Temporal Bipartite Projection and Link Prediction for Online Social Networks

Tsunghan Wu*, Sheau-Harn Yu*, Wanjiun Liao† and Cheng-Shang Chang‡
*Graduate Institute of Electrical Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.
Email: d99921033@ntu.edu.tw, r99921047@ntu.edu.tw
†Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.
Email: wjliao@ntu.edu.tw
‡Institute of Communications Engineering
National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C.
Email: cschang@ee.nthu.edu.tw

***Abstract*—In user-item networks, the link prediction problem has received considerable attentions and has many applications (e.g., recommender systems, ranking item popularity) in recent years. Many previous works commonly fail to utilize the dynamic nature of the networks. This paper focuses on dealing with the temporal information and proposes an algorithm to cope with the link prediction problem on bipartite networks. We describe a temporal bipartite projection method that yields a projected item graph, called the temporal projection graph (TPG). Based on the TPG, we propose a scoring function called STEP (Score for TEmporal Prediction) for each user-item pair. STEP leverages the historical behaviors of individual users and the social aggregated behaviors learned from the TPG for the link prediction problem. Furthermore, we use TPG and PageRank to rank the popularity of items. To validate our algorithms, we perform various experiments by using the DBLP author-conference dataset, the Flickr dataset and the Delicious dataset. We show that our results of the link prediction problem for new links are substantially better than other temporal link prediction algorithms. We also find the item rankings generated by our approach match very well with that existed in the real world.**

***Keywords*-bipartite network; bipartite network projection; link prediction; PageRank;**

I. Introduction

Online social networks are very popular now and they are very dynamic in nature. In such networks, there are many people interacting with each other, expressing their opinions, sharing files, viewing or purchasing things all the time from all over the world. In the literature, sequences of bipartite networks indexed by time are commonly used for modelling (or logging) the interactions in online social networks. In general, there are two kinds of nodes in such bipartite networks, users and items (such as products, files, and articles) and such bipartite networks are called the *user-item networks*. A (weighted) link between a user node and an item node at a certain time represents the specific relationship between that user and that item such as buying, sharing, or discussing certain topics at that time. By studying the dynamics of such bipartite networks, one might be able to infer the popularity of an item and the preference of each individual to make personal recommendation or market survey. For this, we address two specific problems in this paper:

(i) Given a temporal sequence of user-item networks, how can one generate an item graph to characterize the relations among items?
(ii) Given a temporal sequence of user-item networks, how can one predict future links between users and items?

The second problem is commonly known as the *link prediction* problem in the literature and has received a lot of attention lately. There are roughly four classes of approaches for the link prediction problem: methods based on similarity of nodes [3, 4, 5], methods based on classification or regression [6, 7, 8], methods based on graph mining [9, 10], and methods based on tensor factorization and Holt-Winters forecasting [2, 11, 12, 13].

The first problem is called the *temporal bipartite projection* problem in this paper and it seems to be less studied in the literature. To deal with the huge amount of data in a temporal sequence of bipartite networks, one has to "condense" them into a simpler (and preferably meaningful) representation for future data processing. The simplest method is to collapse the sequence of bipartite networks (by a weighted sum) into a single weighted bipartite network and then perform bipartite projection (and link prediction) based on the weighted bipartite network (see e.g., [3, 4, 5, 6, 7, 8]). The drawback of such an approach is losing the valuable temporal information (e.g., the order of events) and individual user history. In particular, it would be very difficult to find out whether an item is the substitute of another item without the temporal information.

There are few previous related works that have taken the temporal information into account [2, 9, 10, 14, 15]. Eubak et al. [14] proposed using dynamic bipartite graphs to model the physical contact patterns for disease outbreaks.

Koren [15] considered the problem of tracking the temporal dynamics of customer preferences to products. By studying time evolution graphs, Berlingerio et al. [9, 10] developed GERM for mining graph evolution rules. Dunlavy et al. [2] used the matrix and tensor factorization for the link prediction problem. One drawback of tensor factorization is that one loses the physical insights of the temporal information during the factorization process. Besides, their experimental results are not substantially better than those from the well-known (truncated) Katz method when using the DBLP dataset for comparison.

For the first problem, we propose in this paper a temporal bipartite projection (TBP) method to generate an item graph, called the temporal projection graph (TPG), that characterizes the transition tendencies among *similar* items. The key idea is to identify the *transitions* made by each user from one item to another similar item over time. By assigning each transition a weight, the sum of the weights of all the transitions from one item to another item (over all users) is then the transition tendency between these two items. Intuitively, transition tendency obtained this way can be viewed as a social aggregated change of the preference among all items. We then construct the TPG as a directed weighted item graph by assigning the transition tendencies as its edge weights. There are some factors that one might need to consider for the weight assignment of a transition, including computational complexity, occurrence of the transition, the duration of the transition, weight between a user and an item in the transition, and normalization among users.

There are several things that can be inferred from the TPG. In particular, one can use PageRank [1] to rank the popularity of items. Also, one can use transition tendencies to identify substitute relationship between two items. If the transition tendency from one item to another item is high and the transition tendency of the opposite direction is close to 0, then we may infer that the latter is a substitute item of the former as the transitions are basically made one way. On the other hand, if the transition tendencies in both directions are high, then we may infer that these two items could be substitutes of each other.

For the link prediction problem, we propose the STEP (Score for TEmporal Prediction) method that uses the social aggregated behavior from the TPG and each individual history. For each user-item pair, STEP computes a score that is used for ranking the likelihood of the occurrence of a future link between a user and an item. The idea of STEP, quite similar to the construction of the TPG, is to identify all the *potential transitions* by looking at the history of that user. By assigning a weight to each potential transition, the score for each user-item pair is then the sum of the weights of all the potential transitions. As the weight assignment for a transition, the weight assignment for a potential transition also takes the temporal information into account. Moreover, since a potential transition is an event that has not happened yet, we also use the social aggregated behavior from the TPG for predicting the occurrence of a potential transition.

To test our methods, we use the DBLP dataset [16] that contains publications in computer science journals and proceedings from year 1936 to 2011. We follow the same setting as in [2] to parse the DBLP dataset into a sequence of author-conference bipartite networks. Thus, the users and the items in our setting are simply the authors and the conferences in the DBLP dataset, respectively. Our numerical results reveal many interesting findings that match very well with what really happened in the real world. In particular, we find out that INTERSPEECH is a substitute conference for EUROSPEECH. Moreover, INFOCOM is one of the three conferences with very large *self* transition tendencies. This shows that the authors of INFOCOM are very loyal to INFOCOM. We also use PageRank to rank the popularity of the conferences in the DBLP dataset. In particular, INFOCOM is constantly ranked within the top 15 popular conferences in all our training periods. Also, for the link prediction problem, our result is substantially better than the method with the best performance in [2] for the DBLP dataset [16], the Flickr dataset [17] and the Delicious dataset [17].

The rest of the paper is organized as follows. In Section II, we propose a temporal bipartite projection method for the construction of the TPG. In section III, we propose STEP for link prediction. In Section IV, we report our experimental results by using the DBLP author-conference dataset, the Flickr dataset and the Delicious dataset. The paper is concluded in Section V, where we address possible extensions of our work.

II. Temporal Bipartite Projection

A. Temporal projection graph

The traditional bipartite projection method is to project a bipartite graph into a unipartite graph of one kind of nodes in the original bipartite graph. For a bipartite network with users and items as their two kinds of nodes, called the user-item network (graph) in this paper, the traditional method is to project such a graph into either a user graph (a graph with all its nodes being users) or an item graph (a graph with all its nodes being items). There are many methods for bipartite projection (see e.g., [4]). In particular, we show in Fig. 1 an example of bipartite projection that results in an unweighted graph, where a link is created in the projected graph *if the two nodes in the user-item graph have common neighbors*. Since the traditional bipartite projection method simply maps a user-item bipartite network to either a user network (graph) or an item network (graph), one has to collapse all the temporal bipartite networks into a single one in order to use such a method. This results in the loss of temporal information and that motivates us to define the temporal bipartite projection that takes the temporal issue into account.

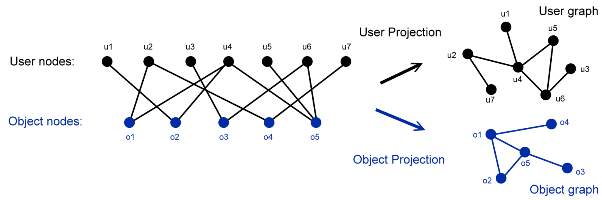


Figure 1. An example of the traditional bipartite projection method.

As in the traditional bipartite projection, one can project a sequence of weighted bipartite networks into either a *user* network or an *item* network. In [18], Goyal et al. proposed several methods to learn user influence probabilities by looking into a sequence of action logs (with each log indicating a certain user performing an action on a certain item at a certain time). In the discrete-time setting, a sequence of action logs can be equivalently represented by a sequence of bipartite networks. As such, the user influence graph learned by Goyal et al. [18] is in fact a projection of a sequence of bipartite networks into a *user* network.

In this paper, we focus on the other direction of the projection. Our temporal bipartite projection (TBP) method is a mapping from a sequence of weighted bipartite networks to a weighted directed *item* graph. Without loss of generality, we assume each bipartite network consists of two kinds of nodes: a set of N users $U = \{1, 2, \ldots, N\}$ and a set of M items $\mathcal{I} = \{1, 2, \ldots, M\}$. We consider a sequence of bipartite networks $\{G^t, t = 1, 2 \ldots, T\}$ with T being the length of the training period. Let $g^t_{n,m}$, $n = 1, 2, \ldots, N$, $m = 1, 2, \ldots, M$, $t = 1, 2, \ldots, T$, be the weight between user n and item m in the bipartite network G^t. In this paper, we assume that all the weights are nonnegative, i.e., $g^t_{n,m} \geq 0$. For simplicity, we call $g^t_{n,m}$ the weight between user n and item m at time t.

In addition to the sequence of bipartite networks, we assume that we have some preliminary knowledge regarding the similarity of items. For this, we assume there is an item similarity graph $G_{\mathcal{I}} = (\mathcal{I}, E_{\mathcal{I}})$ in which an edge $(m_1, m_2) \in E_{\mathcal{I}}$ indicates that the two items m_1 and m_2 are similar. With the graph $G_{\mathcal{I}}$, we can take the temporal issue into account by identifying transitions between two similar items. Specifically, A transition for a user n from item m_1 at time t_1 to another item m_2 at time t_2 ($t_2 > t_1$) if (i) these two items are similar, (ii) both the weight between user n and item m_1 at time t_1 and the weight between user n and item m_2 at time t_2 are positive, i.e., $g^{t_1}_{n,m_1} > 0$ and $g^{t_2}_{n,m_2} > 0$. In practice, the physical meaning of a transition might reflect the change of the behavior (preference) of a user, e.g., a user buys one item earlier and then buys another similar item some time later. If these two items are the same, a self transition also reflects the loyalty of that user to that item. The concept of a transition is formalized as follows:

Definition 1: (**Transition for a user between two items**) If (i) $(m_1, m_2) \in E_{\mathcal{I}}$ and (ii) $g^{t_1}_{n,m_1} > 0$ and $g^{t_2}_{n,m_2} > 0$ for some user n in U, some items m_1 and m_2 in $\mathcal{I}$, and $1 \leq t_1 < t_2 \leq T$, then we say there is a transition for user n from item m_1 at time t_1 to another item m_2 at time t_2. Such a transition is denoted by the five tuple (n, m_1, t_1, m_2, t_2).

Intuitively, each transition records the change of the preference of a user over time. If we assign each transition a weight and then sum up the weights of all the transitions, the result could be viewed as a social aggregated change of the preference among all items. This leads us to define the *transition tendency* between two items.

Definition 2: (**Transition tendency between two items**) Suppose the weight for a transition (n, i, t_1, j, t_2) is $w(n, i, t_1, j, t_2)$. Then the *transition tendency* from item i to item j, denoted by $TranTend(i, j)$, is

$$\sum_{n=1}^{N} \sum_{t_1=1}^{T-1} \sum_{t_2=t_1+1}^{T} w(n, i, t_1, j, t_2).$$

Clearly, the matrix $\hat{G} = (\hat{g}_{i,j})$ induces a directed weighted item graph. The graph is called the temporal projection graph (TPG) of the sequence of the bipartite networks $\{G^t, t = 1, 2 \ldots, T\}$.

Definition 3: (**Temporal projection graph (TPG)**) The temporal projection graph (TPG) of the sequence of the bipartite networks $\{G^t, t = 1, 2 \ldots, T\}$ is the directed weighted item graph with the adjacency matrix $\hat{G} = (\hat{g}_{i,j})$, where

$$\begin{aligned} \hat{g}_{i,j} &= TranTend(i, j) \\ &= \sum_{n=1}^{N} \sum_{t_1=1}^{T-1} \sum_{t_2=t_1+1}^{T} w(n, i, t_1, j, t_2). \end{aligned} \quad (1)$$

One problem of the TPG is that the range of its edge weights, i.e., transition tendencies, might be very large. For this, we propose normalizing the edge weights to obtain item transition probabilities.

Definition 4: (**Item transition graph (ITG)**) The item transition graph (ITG) of the sequence of the bipartite networks $\{G^t, t = 1, 2 \ldots, T\}$ is the directed weighted item graph with the adjacency matrix $P = (p_{i,j})$, where

$$p_{i,j} = \frac{\hat{g}_{i,j}}{\sum_{m=1}^{M} \hat{g}_{i,m}}. \quad (2)$$

The probability $p_{i,j}$ is the item transition probability that indicates the probability of a user to switch from item i to item j.

One possible application of the ITG is for ranking the popularity of items. Analogous to PageRank [1], we can model the behavior of a user by a random walk with random jumps and then use that to compute the steady state probability for a user to visit a specific item in the ITG. The steady state probabilities of these items, also called PageRank here, are then used for ranking these items. Specifically, suppose that there are M items and a user uniformly selects an item with probability $1/M$. Once he/she is on an item, he/she continues the random walk with probability β. This is done by selecting one of the neighboring items according to the item transition probabilities. On the other hand, with probability $1-\beta$ he/she performs a random jump and starts a new item *uniformly* among all the M items. Clearly, such a random walk induces a Markov chain and the steady probability of item i, denoted by $PR(i)$, satisfies

$$PR(i) = (1-\beta)\frac{1}{n} + \beta \sum_{m=1}^{M} PR(m) \cdot p_{mi}. \tag{3}$$

In all our experiments, the parameter β is set to be 0.85 as in [1].

B. Weight assignment for a transition

Now the problem remains to find a method to assign the weight for a transition (n, i, t_1, j, t_2). There are several factors that one might need to consider for this.

- (i) *Computational complexity*: The effort for computing the weight should be minimum for a large dataset.
- (ii) *Occurrence of the transition*: A recent transition should carry a larger weight than an obsolete transition.
- (iii) *Duration of the transition*: A transition with a shorter time difference should carry a larger weight than that with a larger time difference.
- (iv) *Weight between a user and an item in the transition*: A transition with a larger weight between a user and an item at time t_1 should be assigned with a larger weight than another transition with a smaller weight between the same user and another item at the same time.
- (v) *Normalization among users*: As transition tendency can be viewed as a social aggregated preference of items, the weight assignment should avoid one user dominating the social aggregated preference.

There are many approaches for weight assignment in learning influence probabilities in [18]. The simplest one is to use the discrete-time Bernoulli model in [18], i.e.,

$$w(n, i, t_1, j, t_2) = 1, \tag{4}$$

when $t_2 - t_1$ is not greater than a specific time window. Such a model, though having the lowest computational complexity, was shown to achieve comparable performance to the other models in [18] that require much more intensive computational efforts. Unfortunately, such a simple model does not perform well in our setting for link prediction (from our experiments). Instead, we adopt a more complicated weight assignment approach by taking the discounting factor and the other factors into account. Specifically, for all our experiments, we use

$$\begin{aligned} w(n, i, t_1, j, t_2) &= \alpha(t_1) \times \frac{g_{n,i}^{t_1}}{\sum_{\ell=1}^{M} g_{n,\ell}^{t_1}} \\ &\times \frac{\alpha(T-(t_2-t_1))}{\sum_{s=t_1+1}^{T} \mathbf{I}_n^s \times \alpha(T-(s-t_1))}, \end{aligned} \tag{5}$$

where $\alpha(t) = (0.8)^{T-t}$ is the discounting factor and

$$\mathbf{I}_n^t = \begin{cases} 1 & \text{if } \sum_{j=1}^{M} g_{n,j}^t > 0, \\ 0 & \text{otherwise.} \end{cases}$$

We note that the second term in the product is to consider the factor for the weights between a user and an item in (iv), while the third term in the product is to avoid one user dominating the social aggregated preference in (v).

Regarding the computation complexity of the TPG from the weight assignment in (5), we note that for each user $n = 1, 2, \ldots, N$, we can precompute $\sum_{\ell=1}^{M} g_{n,\ell}^t$ for all $t = 1, 2, \ldots, T$, with $O(NMT)$ complexity and store them in $O(NT)$ memory. Also, we can precompute $\sum_{s=t+1}^{T} \mathbf{I}_n^s \times \alpha(T-(s-t))$ for all n and t with $O(NT^2)$ computational complexity and store them in $O(NT)$ memory. Note that the computational complexity of $\mathbf{I}_n^s$ is $O(1)$ by using $\sum_{\ell=1}^{M} g_{n,\ell}^s$ in memory. With these in memory, the computation complexity of the weight in (5) is $O(1)$. Thus, the computation complexity of $\hat{g}_{i,j}$ in (1) is $O(NT^2)$. For all the M^2 edge weights in the TPG, the computation complexity is then $O(NT^2M^2)$ as the computational complexity for the precomputation is only $O(N \max(MT, T^2))$ (with additional $O(NT)$ memory).

C. An illustrating example of the TBP method and the TPG

We give an example of how the TBP method works and how to calculate $\hat{g}_{i,j}$ and $PR(i)$ here.

Let $U = \{1, 2\}$, $\mathcal{I} = \{1, 2, 3\}$, and $T = 3$, i.e., there are two users, three items, and three user-item networks. The sequence of the user-item bipartite networks $\{G^t, t = 1, 2, 3\}$ are characterized by the following three biadjacency matrices:

$$G^1 = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix}, G^2 = \begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, G^3 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

We assume that these three items are similar to each other, i.e., the item similarity graph $G_{\mathcal{I}}$ is a clique of three nodes. There are eight transitions for user 1:
$(1, 1, 1, 2, 2)$, $(1, 1, 1, 2, 3)$, $(1, 1, 1, 3, 3)$, $(1, 2, 1, 2, 2)$, $(1, 2, 1, 2, 3)$, $(1, 2, 1, 3, 3)$, $(1, 2, 2, 2, 3)$ and $(1, 2, 2, 3, 3)$.
Also, there are two transitions for user 2:
$(2, 1, 1, 1, 3)$ and $(2, 1, 1, 3, 3)$.

Then we can start to calculate the weight of each transition and then sum them up to obtain the weights in the TPG.

In Fig. 2, we show how one can obtain the TPG by adding the weight for each of the ten transitions in the directed weighted item graph. To ease our presentation in Fig. 2, we denote $\Delta g_{i,j}$ the weight for such a transition $(1, i, t_1, j, t_2)$ for user 1 or $(2, i, t_1, j, t_2)$ for user 2 calculated from (5). Each sub-figure illustrates the weights in the directed item graph after adding $\Delta g_{i,j}$ in each transition. After adding all the weights of the ten transitions, we obtain the TPG for this example in Fig. 2(k).

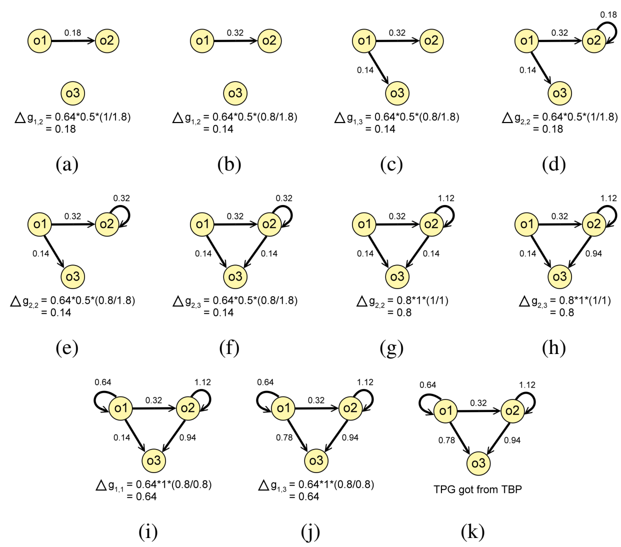


Figure 2. An illustrating example of the TBP method and the TPG. (a) $(1, 1, 1, 2, 2)$. (b) $(1, 1, 1, 2, 3)$. (c) $(1, 1, 1, 3, 3)$. (d) $(1, 2, 1, 2, 2)$. (e) $(1, 2, 1, 2, 3)$. (f) $(1, 2, 1, 3, 3)$. (g) $(1, 2, 2, 2, 3)$. (h) $(1, 2, 2, 3, 3)$. (i) $(2, 1, 1, 1, 3)$. (j) $(2, 1, 1, 3, 3)$. (k) The TPG from the TBP method.

From Fig. 2(k), we can obtain all transition tendencies as follows:
$\hat{g}_{2,2} = 1.12$, $\hat{g}_{2,3} = 0.94222$, $\hat{g}_{1,3} = 0.78222$, $\hat{g}_{1,1} = 0.64$, $\hat{g}_{1,2} = 0.32$, and the others are zero.

We can use (2) and the TPG (see Fig. 2(k)) to calculate the transition probabilities of all items pairs as follows:

$$[p_{i,j}] \simeq \begin{bmatrix} 0.36735 & 0.18368 & 0.44898 \\ 0 & 0.5431 & 0.4569 \\ 0 & 0 & 0 \end{bmatrix}.$$

The PageRank of these three items are as follows:
$PR(1) \simeq 0.0737$, $PR(2) \simeq 0.114$, and $PR(3) \simeq 0.122$.

III. Link Prediction

In this section, we address the link prediction problem. The link prediction problem is usually attacked by computing a score for each user-item pair based on the sequence of bipartite networks in the training period. The score is then used for ranking the likelihood of the occurrence of a future link between a user and an item. To compute the score for each user-item pair, we propose the STEP (Score for TEmporal Prediction) method that uses the social aggregated behavior from the TPG and each individual history.

A. The STEP method

In this section, we propose the STEP method for link prediction. Our basic assumption for link prediction is that the future behavior of a user depends on *his/her personal history* and *the social preference*. As discussed before, ITG can be viewed as the aggregated social behavior and the item transition probability from item i to item j represents the percentage of users who performed an action on item i and then switched to item j. As such, if a user, say user n, performed an action on item i, then the item transition probability p_{ij} could be used for predicting the event that he/she will switch to item j. Of course, such a probability should be discounted and it depends on when the action on item i is performed. Another factor is the weight between user n and item i at time t, i.e., $g^t_{n,i}$. If such a weight could be interpreted as the number of repetitive actions, e.g., the number of papers in the same conference or the number of posts with the same tag, then it is also reasonable to assume that the probability for user n to perform an action on item j at time $T+1$ is proportional to the weight $g^t_{n,i}$. Specifically, we make the following assumption:

(A1) If user n performed an action on item i at time t with weight $g^t_{n,i}$, then the probability that this user will perform an action on item j at time $T + 1$ is

$$\max[\alpha(t) \cdot g^t_{n,i} \cdot p_{ij}, 1]. \quad (6)$$

where $0 < \alpha(t) \leq 1$ is the discounting factor for the time that the action on item i was performed.

In view of (A1), if we would like to predict whether user n will perform an action on item j at time $T+1$, we need to find all the actions that user n performed on the items that are similar to j before. This leads to the following definition of *potential transitions*.

Definition 5: (**Potential transition for a target user from one item to a target item**) For a target user n in U and a target item j in $\mathcal{I}$, if (i) $(i, j) \in E_{\mathcal{I}}$ and (ii) $g^t_{n,i} > 0$ for some $1 \leq t \leq T$, then we say there is a potential transition for the target user n from item i at time t to the target item j at time $T + 1$. Such a potential transition is denoted by the five tuple $(n, i, t, j, T + 1)$.

Let $P(n, i, t, j, T + 1)$ be the probability of the potential transition $(n, i, t, j, T + 1)$. Then it follows from (6) that

$$P(n, i, t, j, T + 1) = \max[\alpha(t) \cdot g^t_{n,i} \cdot p_{ij}, 1]. \quad (7)$$

If we view each potential transition as a future event, then the event that user n will perform an action on item j at time $T + 1$ is the union of all the potential transitions for user n on item j. To compute the probability for a union of events is in general very difficult (unless these events are independent). However, if the probability of each event is rather *small*, then the probability for a union of events can be well approximated by the sum of the probability of each event. This leads us to the following way to compute the score for the event that user n will perform an action on item j at time $T+1$.

Definition 6: (**Score between a target user and a target item**) Assign a potential transition $(n, i, t, j, T+1)$ with the weight $w(n, i, t, j, T+1) = \alpha(t) \cdot g^t_{n,i} \cdot p_{ij}$. The *score* between the target user n and the target item j, denoted by $Score(n, j)$, is

$$\sum_{t=1}^{T} \sum_{i=1}^{M} w(n, i, t, j, T+1). \quad (8)$$

It is of interest to compare STEP with item-based collaborative filtering [19]. For a classical item-based collaborative filtering approach, one computes a "similarity" score between two items for a user-item network and then use the similarity score to compute a weighted average score between a user and an item. For STEP, the transition probability p_{ij} can be viewed as the "similarity" score between items i and j. The difference here is that such a similarity score is not symmetric and is discounted by a timing factor. As such, STEP can be viewed as an item-based collaborative filtering approach with the enhancement of adding the *timing information* into the discounting factor of an *asymmetric* similarity score.

B. An illustrating example of the STEP method

We consider the same example as in Section II-C and choose $\alpha(t) = 0.8^{T-t} = 0.8^{3-t}$. Note that the transition probability matrix computed there is an upper triangular matrix. In view of (8), there is no need to compute the weight for the potential transitions with a zero transition probability. The scores of all user-item pairs are as follows:

$$\begin{aligned}
Score(1,1) &= \alpha(1) \times g^1_{1,1} \times p_{1,1} \simeq 0.235, \\
Score(1,2) &= \alpha(1) \times g^1_{1,1} \times p_{1,2} + \alpha(1) \times g^1_{1,2} \times p_{2,2} \\
&\quad + \alpha(2) \times g^2_{1,2} \times p_{2,2} + \alpha(3) \times g^3_{1,2} \times p_{2,2} \\
&\simeq 1.877, \\
Score(1,3) &= \alpha(1) \times g^1_{1,1} \times p_{1,3} + \alpha(1) \times g^1_{1,2} \times p_{2,3} \\
&\quad + \alpha(2) \times g^2_{1,2} \times p_{2,3} + \alpha(3) \times g^3_{1,2} \times p_{2,3} \\
&\simeq 1.768, \\
Score(2,1) &= \alpha(1) \times g^1_{2,1} \times p_{1,1} + \alpha(3) \times g^3_{2,1} \times p_{1,1} \\
&\simeq 0.838, \\
Score(2,2) &= \alpha(1) \times g^1_{2,1} \times p_{1,2} + \alpha(3) \times g^3_{2,1} \times p_{1,2} \\
&\simeq 0.419, \\
Score(2,3) &= \alpha(1) \times g^1_{2,1} \times p_{1,3} + \alpha(3) \times g^3_{2,1} \times p_{1,3} \\
&\simeq 1.024.
\end{aligned}$$

Then we can sort these numerical results to rank these user-item pairs for link prediction. In this example, the pair of user 1 and item 2 has the highest score.

Table I
THE STATISTICS OF AUTHOR-CONFERENCE NETWORKS PARSED FROM DBLP

Training Years	Test Year	Auths	Confs.	Training Links (%Dens)	Test Links (%Dens)
1991-2000	2001	8749	1599	130171 (0.09)	13004 (0.09)
1992-2001	2002	10287	1746	154926 (0.09)	16818 (0.09)
1993-2002	2003	12102	1909	186160 (0.08)	21464 (0.09)
1994-2003	2004	14343	2107	225072 (0.07)	28801 (0.10)
1995-2004	2005	17628	2383	281385 (0.07)	37389 (0.09)
1996-2005	2006	21573	2661	354135 (0.06)	43495 (0.08)
1997-2006	2007	26060	2952	439247 (0.06)	52678 (0.07)
1998-2007	2008	30758	3290	531209 (0.05)	60079 (0.06)
1999-2008	2009	35851	3550	631617 (0.05)	69420 (0.05)
2000-2009	2010	40929	3789	734305 (0.05)	71183 (0.05)

IV. EXPERIMENTAL RESULTS

A. The DBLP dataset

To test our method, we use the DBLP dataset [16] that contains publications in computer science journals and proceedings from year 1936 to 2011. We follow the same setting as in [2] to parse the DBLP dataset into a sequence of author-conference bipartite networks. Thus, the users and the items in our setting are simply the authors and the conferences in the DBLP dataset, respectively. The granularity of time is in year here. We only consider publications of type *inproceedings* in DBLP and obtain the results from year 1991 to 2010. The training period $T = 10$ (years). Also, we only keep those authors with more than 10 publications in the training period. The statistics of the data is listed in Table I (%Dens means the density of links in percentage, i.e., %Dens = $\frac{\#links}{\#authors \times \#conferences}$). We note that there are some minor differences between our parsing results and the results in [2]. It might be due to the updates of the DBLP dataset.

B. Transition tendency and PageRank

After parsing the DBLP dataset into a sequence of author-conference networks, we use the TBP method for the sequence of author-conference networks from year 1991 to 2000 to obtain the transition tendencies among all the 1599 conferences. In our experiments, we assume that the item similarity graph $G_{\mathcal{I}}$ is a clique and thus all the 1599 conferences are similar.

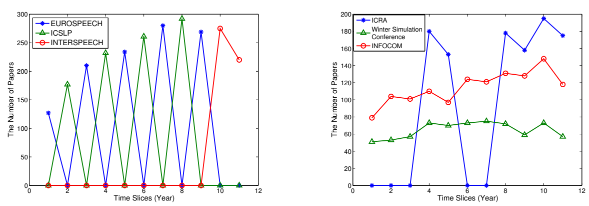
We list the three top transition tendencies (by using the weight assignment in (5)) in the TPG here:

1st: $TranTend$(EUROSPEECH, INTERSPEECH) $= 173.904$,

2nd: $TranTend$(ICRA, ICRA) $= 170.952$,

3rd: $TranTend$(Winter Simulation Conference, Winter Simulation Conference) $= 157.485$, The maximum transition tendency (i.e., maximum edge weight) in the TPG is 173.904, which is a significant conference transition because it is 607.4 times higher than mean transition tendency (which is only 0.2863). The maximum transition tendency is from conference EUROSPEECH to INTERSPEECH. INTERSPEECH is in fact an annual conference that merged the traditional two biennial conferences ICSLP (held in even years) and EUROSPEECH (held in odd years). As such, EUROSPEECH is replaced by INTERSPEECH (see Fig. 3(a) for the number of papers in EUROSPEECH, ICSLP, and INTERSPEECH during 1991 to 2001). This interesting truth is revealed from our transition tendency index as we have $TranTend$(EUROSPEECH, INTERSPEECH) $= 173.904$ and $TranTend$(INTERSPEECH, EUROSPEECH) $= 0$ in the TPG.

The second and third largest transition tendencies are $TranTend$(ICRA, ICRA) $= 170.952$, $TranTend$(Winter Simulation Conference, Winter Simulation Conference) $= 157.485$. These two transition tendencies are the transitions from one conference to itself (self transitions). As such, we might infer that ICRA and Winter Simulation Conference are *home* conferences for their areas and the authors in their areas have high loyalty to them (see Fig. 3(b) for the paper counts during 1991 to 2001 of these three conferences).



(a) (b)

Figure 3. The number of papers of some conferences from 1991 to 2001. (a) The number of papers in EUROSPEECH, ICSLP, and INTERSPEECH. (b) The number of papers of attractive conferences: ICRA, Winter Simulation Conference, and INFOCOM.

Once we obtain the TPG from the DBLP dataset, then we can use that to compute PageRank [1] for "ranking" the conferences in the dataset. In Table II, we show the ranking of the conferences from PageRank for each 10 year training period. As PageRank computes the steady probability for a random "walker" with random jumps to attend a conference, our ranking by PageRank merely reflects the "popularity" of the conferences in the future and it should not be considered as a mean for ranking the "prestige" of the conferences. In Table III, we list the number of authors and the number of papers in Globecom and INFOCOM for the last five 10 year training periods. From Table II and III, we can find an interesting phenomenon. In the 1996-2005 period, there were still more papers and authors in INFOCOM than in Globecom. But Globecom had higher PageRank index than INFOCOM at the same time. Not surprisingly, there were more papers and authors in Globecom than INFOCOM during the next period (1997-2006). In the following periods, Globecom always has higher PageRank index than INFOCOM and Globecom also has more papers and more authors than INFOCOM. This shows that the popularity index by using PageRank is closely related to the number of papers and the number of authors in a conference in the future. However, a conference with a larger number of authors may not have a higher PageRank index than another conference with a much smaller number of authors. Lower PageRank index represents the number of papers and authors of a conference will decrease in the future.

Table III
NUMBERS OF AUTHORS AND PAPERS IN GLOBECOM AND INFOCOM

	Globecom		INFOCOM	
	papers	authors	papers	authors
1996-2005	1537	4345	2029	5629
1997-2006	2545	7297	2199	6269
1998-2007	3554	10363	2352	6862
1999-2008	4642	13757	2491	7433
2000-2009	5696	17130	2686	8177

C. Link prediction results

We define *new links* and *old links* in the sequence of user-item bipartite networks as follows.

Definition 7: **(New link)** A link with a positive weight between a user n and an item j at time t is called a *new* link at time t if

$$\sum_{s=1}^{t-1} g^s_{n,j} = 0 \text{ and } g^t_{n,j} > 0.$$

In the setting of the author-conference networks, a new link corresponds to a paper that an author has not published in that conference before. We use the STEP method for the new link prediction problem on the DBLP author-conference dataset, and compare the prediction performance with the result in Dunlavy *et al.* [2]. We predict the new links by ranking $Score(n, j)$ in (8) for the DBLP author-conference networks with the training period from year 1991 to 2000. The prediction result is then compared with what really happened in 2001 for validation. The precision-recall curve is shown in Fig. 4, where *precision* is the ratio of the number of true new links to the number of top ranked author-conference pairs and *recall* is the ratio of the number of true new links among the top ranked list of author-conference pairs to the total number of new links in the testing year.

The number of true new links in 2001 among the top 1000 scores computed by STEP (see Fig. 4) is 116, and thus the precision for STEP is 11.6%. We also reproduce the result by the method TKatz-CWT in [2] that has the best performance among all the methods discussed in [2]. That method only achieves 82.56 true positive prediction (on average). Thus its precision is only 8.256%, which is lower than 11.6% by STEP.

Table II
PAGERANK FOR THE CONFERENCES IN THE DBLP DATASET

	1991-2000	1992-2001	1993-2002	1994-2003	1995-2004	1996-2005	1997-2006	1998-2007	1999-2008	2000-2009
1	HICSS	HICSS	HICSS	IPDPS	HICSS	HICSS	IPDPS	Globecom	Globecom	Globecom
2	AAAI/IAAI	ICRA	IPDPS	HICSS	IPDPS	IPDPS	ICRA	IPDPS	ICC	ICC
3	INFOCOM	INFOCOM	ICRA	ICRA	ICRA	ICRA	HICSS	ICRA	IPDPS	ICIP
4	Euro-Par	INTERSP	INFOCOM	INTERSP	SAC	SAC	Globecom	HICSS	SAC	ICASSP
5	ICDE	Euro-Par	INTERSP	ICME	ICME	INTERSP	SAC	ICME	ICME	ICRA
6	PDPTA	IPDPS	Euro-Par	Euro-Par	INTERSP	ICME	ICME	ISCAS	ISCAS	IROS
7	ICRA	ICSE	SAC	SAC	ICIP	Globecom	IROS	SAC	ICRA	SAC
8	ICPR	DEXA	PDPTA	DATE	DATE	DATE	INTERSP	IROS	INTERSP	IPDPS
9	DAC	VLDB	VLDB	INFOCOM	ICCS	INFOCOM	INFOCOM	ICC	ICIP	ISCAS
10	SIGMOD	DAC	ICDE	ICIP (3)	AAMAS	AAMAS	AAAI	INFOCOM	INFOCOM	ICME

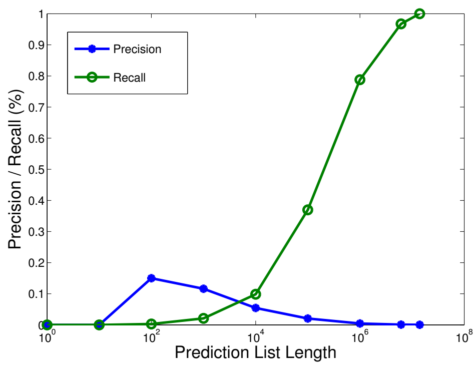


Figure 4. The precision-recall curve of using STEP for new link prediction.

Table IV
CORRECT PREDICTIONS OF NEW LINKS AMONG THE TOP 1000 SCORES OF STEP AND TKATZ-CWT.

Test Year	TKatz-CWT	STEP
	New Links	
2001	82.56	116
2002	90.11	123
2003	84.89	120
2004	106.22	128
2005	113.78	127
2006	86.44	104
2007	98.89	156
Mean	94.70	124.86

To compare STEP with the methods in [2], we list the new link prediction result by STEP and the method with the best performance (TKatz-CWT) in [2] in Table IV. As shown in Table IV, our STEP method outperforms the method with the best performance in [2].

D. Flickr and Delicious datasets

To further test the performance of our link prediction algorithm, we use the Flickr dataset for the year of 2004 and the Delicious dataset for the year of 2004 in [17]. In these two datasets, the user-item networks are user-tag networks. The weight between a user and a tag is the number of times that the user posts that tag in a month. In these two datasets, we use the first 10 months in 2004 as the training period, i.e., $T = 10$ (with the data in each month being a user-tag network), and the last two months, i.e., November and December, as the testing period. The statistics of these datasets are shown in Table V. Once again, we compare our STEP with TKatz-CWT in [2]. In Table VI, we show the number of correct predictions among the top 1000 scores for *all links* (both new links and old links) by STEP and TKatz-CWT. Our link prediction results are better than those by TKatz-CWT in [2] for these two datasets.

Table V
THE STATISTICS OF THE FLICKR DATASET AND THE DELICIOUS DATASET

	Training data			**Test data**		
	Users	Items	Links	Users	Items	Links
Flickr	9976	76088	315191	14090	93526	421181
Delicious	10682	79449	716061	14134	77680	552673

Table VI
CORRECT PREDICTIONS OF ALL LINKS AMONG THE TOP 1000 SCORES OF STEP AND TKATZ-CWT.

Dataset	TKatz-CWT	STEP
	All Links	
Flickr	186	336
Delicious	173	664

V. CONCLUSION

In this paper, we addressed two specific problems: (i) the temporal bipartite projection problem for generating an item graph to characterize the relations among items, and (ii) the link prediction problem for predicting links between users and items. For the first problem, we proposed a temporal bipartite projection method in Section II to generate the temporal projection graph (TPG) that characterizes the transition tendencies among items. The transition tendency from one item to another item is computed from a weighted sum of all the transitions and thus can be viewed as a social aggregated behavior. There are several things that can be inferred from the TPG. In particular, one can use PageRank [1] to rank the popularity of items. For the link prediction problem, we proposed the STEP method that computes a score for ranking the likelihood of the occurrence of a future link between a user and an item. As for the computation of transition tendencies, the score for a user-item pair computed by STEP is also a weighted sum of potential transitions between that user and that item. Since the weight of a potential transition depends on the transition probability from one item to another, STEP in fact uses individual user history and the social aggregated behavior from the TPG. To validate our approach, we performed various experiments by using the DBLP author-conference dataset. Our numerical results revealed many interesting findings that match very well with that really existed in the real world. The ranking results generated by ITG and PageRank can tell us the popularity of items in the future. Meanwhile, for the new link prediction problem, our result is substantially better than the method with the best performance in [2]. We also tested our results on the Flickr and Delicious datasets. For the link prediction problem, our result is also better than the method with the best performance in [2].

There are several issues that require further study:

(i) Weight assignment: Here we discussed several factors that one might consider for the weight assignment of a transition and a potential transition. However, as discussed in [18] for learning influence probabilities, there are still many choices for computing the weights. It is not clear which choice would be better.

(ii) Fusion of predictors: a future action of a user could be influenced by several factors. Here we only consider the global influence by constructing the item transition probabilities and then use those for an item-based link prediction. On the other hand, there is local influence (by friends) and the influence probabilities in [18] can also be used for user-based link prediction. How to fuse these predictors to improve the accuracy of link prediction could be an interesting research topic.

REFERENCES

[1] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

[2] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data*, 5(2), 2011.

[3] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[4] Tao Zhou, Jie Ren, Matus Medo, and Yi-Cheng Zhang. Bipartite network projection and personal recommendation. *Physical Review E*, 76, 046115, 2007.

[5] Oussama Allali, Clemence Magnien, and Matthieu Latapy. Link prediction in bipartite graphs using internal links and weighted projection. In *IEEE INFOCOM WKSHPS*, April 2011.

[6] Jerome Kunegis and Andreas Lommatzsch. Learning spectral graph transformations for link prediction. In *ACM ICML '09*, pages 561–568, June 2009.

[7] Jerome Kunegis, Ernesto W. De Luca, and Sahin Albayrak. The link prediction problem in bipartite networks. In *IPMU '10*, pages 380–389, June 2010.

[8] Nesserine Benchettara, Rushed Kanawati, and Celine Rouveirol. Supervised machine learning applied to link prediction in bipartite social networks. In *IEEE/ACM ASONAM '10*, pages 326–330, August 2010.

[9] Michele Berlingerio, Francesco Bonchi, Bjorn Bringmann, and Aristides Gionis. Mining graph evolution rules. In *ECML PKDD '09*, September 2009.

[10] Bjorn Bringmann, Michele Berlingerio, Francesco Bonchi, and Aristides Gionis. Learning and predicting the evolution of social networks. *IEEE Intelligent Systems*, 25(4):26–35, 2010.

[11] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[12] C. Chatfield. The holt-winters forecasting procedure. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 27(3):264–279, 1978.

[13] Chris Chatfield and Mohammad Yar. Holt-winters forecasting: Some practical issues. *Journal of the Royal Statistical Society, Series D (The Statistician)*, 37(2):129–140, 1988.

[14] Stephen Eubank, Hasan Guclu, V. S. Anil Kumar, Madhav V. Marathe, Aravind Srinivasan, Zoltan Toroczkal, and Nan Wang. Modeling disease outbreaks in realistic urban social networks. *Letters to Nature*, 429:180–184, 2004.

[15] Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[16] Dblp dataset. http://www.informatik.uni-trier.de/~ley/db/.

[17] Olaf Görlitz, Sergej Sizov, and Steffen Staab. Pints: peer-to-peer infrastructure for tagging systems. In *IPTPS*, page 19, 2008.

[18] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM, 2010.

[19] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. Springer, 2011.