posts created, reblogged and liked. Note that we restrict to users with at least 5 records in each month. On average, there are more than 150 million newly created posts, 2.5 billion reblogged posts and 2 billion likes per month. We show the

[4]The reported datasets and results are deliberately incomplete and subject to anonymization, and thus do not necessarily reflect the real portfolio at any particular time.

[5]www.facebook.com

distribution of each post type in Figure 2(b). Almost 80% of posts are *photo* posts suggesting image features are a crucial component for analyzing posts. Figure 2(c) reports the average number of reblogs a new post gets for each post type. We can see in the figure that photo, quote and chat posts are reblogged significantly more than other types of posts. Overall, a new post gets reblogged more than 15 times on average illustrating the high sharing activity in Tumblr. We have also found that about 8.3% of users do not have any posts and about 12.2% of users do not have any activity information. More detailed analysis of the Tumblr data can be found in [6].

4. METHODS

In this section, we describe a natural way of combining various user/blog features and the follower graph to enable the inductive setting, i.e., recommendations for new users and blogs. We first describe the Inductive Matrix Completion method for blog recommendation, which is based on the proposition that user-blog follow behavior arises from applying a low-rank matrix to user and blog features. Next we motivate and present our proposed Boosted Inductive Matrix Completion method. We briefly establish the notation used before describing our proposed approaches.

Notation: We denote the follower graph by $\mathcal{G} = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$, where $\mathcal{V}_1$ ($m = |\mathcal{V}_1|$) and $\mathcal{V}_2$ ($n = |\mathcal{V}_2|$) is the set of users and blogs, respectively; $\mathcal{E} = \{e_{ij} | i \in \mathcal{V}_1, j \in \mathcal{V}_2\}$ is the set of edges indicating user i follows blog j. Let $A \in \mathbb{R}^{m \times n}$ be the adjacency matrix of $\mathcal{G}$, where each row corresponds to a user and each column corresponds to a blog, such that $A_{ij} = 1$, if user i is following blog j and 0 otherwise. That is, we treat missing values as zeros. Note that $\mathcal{G}$ is a directed graph, i.e., A is non-symmetric. Let $X \in \mathbb{R}^{m \times f_u}$ and $Y \in \mathbb{R}^{n \times f_b}$ denote the user and blog feature matrices, respectively.

4.1 Matrix Completion

The low rank matrix completion (MC) approach is one of the most popular and successful collaborative filtering methods for recommender systems [23]. The goal is to recover the underlying low rank matrix by using the observed entries of A, which is typically formulated as follows:

$$\min_{U,V} \sum_{(i,j)\in\Omega} (A_{ij} - (UV^T)_{ij})^2 + \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2), \qquad (1)$$

where $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$ with r being the dimension of the latent feature space; $\Omega \in [m] \times [n]$ is the set of observed entries; λ is a regularization parameter. Note that matrix completion only utilizes samples from the follower graph A and ignores the side information that might be present in the system.

4.2 Inductive Matrix Completion

The standard matrix completion formulation is restricted to the transductive setting, i.e., predictions can only be made for existing users and items without re-training for latent factors of new users or items. Furthermore, the standard formulation suffers performance with extreme sparsity in the data, which is the case for Tumblr as about 50% of users do not have any followers and about 25% of users are not following any blogs. One simple way to make predictions for such users is to use a popularity based global ranking of blogs and recommend the top ranked ones. In order to make meaningful predictions, one would need more information about users and blogs. For Tumblr, such information can be obtained from rich content (photos, text) and activity (reblog, like) information.

Recently, a novel inductive matrix completion (IMC) approach was proposed and theoretically analyzed by [16] to alleviate data sparsity issues as well as enable predictions for new users and items by incorporating side information of users and items given in the form of feature vectors. The main idea is to model A_{ij} using user i's feature vector $\mathbf{x}_i \in \mathbb{R}^{f_u}$, item j's feature vector $\mathbf{y}_j \in \mathbb{R}^{f_b}$ and a low-rank matrix $Z \in \mathbb{R}^{f_u \times f_b}$ as

$$A_{ij} = \mathbf{x}_i^T Z \mathbf{y}_j. \qquad (2)$$

That is, the interaction between user i and item j is generated by applying their respective feature vectors to Z. For a new item b, the predictions A_{ib} for each user i can be calculated with the feature vector $\mathbf{y}_b$ available.

By factoring $Z = WH^T$, the goal of IMC is to recover $W \in \mathbb{R}^{f_u \times r}$ and $H \in \mathbb{R}^{f_b \times r}$ using the observed entries in A. The IMC objective is given as

$$\min_{W,H} \sum_{(i,j)\in\Omega} \ell(A_{ij}, \mathbf{x}_i^T W H^T \mathbf{y}_j) + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2),$$

for some loss function ℓ that measures the difference between the observations and predictions, e.g., squared loss $\ell_s(a,b) = (a-b)^2$ or logistic loss $\ell_l(a,b) = \log(1+e^{-ab})$. The number of parameters to learn is $(f_u + f_b) \times r$ depending only on the number of user and item features, whereas there are $(m+n) \times r$ parameters in the standard matrix completion. Note that matrix completion is a special case of IMC when $X = I$ and $Y = I$.

For a convex loss function ℓ, the above IMC objective becomes a convex function when either W or H is fixed (similar to the standard matrix completion case). The computational cost to solve the optimization problem differs based on the choice of the loss function ℓ. In our experiments, we use the squared loss in the objective and employ the alternative minimization approach in [15]. Under this setting, the computational cost for each step is $O((nnz(A) + mf_u + nf_b)r^2 c)$, where $nnz(A)$ is the number of non-zeros in A and c is a small constant. In our experiments, f_u, f_b and r are very small (few hundreds) and the solution converges in less than 10 iterations.

4.3 Boosted Inductive Matrix Completion

Next we present our method called Boosted Inductive Matrix Completion (BIMC). One issue with IMC is that the model is too rigid as it heavily depends on the user features X and item features Y. That is, user and item features from different sources should support the underlying structure of the follower graph A in order to make good predictions. Let $X = U_X \Sigma_X V_X^T$, where $U_X \Sigma_X V_X^T$ is the SVD of X. Similarly, let $Y = U_Y \Sigma_Y V_Y^T$ be the SVD of Y. From the IMC formulation, we have

$$A = XZY^T = U_X(\Sigma_X V_X^T Z V_Y \Sigma_Y) U_Y^T = U_X \hat{Z} U_Y^T,$$

where $\hat{Z} = \Sigma_X V_X^T Z V_Y \Sigma_Y$. Thus, the subspace spanned by U_X must have significant overlap with that of A to achieve small error. For example, like and reblog activity features can be quite helpful as a direct reflection of user interest. Similar arguments can be made for Y as well.

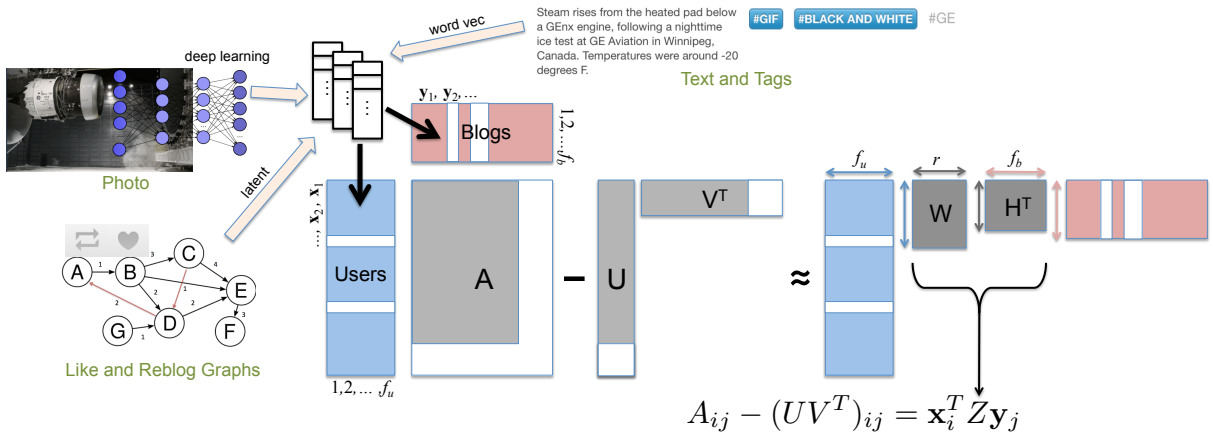


Figure 3: Boosted Inductive Matrix Completion model. Shaded areas represent available information.

However, features from various sources may not always support the matrix A and IMC can suffer performance significantly in such case. For instance, text data in Tumblr is extremely sparse and noisy, and thus may not directly reveal user preference. Moreover, it is not always the case that all users and items have features (as shown in Section 3), in which IMC would not be able to make any predictions.

To address these problems, we propose to combine both standard matrix completion and inductive matrix completion, and thereby better utilize the power of both approaches. That is, we combine the power of MC to reduce the noise level in the input data as well as the advantage of IMC to incorporate side information of users and items. Our idea is to model A_{ij} as

$$A_{ij} = (UV^T)_{ij} + \alpha \mathbf{x}_i^T Z \mathbf{y}_j, \tag{3}$$

where the parameter α adjusts the contribution of features in the final prediction. Choosing a good α is crucial for both performance and solving the optimization problem, which can be difficult to tune. Furthermore, simultaneously solving for all four latent factor matrices U, V, W and H will lead to slower convergence due to the increased number of parameters.

Thus, our strategy is to first learn the latent factor matrices U and V of the MC model as in (1). The resulting approximation error or residual matrix $R = A - UV^T$ represents links in the follower graph that MC could not fully capture. Then we model R_{ij} with IMC as

$$R_{ij} = A_{ij} - (UV^T)_{ij} = \mathbf{x}_i^T Z \mathbf{y}_j. \tag{4}$$

In other words, we first try to find the support of the follower graph A with the latent factors U and V and focus on the part that it can not accurately model using IMC. This is especially useful when the norm of the residual from (1) is large, which suggests a significant deviation from low rank structure in A. Our objective is

$$\min_{W,H} \sum_{(i,j)\in\Omega} \ell(A_{ij} - (UV^T)_{ij}, \mathbf{x}_i^T W H^T \mathbf{y}_j) + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2).$$

With ℓ being the squared loss and fixing H, the gradient of the above objective in matrix form is given as

$$X^T(A - UV^T - XWH^TY)Y^TH + \lambda W.$$

Note that we do not have to explicitly form $A - UV^T$, which would be a dense matrix and infeasible to store in memory.

Our approach eliminates the need to fine tune the parameter α. Furthermore, existing efficient solvers for MC and IMC with theoretical guarantees [17, 15] can be directly applied. An overview of our proposed BIMC model can be found in Figure 3, where we can see that BIMC can handle both sparsity in A as well as users/items without features. Given a user i with features $\mathbf{x}_i$ and blog j with features $\mathbf{y}_j$, we use (3) to obtain predictions with the learned factors and α set to 1. Finally, we note that a converse approach can also be used, i.e., learn the IMC model first, and then train the MC model on the resulting residual matrix; we found the results to be comparable, so we only present results for the former.

We illustrate the advantages of BIMC compared to IMC when features are noisy with the following experiment on the MovieLens-100K dataset[6]. We compute the rank-20 SVD of the user-movie matrix $A = U_A \Sigma_A V_A^T$ and set $X = U_A$ and $Y = V_A$, i.e., the left and right singular vectors of A to be the user and movie features, respectively. Then we perturb the columns of X by adding noise and measure the relative approximation error for IMC: $\|A - XWH^TY^T\|_F / \|A\|_F$ and for BIMC: $\|A - UV^T - XWH^TY^T\|_F / \|A\|_F$. Results are given in Figure 4 confirming that BIMC achieves lower approximation error rates than IMC due to modeling the residual matrix R as in (4). This shows that BIMC is robust to noisy features whereas IMC suffers performance as the noise level increases in the user features X.

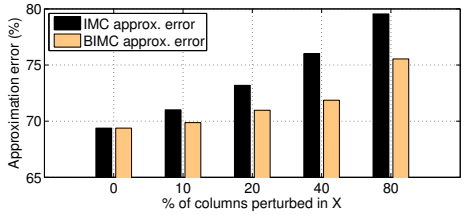


Figure 4: Relative approximation error for IMC and BIMC with different levels of noise in features.

5. FEATURE EXTRACTION

As described in Section 3, Tumblr data contains three main sources of side information of users and blogs: (1) likes and reblogs, (2) tags and text, and (3) images. Extracting

[6] 100,000 ratings from 1,000 users on 1,700 movies; `http://grouplens.org/datasets/movielens/`

useful features from these sources is a crucial step of the recommender system. We discuss the details of how we extract these user and blog features as follows.

User Activity: For user activity information, we use likes and reblogs from Tumblr of the 1 million sampled users and blogs. Both like and reblog activities can be represented as a weighted graph similar to the follower graph A, where the edge weight between a user i and blog j is set to be the number of liked and reblogged posts of blog j by user i. The edge weights in both cases follow a power-law distribution. Each user likes or reblogs about 340 posts from 27 blogs and each blog gets a total of 410 likes or reblogs on average. In the experiments, we aggregate like and reblog graphs to a single activity graph from the training data and use a log-scale of the edge weights. One way to obtain useful and robust features is to consider the principal components of the adjacency matrix corresponding to the activity graph. That is, we compute p principal components and use them as latent user and blog features for IMC. Thus, we have $f_u = f_b = p$ user activity features for users and blogs, where we empirically set $p = 500$ in the experiments.

Tags and Text: Tags and text used in posts of Tumblr are extremely sparse and noisy. There are an average of 28.7 words and 4.8 tags per post. Furthermore, tags are unconstrained in Tumblr, where a user can put in any arbitrary text. Existing models based on bag-of-words (e.g., LSA, LDA) can suffer from such issues. Therefore, we utilize word2vec, which is a recent neural network inspired method that learns word embeddings in the vector space [27]. word2vec utilizes the technique called skip-gram with negative samples, which tries to represent each of the words by a vector such that words in similar contexts are close to each other. This representation is accomplished by maximizing the predicted probability of words co-occurring in the training corpus. In our work, we first compute d-dimensional vector representations of each word using word2vec, and then cluster these words into c clusters by the k-means algorithm. Using the cluster information, we finally create a histogram of word clusters for each post as a compact representation of tags and text used in that blog. We set $d = 300$ and $c = 1,000$ and processed both textual features for each month in the training data.

Images: Images are an important part of Tumblr data as shown in Section 3. We randomly sampled about 800K images per month from blogs that appear in the training data. We trained a convolutional neural network (CNN) [24, 9] on 1.5M Flickr images with labels due to the unavailability of image labels for the Tumblr dataset. The CNN is composed of seven hidden layers, which consist of five successive convolutional layers followed by two fully connected layers, plus a final soft-max layer. The nonlinearity of each neuron in this CNN is modeled by Rectified Linear Units (ReLUs) $f(x) = \max(0, x)$, which accelerates learning compared with saturating nonlinearity such as tanh units. The CNN takes a 224×224 pixel RGB image as input. Each convolutional layer convolves the output of its previous layer with a set of learned kernels, followed by ReLU non-linearity, and two optional layers, local response normalization and max pooling. The local response normalization layer is applied across feature channels, and the max pooling layer is applied over neighboring neurons. The output of the 7th layer is fed into the last soft-max layer, which outputs confidence scores over the pre-defined 958 categories for a given input image. Using the neural network, we extracted deep learning features from the sampled Tumblr images. For users, we averaged the resulting feature vector over all images that the user posted, liked and reblogged. For blogs, only posted and reblogged images were considered as reblogged posts also become a post of the blog.

6. EXPERIMENTAL EVALUATION

In this section, we present the experimental setup used to evaluate our proposed method BIMC in comparison to IMC as well as several other baseline methods on the Tumblr dataset for blog recommendation with additional side information of users and blogs. From the Tumblr follower graph, we randomly sampled 1 million users and blogs resulting in about 12 million follows, i.e., nonzero elements in A. Both user activity and user generated content information were collected over a 5 month period from Tumblr post data.

6.1 Baselines and Evaluation Metrics

We perform both offline and temporal evaluations. For the offline evaluation, we use 10-fold cross-validation. Temporal evaluation is used to simulate online evaluations, where we use data from preceding 4 months as training and the remaining month as testing.

In both cases, we compare BIMC against the standard matrix completion formulation (MC) and the Singular Value Decomposition (SVD), which has been shown to perform well for top-N recommendation tasks [8]. We also compare with methods that incorporate side information including the inductive matrix completion (IMC). Another popular approach is the collective matrix completion (CMC) [33, 4]. The goal of CMC is to jointly recover a collection of matrices with shared low rank structure, which is different from IMC. Specifically, given user-item matrix A, user features X and item features Y, CMC finds a joint factorization as:

$$A = UV^T, \quad X = UP^T \text{ and } Y = VQ^T.$$

That is, the shared user latent factor matrix U is obtained from both A and X (similarly V is obtained from A and Y). Recent work by [11] provides consistency guarantees for CMC, thus we use the algorithm presented in [11]. The Katz measure [34, 32], which is one of the most successful proximity measures for link prediction, is also included in the comparison as a graph-based approach. We compute Katz scores[7] between users and blogs on the combined (symmetric) matrix $C = \begin{bmatrix} S_u & A \\ A^T & S_b \end{bmatrix}$, where S_u and S_b are similarity matrices between users and blogs, respectively, computed from their features. Lastly, we report results of using a simple global popularity ranking (Global) for recommendation as a baseline, where blogs are ranked by the number of followers. We use rank $r = 10$ for MC and SVD, rank $r = 100$ for CMC, IMC and BIMC, and set $\lambda = 0.1$ for all methods, which are determined using cross-validation.

We measure the recommendation performance using precision (PRC@10) and recall (RCL@10) at top-10 generated by each method, which is the region of practical interest for recommender systems. We also report the AUC (area under the ROC curve) of each method for completeness.

[7]The Katz measure is defined as $\sum_{i=1}^{\infty} \beta^i C^i$. We set $\beta = 10^{-6}$.

Table 1: Offline evaluation results of the proposed Boosted Inductive Matrix Completion method (BIMC) in comparison to several baselines.

Method	PRC@10	RCL@10	AUC
Global	1.03%	4.80%	0.8687
SVD	1.28%	5.10%	0.8530
MC	1.28%	5.07%	0.8515
Katz	1.90%	8.15%	0.9209
CMC	0.49%	2.41%	0.8996
IMC	2.93%	11.33%	0.9075
BIMC	**3.21%**	**12.28%**	**0.9221**

Table 2: Temporal evaluation results of the proposed Boosted Inductive Matrix Completion method (BIMC) in comparison to several baselines.

Method	PRC@10	RCL@10	AUC
Global	1.01%	4.70%	0.8626
SVD	1.24%	4.82%	0.8479
MC	1.19%	4.53%	0.8464
Katz	1.33%	5.69%	0.9125
CMC	0.46%	1.81%	0.8932
IMC	2.85%	10.38%	0.8953
BIMC	**3.12%**	**11.32%**	**0.9129**

6.2 Experimental Results

Results of the proposed BIMC method with user and blog features are shown in comparison to the baselines: Global, SVD, MC, Katz, CMC, and IMC for PRC@10, RCL@10 and AUC in Table 1 for the offline evaluation and Table 2 for the temporal evaluation (that simulated A/B testing conditions).

6.2.1 Performance comparison

It is very interesting to see in Table 1 that the simple Global method outperforms both SVD and MC baselines in terms of AUC. This can be explained by the facts that most users follow highly popular blogs such as institutions or celebrities [6] and that both SVD and MC suffer from data sparsity. Yet, it is important to note that Global is outperformed by both SVD and MC for precision and recall results. Table 1 also shows that our proposed method, BIMC, achieves the best performance in all three evaluation metrics out of all methods. Note that the two best performing methods, BIMC and IMC, both utilize side information of users and blogs. This implies that such information is crucial to improve the recommendation quality.

In contrast, CMC performs the worst in the top-k list as seen in Table 1 showing that there does not exist significant shared low-rank structure between the follower graph and user/item features. Moreover, textual features in Tumblr are extremely sparse and noisy, which makes the CMC formulation more problematic. This clearly demonstrates that BIMC and IMC incorporate user/item features more effectively in a robust manner. Nonetheless, CMC still achieves similar AUC results compared with IMC. Katz performs comparably to BIMC in terms of AUC, but not in the top-k list, which can also be explained by the fact that similarities in S_u and S_b are affected by noisy features. In sum, we can see that BIMC achieves superior performance than other methods by successfully incorporating rich user and blog features.

Table 3: AUC results for different groups of users of the proposed Boosted Inductive Matrix Completion method (BIMC) in comparison to several baselines.

Method	Low	Medium	High
Global	0.8639	0.8623	0.8403
SVD	0.8488	0.8782	0.8715
MC	0.8473	0.8767	0.8709
Katz	0.9199	0.9304	0.9194
CMC	0.8978	0.9063	0.9058
IMC	0.9040	0.9299	0.9162
BIMC	**0.9209**	**0.9316**	**0.9198**

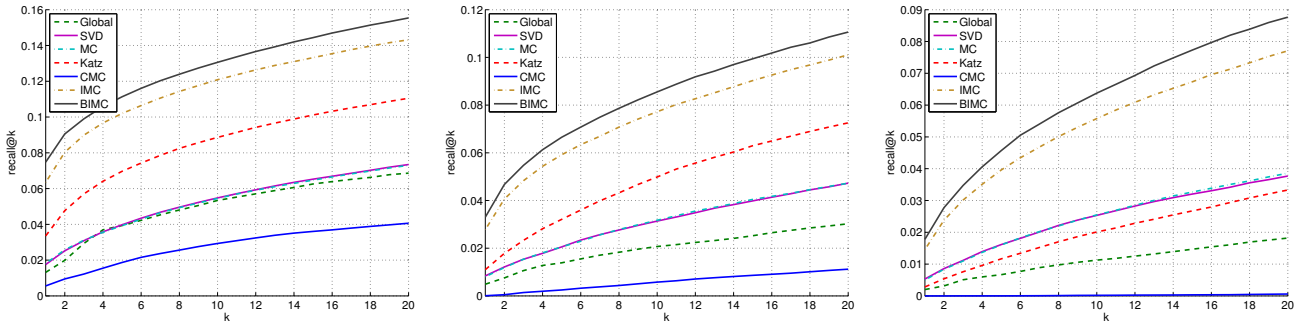
6.2.2 Temporal evaluation

Although cross-validation is a widely accepted evaluation methodology, it can produce biased results when temporal effects are not considered when splitting the data into training and testing sets. Thus, we evaluated all methods using another dataset divided into training and testing using a fixed date and time. Specifically, we use data from a 4-month period as training and the following 5th month data as testing. This evaluation is more similar than the offline-evaluation to A/B testing, which is broadly used in industry. Results for the temporal evaluation is given in Table 2, in which we can observe very similar results to the offline case in Table 1. This suggests that our proposed methods would also perform well in production settings.

6.2.3 Performance for users and items with different levels of sparsity

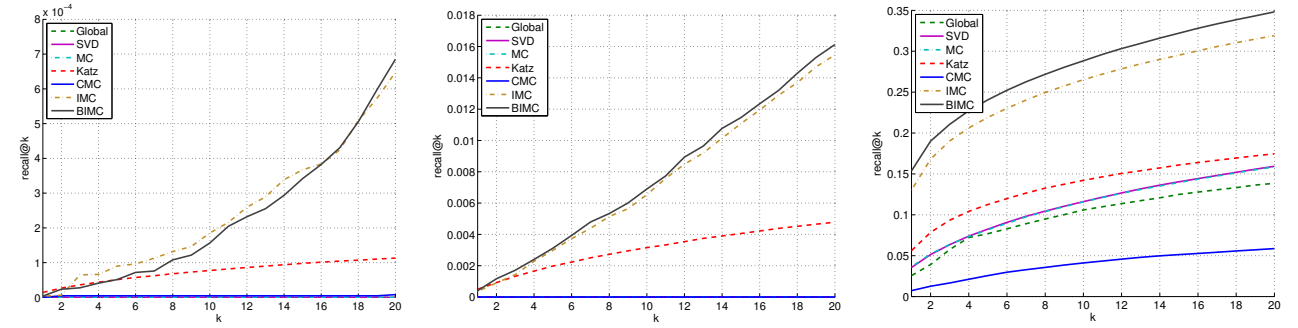
In order to better understand the effect of utilizing rich information about users and items in BIMC, we divide users as well as blogs into different categories based on the number of people they follow and the number of people who follow them. In other words, user and blog segmentation is done based on the number of nonzero elements in A, to examine how the methods perform under different levels of sparsity in the data. Specifically, users are partitioned into three groups based on the number of followees n_f: $n_f \leq 40$ (Low), $40 < n_f \leq 100$ (Medium) and $n_f > 100$ (High), where each group consists of about 89.36%, 7.81% and 2.83% of users. Similarly, we also partition the blog dimension (with the same thresholds), where each group consists of about 95.12%, 3.26% and 1.63% blogs respectively.

For each user category, we present Recall@k in Figure 5 for $k = 1, 2, \cdots, 20$. As shown in Figure 5, BIMC outperforms all other baselines for all user groups in terms of Recall@k. The second best method is IMC followed by Katz. This explicitly shows that utilizing both user and item features significantly helps in dealing with different sparsity conditions including cold-start. For the Low user group in Figure 5, it is interesting to see that SVD and MC suffers from severe sparsity and therefore perform comparably with the Global baseline. Note that the performance decreases for all methods as we move from Low to High user groups. This can be explained by the facts that users who already follow many popular blogs would need to be recommended more diverse blogs in the long-tail, which is generally a much harder task. Table 3 presents AUC results on all three user groups, where we can see that BIMC achieves the largest AUC values across all user categories. As discussed in Section 6.2.1, CMC is not able to make any good predictions in the top-



(a) Users with low activity (b) Users with medium activity (c) Users with high activity

Figure 5: Recall@k results for user groups with different activity levels (low/medium/high) the proposed Boosted Inductive Matrix Completion method (i.e., BIMC) in comparison to several baselines ($k = 1, 2, \cdots, 20$).



(a) Blogs with low popularity (b) Blogs with medium popularity (c) Blogs with high popularity

Figure 6: Recall@k results for blog groups with different popularity levels (low/medium/high) of the proposed Boosted Inductive Matrix Completion method (i.e., BIMC) in comparison to several baselines ($k = 1, 2, \cdots, 20$).

k list, but still achieves reasonable AUC levels as shown in Table 3. This set of results shows that BIMC successfully handles data sparsity by incorporating rich user and blog features, and significantly improves over other methods.

Next, we analyze the performance of all methods for blog groups with different popularity levels as shown in Figure 6, where we can observe similar trends as found in Figure 5 with different user groups. It can be seen in Figure 6 that BIMC and IMC outperforms all other baselines for all blog groups in terms of Recall@k, while all other baselines suffer from data sparsity, and cannot make any correct retrieval. For blogs with high popularity, their performances are much better. For this group, SVD and MC perform slightly better than the Global, but still much worse than BIMC and IMC. Another set of interesting results is the fact that IMC performs comparably with BIMC for items with low and medium popularity. This can be explained by the fact that the MC step in BIMC is suffering from data sparsity in both of these cases, and is not helping the BIMC as much as in the case of items with high popularity.

Finally, we analyze the performance of all methods for user groups and item groups jointly. Specifically, Figures 7 and 8 show the performances of all methods for all user and item groups jointly in terms of Precision@k and Recall@k respectively. It can be observed that both precision and recall increases significantly for users and items with high activity/popularity. For users with low and medium activity, all methods other than BIMC and IMC severely suffer from data sparsity. For users with high activity, BIMC outperforms IMC, both of which significantly outperform other baselines. Another interesting result is the fact that IMC outperforms or performs comparably with BIMC for user and items with low activity/popularity, showing that the the MC step in BIMC suffers from data sparsity and cannot effectively help IMC, in which case all prediction from BIMC depends on IMC step alone. Overall, this set of experiments clearly demonstrate the power of BIMC over IMC, as well as utilizing rich set of user and item features in BIMC and IMC over other baselines.

7. CONCLUSIONS

Recommending blogs to follow is one of the core tasks for online microblogging sites such as Tumblr for improving user engagement as well as advertising revenue. In this paper, we propose a novel boosted inductive matrix completion (BIMC) model for the task that combines the power of an inductive matrix completion model together with a standard matrix completion model. The proposed BIMC model focuses on the residual matrix that is calculated from the approximation matrix of a standard matrix completion (MC) model, and learns an inductive matrix completion model (IMC) to effectively utilize the rich side information of users and blogs to learn the missing links in the follower graph where a standard MC fails to learn. We utilize state-of-the art deep learning methods such as word2vec and convolutional neural networks to extract a comprehensive set of features. An extensive set of experiments conducted on large-scale real-world data from Tumblr demonstrate the effectiveness of the proposed BIMC over MC and IMC methods as well as several other baselines.

8. REFERENCES

[1] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *SIGKDD*, pages 19–28, 2009.

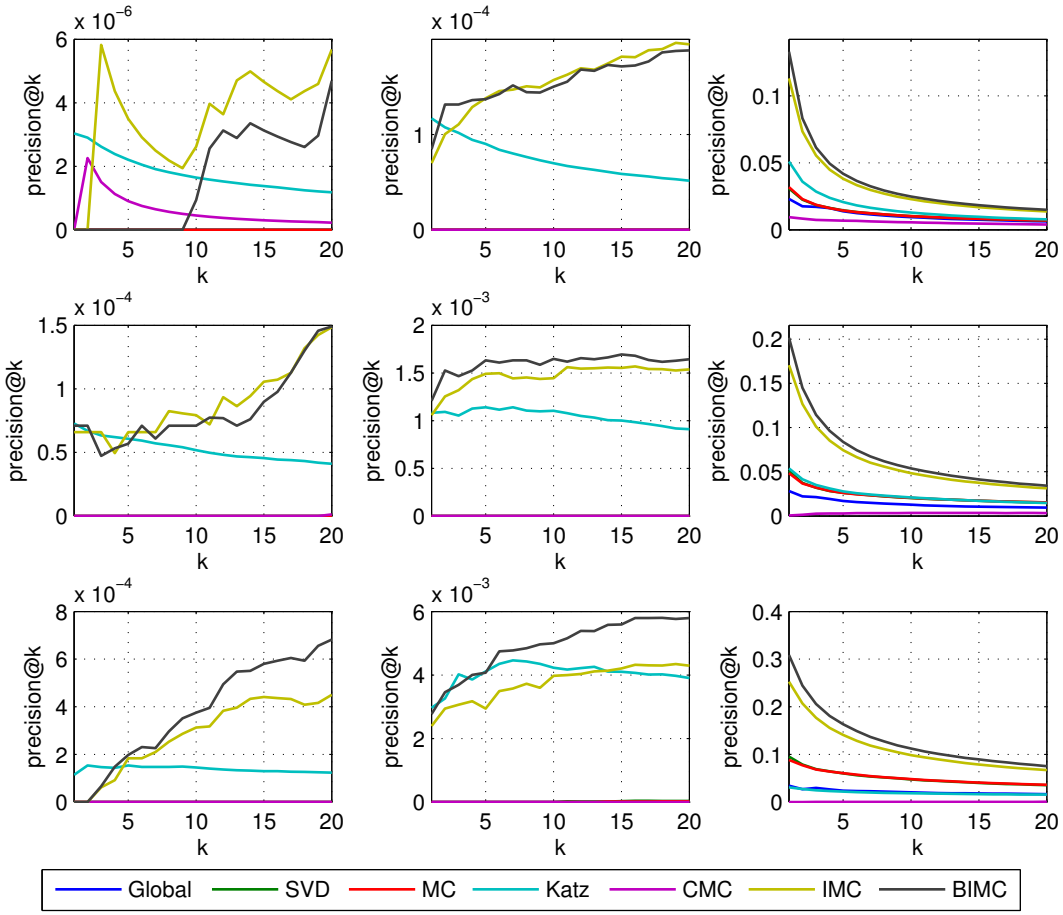


Figure 7: Precision at top-k results for different user and blog groups. Both users and blogs are divided into three groups (n_d: the number of links): (1) $n_d \leq 40$, (2) $40 < n_d \leq 100$, and (3) $n_d > 100$. From left to right are blog groups 1, 2 and 3, and from top to bottom are user groups 1, 2 and 3. BIMC outperforms other methods in most user-blog group combinations.

[2] M. G. Armentano, D. Godoy, and A. A. Amandi. Followee recommendation based on text analysis of micro-blogging activity. *Information Systems*, 38(8):1116–1127, 2013.

[3] R. M. Bell and Y. Koren. Lessons from the Netflix Prize Challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79, 2007.

[4] G. Bouchard, D. Yin, and S. Guo. Convex collective matrix factorization. In *AISTATS*, pages 144–152, 2013.

[5] E. Candès and B. Recht. Exact matrix completion via convex optimization. *CACM*, 55(6):111–119, 2012.

[6] Y. Chang, L. Tang, Y. Inagaki, and Y. Liu. What is Tumblr: A statistical overview and comparison. *SIGKDD Explor. Newsl.*, 16(1):21–29, 2014.

[7] Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward. Coherent matrix completion. In *ICML*, pages 674–682, 2014.

[8] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-N recommendation tasks. In *RecSys*, pages 39–46, 2010.

[9] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[10] O. U. Florez and L. Nachman. Deep learning of semantic word representations to implement a content-based recommender for the RecSys Challenge'14. In *ESWC*, 2014.

[11] S. Gunasekar, M. Yamada, D. Yin, and Y. Chang. Consistent collective matrix completion under joint low rank structure. In *AISTATS*, pages 306–314, 2015.

[12] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh. WTF: The Who to Follow service at Twitter. In *WWW*, pages 505–514, 2013.

[13] J. Hannon, M. Bennett, and B. Smyth. Recommending Twitter users to follow using content and collaborative filtering approaches. In *RecSys*, pages 199–206, 2010.

[14] L. Hong, A. S. Doumith, and B. D. Davison. Co-factorization machines: modeling user interests and predicting individual decisions in Twitter. In *WSDM*, pages 557–566, 2013.

[15] Y. Hsiang-Fu, P. Jain, P. Kar, and I. S. Dhillon. Large-scale multi-label learning with missing labels. In *ICML*, pages 593–601, 2014.

[16] P. Jain and I. S. Dhillon. Provable inductive matrix completion. *CoRR*, abs/1306.0626, 2013.

[17] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *STOC*, pages 665–674, 2013.

[18] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142, 2010.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[20] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *JMLR*, 11:2057–2078, 2010.

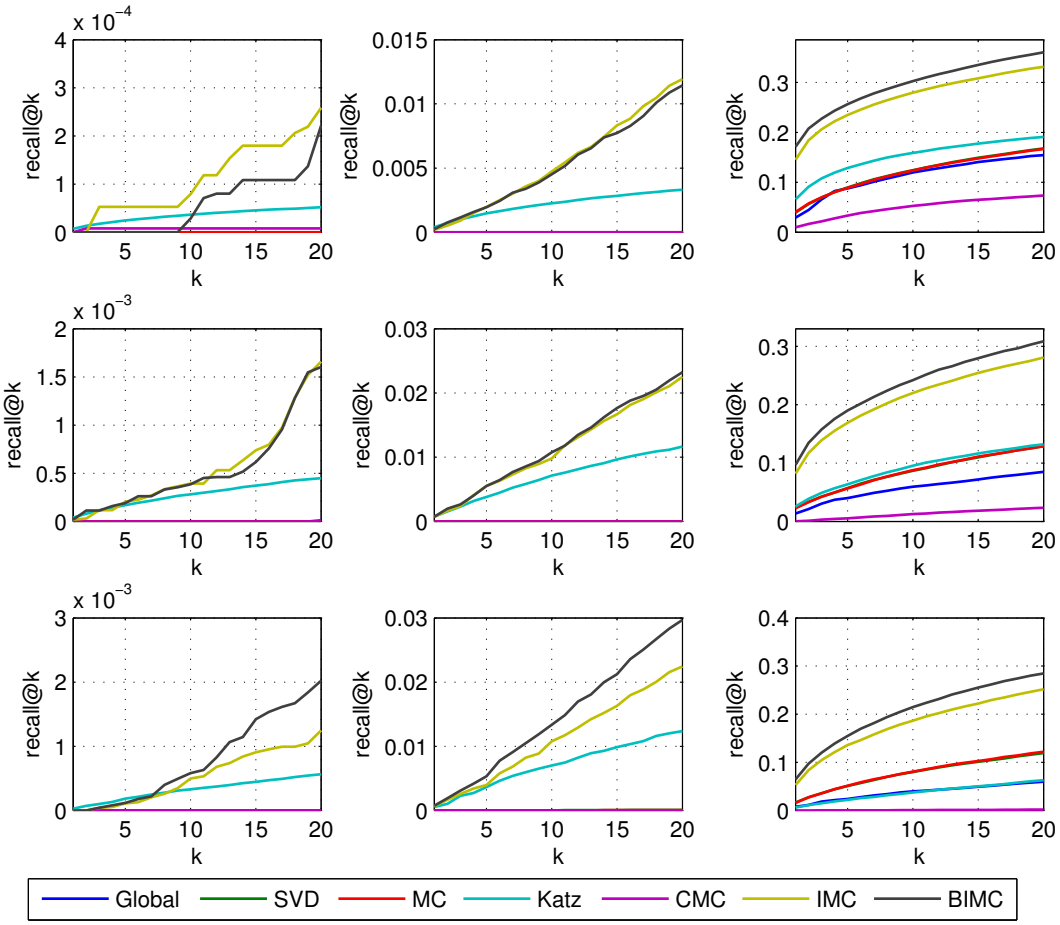


Figure 8: Recall at top-k results for different user and blog groups. Both users and blogs are divided into three groups (n_d: the number of links): (1) $n_d \leq 40$, (2) $40 < n_d \leq 100$, and (3) $n_d > 100$. From left to right are blog groups 1, 2 and 3, and from top to bottom are user groups 1, 2 and 3. BIMC outperforms other methods in most user-blog group combinations.

[21] Y. Kim and K. Shim. Twitobi: A recommendation system for Twitter using probabilistic modeling. In *ICDM*, pages 340–349, 2011.

[22] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR*, pages 195–202, 2009.

[23] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[24] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[25] A. Levi, O. Mokryn, C. Diot, and N. Taft. Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *RecSys*, pages 115–122, 2012.

[26] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, pages 287–296, 2011.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[28] Y. Moshfeghi, B. Piwowarski, and J. M. Jose. Handling data sparsity in collaborative filtering using emotion and semantic based features. In *SIGIR*, pages 625–634, 2011.

[29] N. Natarajan, D. Shin, and I. S. Dhillon. Which app will you use next? collaborative filtering with interactional context. In *RecSys*, pages 201–208, 2013.

[30] A. Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *NIPS*, pages 2643–2651, 2013.

[31] Y. Shi, M. Larson, and A. Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45, 2014.

[32] D. Shin, S. Si, and I. S. Dhillon. Multi-scale link prediction. In *CIKM*, pages 215–224, 2012.

[33] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *SIGKDD*, pages 650–658, 2008.

[34] V. Vasuki, N. Natarajan, Z. Lu, B. Savas, and I. S. Dhillon. Scalable affiliation recommendation using auxiliary networks. *ACM TIST*, 3(1):3:1–3:20, 2011.

[35] J. Weston, S. Chopra, and K. Adams. #TagSpace: Semantic embeddings from hashtags. In *EMNLP*, pages 1822–1827, 2014.

[36] D. Yin, L. Hong, and B. D. Davison. Structural link analysis and prediction in microblogs. In *CIKM*, pages 1163–1168, 2011.

[37] G. Zhao, M. L. Lee, W. Hsu, W. Chen, and H. Hu. Community-based user recommendation in uni-directional social networks. In *CIKM*, pages 189–198, 2013.

[38] Y. Zhen, W.-J. Li, and D.-Y. Yeung. Tagicofi: Tag informed collaborative filtering. In *RecSys*, pages 69–76, 2009.