Relationship Prediction in Dynamic Heterogeneous Information Networks

No Author Given

No Institute Given

Abstract. Most real-world information networks, such as social networks, are heterogeneous and relations between different entities have different semantic meanings. Therefore techniques for link prediction in homogeneous networks can not be directly applied on heterogeneous ones. On the other hand, works that investigate link prediction in heterogeneous networks, do not consider the dynamics of networks in sequential time intervals. In this work we propose a technique that leverages a combination of latent and topological meta path-based features to predict a target relationship between two nodes of given types in a dynamic heterogeneous information network. Our results indicates that combining these features helps in building a more accurate predictive model compared to current techniques that use either of these features.

Keywords: Link prediction · Relationship prediction · Dynamic heterogeneous networks · Social networks · Network topology · Meta path.

1 Introduction

The goal of link prediction in a network graph [7] is to estimate the likelihood of the relationship between two nodes in future, based on the observed graph. Predicting such connections in a network have multiple applications such as friend/item/ad recommending, network completion, or biological applications such as predicting protein-protein interactions. Traditional link prediction techniques, such as [7], consider networks to be homogeneous, i.e., graphs with only one type of edges and nodes. However, most real-world networks, such as social networks, scholar networks, patient networks [4] and knowledge graphs [19] are heterogeneous information networks (HINs) [14] and have multiple node and relation types. For example, in a bibliographic network there are nodes of types authors, papers, and venues, and edges of types write, cite and publish.

In a HIN relations between different entities have different semantic meanings. Thus techniques for homogeneous networks can not be directly applied on heterogeneous ones. A few works such as [17, 15] investigated the problem of link/relationship prediction in HINs, however, they do not consider the dynamics of social networks and ignore analysis of sequence of network snapshots. On the other hand, it has been shown that for link prediction in homogenous networks incorporating temporal changes helps in a more accurate prediction [23]. Previous work on temporal link prediction scarcely studied HINs and to the best of our knowledge, the problem of relationship prediction for dynamic heterogeneous networks was not studied before.

In this work we study the problem of predicting relationships in a dynamic heterogeneous information network (DHIN) i.e., a network with different types of nodes and links associated with timestamps, which is stated as follows: *Given a DHIN graph G, how can we predict the future structure of G?*

Amin ▶*This section is done up to here! Contributions TBA.*◀

The main contributions of our work include:

- We present a technique, called `RelationPredict`, that predicts a target relationships between two nodes of given types;
- An evaluation of the accuracy and performance of the proposed algorithm on real social network data.

[23] [17] [16] [6] [20] [18] [15] [21] [7]

2 Problem Statement

The following definition is an extended version of HIN [17] for the dynamic network case.

Definition 1 (Dynamic heterogeneous information network). *A dynamic heterogeneous information network (DHIN) is a directed graph $G = (V, E)$ with a node type mapping function $\phi : V \to \mathcal{A}$ and a link type mapping function $\psi : E \to \mathcal{R}$, where V, E, $\mathcal{A}$, and $\mathcal{R}$ denote sets of nodes, links, node types, and relation types. Each node $v \in V$ belongs to a node type $\phi(v) \in \mathcal{A}$, each link $e \in E$ belongs to a relation $\psi(e) \in \mathcal{R}$, and $|\mathcal{A}| > 1$ and $|\mathcal{R}| > 1$. Also each edge $e = (u, v, t)$ is a temporal edge from a vertex u to a vertex v at time t.* □

The DBLP bibliographic network[1] is an example of a DHIN, containing different types of nodes such as papers, authors, topics, and publication venues, with publication links associated with date. Another example is the Twitter social network with nodes of types posted tweets, users, topics, and hashtags and time window associated with these tweets.

In the context of a heterogenous network, a *relation* can be in the form of a *direct link* or an *indirect link*, where an indirect link is a sequence of direct links in the network. Thus, two nodes might not be directly connected, however they might be connected considering the semantic of a sequence of links of different types. In this work, we use the terms *relationship prediction* and *link prediction* interchangeably referring to predicting whether two nodes will be connected in future via a *sequence of relations* in the graph, where the *length* of a sequence is greater than or equal to one. For instance in a bibliographic network a direct link exist between an author and a paper he wrote, and an indirect link exist between him and his co-authors through the paper, which they wrote together.

[1] `http://dblp.uni-trier.de/db/`

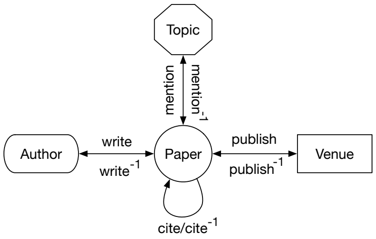


Fig. 1. Network schema for DBLP network.

In order to better understand different types of nodes and their relation in a network, the concept of *network schema* [17] is used. A network schema is a meta structure graph that summarizes a HIN and is formally defined as bellow.

Definition 2 (Network schema). *For a heterogeneous network $G = (V, E)$, the network schema $S_G = (\mathcal{A}, \mathcal{R})$ is a directed meta graph where $\mathcal{A}$ is the set of node types in V and $\mathcal{R}$ is the set of relation types in E.* □

Figure 1 shows the network schema for the DBLP bibliographic network with $\mathcal{A}$=\{*Author*, *Paper*, *Venue*, *Topic*\}. In this paper, we refer to different types of nodes in the DBLP bibliographic network with abbreviations P for paper, A for author, T for topic, and V for venue.

Similar to the notion of network schema that provides a meta structure for the network, a *meta path* [17] provides a meta structure for paths between different nodes in the network.

Definition 3 (Meta path). *A meta path P is a path in a network schema graph $S_G = (\mathcal{A}, \mathcal{R})$, denoted by $P(A_1, A_{n+1}) = A_1 \xrightarrow{R_1} A_2 ... \xrightarrow{R_n} A_{n+1}$, as a sequence of links between node types, which defines a composite relationship between a node of type A_1 and one of type A_{n+1}, where $A_i \subseteq \mathcal{A}$ and $R_i \subseteq \mathcal{R}$.* □

The *length* of a meta path is the number of relations in it. Note that given two node types A_i and A_j, there may exist multiple meta paths of different lengths between them. We call a path $p = (a_1 a_2 ... a_{n+1})$ a *path instance* of a meta path $P = A_1 - A_2 ... - A_{n+1}$ if p follows P in the corresponding HIN, i.e., for each node a_i in p, we have $\phi(a_i) = A_i$. The co-author relationship in DBLP can be described with the meta path $A \xrightarrow{write} P \xrightarrow{write^{-1}} A$ or in short A–P–A. Paths in thick solid lines in Figure 2 correspond to A–P–V–P–A meta paths between *Max* and *Ada*, indicating they published in the same venue, such as *Max–P1–KDD–P3–Ada*. Each meta path indicates a different semantic and defines a unique topology representing a special relation. The relationship between two authors are different in meaning when they are co-authors (A–P–A) versus one citing another's paper (A–P–P–A).

Meta Path-based Similarity Measures. Given a meta path $P = (A_i, A_j)$ and a pair of nodes a and b that $\phi(a) = A_i$ and $\phi(b) = A_j$, several *similarity*

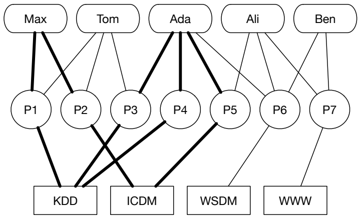


Fig. 2. An example of A–P–V–P–A meta paths between two authors Max and Ada.

measures can be defined between a and b based on the path instances of P. Examples of such similarity or proximity measures in a HIN are *path count* [17, 15], *PathSim* [17] or *normalized path count* [15], *random walk* [15], and *HeteSim* [13]. Given a meta path P a source node a and a target node b, path count measures the number of path instances between a and b following P, and can be calculated by multiplying adjacency matrices of relations in P [15], and random walk is the probability of the random walk that starts from a and ends with b following P. More discussion about these similarity measures can be found in [13, 15, 17]. Without loss of generality, in this work we use the path count as the default measure. For example given the meta path A–P–V–P–A and the HIN in Figure 2, $PC(Max, Ada)$=3 and $PC(Max, Tom)$=3.

We now formally define the problem that we target in this work as follows:

Definition 4 (Relationship prediction problem). *Given a DHIN graph G at time t, and a target relation meta path $P(A_i, A_j)$ between nodes of type A_i and A_j, we aim to predict the existence of a path instance of P between two given nodes of types A_i and A_j at time $t+1$.* □

3 Relationship Prediction Approach

Given a DHIN graph $G = (V, E)$, and the number of graph snapshots t, we first decompose G to a sequence of t HIN graphs $G_1, .., G_t$ based on links with associated timestamps. We then apply our techniques to predict relationships in G_{t+1}. As mentioned in Definition 4, we intend to predict existence of a given type of relationship (target meta path) between two given nodes. Thus we define a new type of graph, called *augmented reduced graph*, that is generated according to a given heterogeneous network and a target relation meta path.

Definition 5 (Augmented reduced graph). *Given a HIN graph $G = (V, E)$ and a target meta path $P(A_i, A_j)$ between nodes of type A_i and A_j, an augmented reduced graph $G^P = (V^P, E^P)$ is a graph, where $V^P \subseteq V$ and nodes in V^P are of type A_i and A_j, and edges in E^P indicates relationships of type P in G.* □

An augmented reduced graph for the network in Figure 2 and target meta path $P(A, A)$=A–P–V–P–A is a graph with nodes of type *Author* and edges that represent relationship of *publishing in the same venue*. For example (*Max*, *Ada*) is an edge in the corresponding augmented reduced graph because they both published at KDD and ICDM. If we consider meta path $P(A, A)$=A–P–A, the augmented reduced graph represents a co-authorship graph, where nodes are of type *Author* and edges, such as (*Max*, *Tom*), represent *co-authorship*.

3.1 Homogenize link prediction

Zhu et al. [23] studied the problem of temporal link prediction in the context of homogeneous networks, where the input is a sequence of graphs $G_1, ..., G_t$ and the output is the estimated G_{t+1}. They present a matrix factorization (MF) with block-coordinate gradient descent technique that for each adjacency matrix G_τ at time τ infers a low rank k-dimensional latent space matrix Z_τ that minimizes

$$\begin{aligned}&\underset{Z_1,..,Z_t}{\operatorname{argmin}} \sum_{\tau=1}^{t} \left\|G_\tau - Z_\tau Z_\tau^T\right\|_F^2 + \lambda \sum_{\tau=1}^{t} \sum_{u} (1 - Z_\tau(u) Z_{\tau-1}(u)^T)\\ &\text{subject to :} \forall u, \tau, Z_\tau \geq 0, Z_\tau(u) Z_\tau(u)^T = 1\end{aligned} \tag{1}$$

, where λ is a regularization parameter, and $(1 - Z_\tau(u) Z_{\tau-1}(u)^T)$ penalizes node u for suddenly changing its latent position. $Z_\tau(u)$ is a row vector denoting u's temporal latent space representation at time τ, and $Z_\tau(u, i)$ indicates the position of u in the i-th dimension at Z. The intuition behind their technique is that 1) nodes move smoothly in the latent space over time and it is less likely to have large moves [12, 22], and 2) user interactions are more likely to occur between similar users in a latent space representation. To predict adjacency matrix G_{t+1} they used $Z_t Z_t^T$, however, they mentioned that G_{t+1} can be formulated as $\varPhi(f(Z_1, ...Z_t))$, where $\varPhi$ and f are link and temporal functions that one may apply techniques such as nonparametric approaches [11] to learn them.

Algorithm 1 is an adaptation of the above MF technique applied on a sequence of augmented reduced graphs G_i^P (Definition 5) given a target meta path P, which changes equation (1) by replacing G_τ with G_τ^P. The algorithm gets as an input a DHIN graph G, the number of graph snapshots t, a target relation meta path $P(A, B)$, the latent space dimension k, and the link to predict (a, b) at $t + 1$. The algorithm first decomposes G into a sequence of t graphs $G_1, .., G_t$ by considering the associated timestamps on edges (line 1). Next from each graph G_i, a corresponding augmented reduced graph G_i^P is generated (lines 2-7) for which nodes are of type a and b (beginning and end of target meta path P). For example given $P(A, A)$=A–P–A, each G_i^P represents the co-authorship graph at time t. Finally the matrix factorization technique in [23] is applied (line 8) to infer latent spaces $Z_1, ..., Z_t$ and estimate G_{t+1}^P by $Z_t Z_t^T$ (line 9). Note that Z_τ depends on $Z_{\tau-1}$ as used in the temporal regularization term in equation (1).

3.2 Dynamic meta path-based relationship prediction

The above homogenize approach does not consider different semantics of meta paths between the source and destination nodes. In fact, Zhu et al. [23] assume that the probability of a link between nodes depends only on their latent

Algorithm. 1 Homogenize Link Prediction

```
Input: A DHIN graph $G$, the number of snapshots $t$, a target meta path $P(A,B)$, the latent space
    dimension $k$, the link to predict $(a,b)$ at $t+1$
Output: The probability of existence of link $(a,b)$ in $G_{t+1}^P$
1: $\{G_1,..,G_t\} \leftarrow DecomposeGraph(G,t)$
2: for each graph $G_i = (V_i, E_i)$ do
3:     for each node $x \in V_i$ that $\phi(x) = A$ do
4:         Follow $P$ to reach a node $y \in V_i$ that $\phi(y) = B$
5:         Add nodes x and y, and edge $(x,y)$ to the augmented reduced graph $G_i^P$
6:     end for
7: end for
8: $\{Z_1,..,Z_t\} \leftarrow MatrixFactorization(G_1^P,..,G_t^P,k)$
9: Return $Pr((a,b) \in E_{t+1}^P) \leftarrow \sum_{i=1}^k Z_t(a,i)Z_t(b,i)$
```

Algorithm. 2 Dynamic Meta path-based Relationship Prediction

```
Input: A DHIN graph $G$, the number of snapshots $t$, a network schema $S$, a target meta path
    $P(A,B)$, the maximum length of a meta path $l$, the latent space dimension $k$, the link to
    predict $(a,b)$ at $t+1$
Output: The probability of existence of link $(a,b)$ in $G_{t+1}^P$
1: $\{G_1,..,G_t\} \leftarrow DecomposeGraph(G,t)$
2: Generate target augmented reduced graphs $G_1^P,..,G_t^P$ following Algorithm 1 lines 2-7
3: $\{P_1,..,P_n\} \leftarrow GenerateMetaPaths(S, P(A,B), l)$
4: $\{Z_1,..,Z_t\} \leftarrow MatrixFactorization(G_1^P,..,G_t^P,k)$
5: $\hat{G}_t^P \leftarrow Z_{t-1}Z_{t-1}^T$ and $\hat{G}_{t+1}^P \leftarrow Z_t Z_t^T$
6: for each pair $(x,y)$, where $x \in V_{t-1}^P$ and $y \in N(x)$ is a nearby neighbor of $x$ in $G_{t-1}^P$ do
7:     Add the feature vector $\langle f_{t-1}^{P_1}(x,y), f_{t-1}^{P_2}(x,y), ..., f_{t-1}^{P_n}(x,y), \hat{G}_t^P(x,y)\rangle$ to the training set $T$
       with $label$=1 if $(x,y)$ is a new link in $E_t^P$ otherwise $label$=0.
8: end for
9: $model \leftarrow Train(T)$
10: Return $Pr((a,b) \in E_{t+1}^P) \leftarrow Test(model, \langle f_t^{P_1}(a,b), f_t^{P_2}(a,b), ..., f_t^{P_n}(a,b), \hat{G}_{t+1}^P(a,b)\rangle)$
```

positions. On the other hand, Sun et al. [15] proposed a supervised learning framework, called *PathPredict*, that uses meta path-based features in a past time interval to predict the relationship building in a future time interval. It learns coefficients associated with features by maximizing the likelihood of new relationship formation. However, their model is learned based on one past interval and does not consider temporal changes as in [23]. Our intuition is that leveraging meta path-based features along with latent space features can help to boost the prediction accuracy. In other word, we combine latent space features with topological meta path-based features in our predictive model.

Algorithm 2 gets as an input a DHIN graph G, the number of graph snapshots t, a network schema S, a target relation meta path $P(A,B)$, the maximum length of a meta path l, the latent space dimension k, and the link to predict (a,b) at $t+1$. Same as Algorithm 1, it decomposes G into a sequence of graphs (line 1). Next it generates augmented reduced graphs G_i^Ps from G_is based on P for which nodes are of type A and B (beginning and end of meta path P)(line 2) as explained in Algorithm 1. It then produces the set of all meta paths between nodes of type A and type B defined in $P(A,B)$ (line 3). This is done by traversing the network schema S (for instance through a BFS traversal) and generating meta paths with the maximum length of l. It then applies the MF technique [23] to find latent space matrices Z_i (line 4) and using them calculates the estimated augmented reduced graph $\hat{G}^P$ at times t and $t+1$ (line 5). The last steps are creating a training dataset with sample pairs (x, y) with feature set containing $\hat{G}^P_t(x, y)$, and meta path-based feature measures $f^{P_i}_t(x, y)$ based on meta path P_i at time t, and *label*=1 if (x, y) is a new link in G^P_{t+1} otherwise *label*=0 (lines 6-8), training the predictive model (line 9), generating features for the given pair (a, b) and testing it using the trained model (line 10). In the following section we explain our learning technique in detail.

Combining latent and meta path-based features. Our hypothesis is that combining latent with topological features can increase the prediction accuracy as we can learn latent features that fit the residual of meta path-based features. However, if the latent features learn similar structure to the topological features, then mixing them may not be beneficial. One way to do so is by changing the loss function in Equation (1) to $\underset{\boldsymbol{\theta_\tau}, Z_\tau}{\operatorname{argmin}} \sum_{\tau=1}^{t} \left\| G^P_\tau - \Phi(Z_\tau Z^T_\tau + \sum_{i=1}^{n} \theta_{i_{\tau-1}} \mathcal{F}^{P_i}_{\tau-1}) \right\|^2_F$ and add another regularization term $\lambda \sum_{\tau=1}^{t} \sum_{i=1}^{n} \theta^2_{i_\tau}$, where n is the number of meta path-based features, $\mathcal{F}^{P_i}$ is the i-th meta path-based feature matrix defined on G_i, and θ_i is weights for feature f_i. Although the gradient descent algorithm used in the MF technique to infer latent space matrices in [23] is fast, it cannot be efficiently applied to the changed loss function. This is because it requires computing meta paths for all possible pairs of nodes in $\mathcal{F}^{P_i}$ for all snapshots, which is not scalable as calculating similarity measures such as PathCount or PathSim can be very costly. For example computing path counts for A–P–V–P–A meta path, can be done by multiply adjacency matrices $AP \times PV \times VP \times PA$.

As an alternative solution we build a predictive model that considers a linear combination of topological and latent features. These features, however, can be combined in different ways that is beyond the scope of this work. Given the training pairs of nodes and their corresponding meta path-based and latent features, we apply logistic regression to learn the weights associated with these features. We define the probability of forming a *new link* in future from node a to b as $Pr(label = 1|a, b; \boldsymbol{\theta}) = \frac{1}{e^{-z}+1}$, where $z = \sum_{i=1}^{n} \theta_i f^{P_i}(a, b) + \sum_{j=1}^{k} \theta_{n+j} Z(a, j) Z(b, j)$, and $\theta_1, \theta_2, ..., \theta_n$ and $\theta_{n+1}, \theta_{n+2}, ..., \theta_{n+k}$ are associated weights for meta path-based features and latent features at current time between a and b. Given a training dataset with l instance-label pairs, we use logistic regression with L_2 regularization to estimate the optimal $\boldsymbol{\theta}$ as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{l} -logPr(label|a_i, b_i; \boldsymbol{\theta}) + \lambda \sum_{j=1}^{n+k} \theta^2_j \tag{2}$$

We preferred combining features in this learning framework since G_i is very sparse and thus the number of newly formed links are much less compared to all possible links. Consequently calculating meta path-based features for training dataset is scalable compared to the MF technique. Moreover, similar to [15], in order to avoid excessive computing of meta path-based measures between nodes that might not be related, we confine samples to pairs that are located in a nearby neighborhood. More specifically, for each source node x in G_i^P, we choose target nodes that are within 3-hop of x but not in 1-hop, i.e, are not connected to x in G_i^P. We first find all target nodes that make a new relationship with x in G_{i+1}^P and label respective samples as positive. Next we sample an equal number of negative pairs, i.e., those targets that do not make new connection, to balance our training set. Once the dataset is built, we perform logistic regression to learn the model and then apply the predictive model to the feature vector $\langle f_t^{P_1}(a,b), f_t^{P_2}(a,b), ..., f_t^{P_n}(a,b), \hat{G}_{t+1}^P(a,b)\rangle$. The output probability can be later interpreted as a binary value based on a user defined cut-off.

3.3 Implementation

Amin ▶*add system and OS spec*◀ We use the implementation of temporal latent space inference for a sequence of dynamic graph snapshots [2][23]. For the classification part, we use the efficient LIBLINEAR [5] package[3] and set the type of solver to L2-regularized logistic regression (primal). We performed 5-fold cross validation for the training phase.

4 Experiments

4.1 Experiment Setting

Dataset. We conduct our experiments on two real-world datasets that have different characteristics and evolution behaviour.

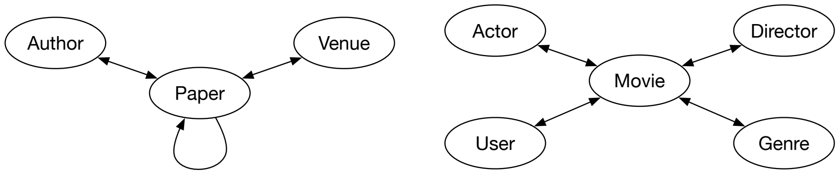
- *Publications dataset:* The *aminer* citation dataset[4] V8 (2016-07-14) is extracted from DBLP, ACM, and other sources. It contains 3,272,991 papers and 8,466,859 citation relationships for 1,752,443 authors, who published in 10,436 venues, from 1930 to 2016. Each paper is associated with abstract, authors, year, venue, and title. Amin ▶*We consider only those papers published since 1996, which includes X papers and Y authors.*◀ Authors in [15] used a similar dataset but considered only authors with more than 5 publications. We generate two datasets: one that contains all publications, and one that considers authors with at least 5 papers. We consider $k = 3, 5, and 10$ different time intervals for the dynamic analysis. In our evaluation, we execute the learned model on the last interval to measure the prediction accuracy.
- *Movies dataset:* The RecSys HetRec 2011 movie data set [1] is an extension of MovieLens10M dataset, published by GroupLeans research group [5] that

[2] `https://github.com/linhongseba/Temporal-Network-Embedding`

[3] `https://github.com/cjlin1/liblinear`

[4] `https://aminer.org/citation`

[5] `http://www.grouplens.org`



(a) DBLP

(b) IMDB

Fig. 3. The simplified network schema used for our experiments.

links the movies of MovieLens dataset with their corresponding web pages at Internet Movie Database (IMDb[6]) and Rotten Tomatoes[7] movie review systems. It contains information of 2,113 users, 10,197 movies, 20 movie genres (avg. 2.04 genres per movie), 4,060 directors, 95,321 actors (avg. 22.78 actors per movie), 72 countries, 855,598 ratings (avg. 404.92 ratings per user, and avg. 84.64 ratings per movie), and 13,222 tags (avg. 22.69 tas per user, avg. 8.12 tas per movie).

Movies once release, users can rate them but a paper is published once and new co-authorship is made only at that time.

In co-authorship all connections are new in MovieDB new connections to an existing movie. This is a common problem with all rating datasets.

We also conduct our experiments on two variations of the DBLP, one with min 5paper as in ... and one with all.

Meta paths and taget relationships. We consider different type of target relationship.

Network schema for the two datasets are shown in Figure X. Note that we consider a simplified version and ignore nodes such as topic fro papers or tag for movies.

We consider .... based on the work in ... we con

Authors in [15] conducted Wald test in a case study and found that the p-value for the feature associated with each meta path and their significance level. From the results, we can see that the shared co-authors, shared venues, shared topics and co-cited papers for two authors all play very significant roles in determining their future collaboration(s). For...

Similarly we only calculated the PC for these meta paths. Note that the goal of our paper is not to select the best features but to show the strength of using...

Table 1 shows meta paths between authors under length 4 for the publications dataset.

Unlike the A–P–A target relation for the publication dataset for which both ends of the relation is of the same kind, we consider U–M as the target meta

[6] http://www.imdb.com

[7] http://www.rottentomatoes.com

Table 1. Publications dataset meta paths (A=author, P=paper, A=venue).

Meta path	**Meaning**
A–P–A	[*The target relation*] Authors are coauthors
A–P–V–P–A	Authors publish in the same venue
A–P–A–P–A	Authors have the same co-author
A–P–P–P–A	Authors cite the same papers

path for the movie dataset to show the effectiveness of our proposed methods in predicting such relationships. Amin ▶*The issue with matrix factorization is that originally G_{n*n} is for homogenous network with the same type of nodes. In our case ZZ^T vs. VU^T* ◀

Table 2. Movies dataset meta paths (U=user,M=movie,A=actor,D=director,G=genre).

Meta path	**Meaning**
U–M	[*The target relation*] A user watches a movie
U–M–A–M	A user watches a movie with the same actor
U–M–D–M	A user watches a movie with the same director
U–M–G–M	A user watches a movie of the same genre
U–M–U–M	A user watches a movie that another user

Baseline methods. Considering the effect of time-wise data decomposition. What if we shorten timespans of each G_t? The extreme is having only one graph or having it for each year. Can we find a trade-off?

- Heterogeneous non-temporal (PathCount, PathSim, NormalPathCount, RandomWalk, SymmetricRandomWalk)
- Homogeneous non-temporal (Katz, Jaccard)
- Homogeneous temporal (BCGD)

Evaluation Metrics. We use prediction error to evaluate the inference accuracy. Given the training graph G1, . . . , Gt, prediction error is defined as... Therefore, a smaller prediction error indicates better inference accuracy. For link prediction accuracy, we use Area Under Curves (both Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves), termed as AUCROC and AUCPR [3]. Also in order to decide which classifier has a lower error rate, we perform McNemar's test that assess the significance of the difference between two correlated proportions.

4.2 Results and Findings

Amin ▶*One reason that A–P–V–P–A is better with intervals is that one may publish in KDD but there are so many publishing there....*◀

4.3 Discussion

Our proposed technique can also be used in other applications. For example link recommendation

predicting missing edges in graphs.

Vertex Recommendation similar to [10]

In this work we modelled the predicted graph $\hat{G}_\tau(i, j)$ as a combination of meta path features and latent features $\Phi(z_i^T z_j + f_D(z_{i,j}; w))$. As explained in [8], one may also augment the model by incorporating some information regarding node affinities using implicit/explicit attributes and define node features x_i, which makes the model $\hat{G}_\tau(i, j) = \Phi(z_i^T z_j + f_D(z_{i,j}; w)$

As shown in [8] and [23], latent features are more predictive of linking behaviour compared to unsupervised scoring techniques such as Katz, PrefferentailAttachemnet, and Adamic.

Experiments in [8] shows combining the latent structure and side-information increases the prediction accuracy.

Amin ▶*TODO*◀ Connection to link privacy research such as [9]

From: Target Defense Against Link-Prediction-Based Attacks via Evolutionary Perturbations

Fard et al. [24] assumed that all links in a network are sensitive, and they proposed to apply subgraph-wise perturbations onto a directed network, which randomize the destination of a link within some subnetworks thereby limiting the link disclosure. Furthermore, they proposed neighborhood randomization to probabilistically randomize the destination of a link within a local neighborhood on a network [25]. It should be noted that both subnetwork-wise perturbation and neighborhood randomization perturb every link in the network based on a certain probability.

As discussed above, link prediction can be applied to predict the potential relationship between two individuals. From another perspective, it may also increase the risk of link disclosure. Even if the data owner removes sensitive links from the published network dataset, it may still be disclosed by link prediction and consequently lead to privacy breach. Michael et al. [26] presented a link reconstruction attack, which is a method that attacker can use link prediction to infer a user?s connections to others with high accuracy, but they did not mention how to defend the so-called link-reconstruction attack. Naturally, one can consider finding a way to prevent the link-prediction attack. Since link-reconstruction attack or link-prediction-based attack aims to find out some real but unobservable links, the defense of link-prediction-based attacks is also target-directed, which means that one has to preserve the targeted links from being predicted. In the literature, most existing approaches on link prediction are based on the similarity between pairwise nodes under the assumption that the more similar a pair of nodes are, the more likely a link exists between them.

In general, since many link prediction algorithms are designed based on network structures, a data owner can add perturbations into the original network to reduce the risk of targeted-link disclosure due to link-prediction-based attacks

From: SmartWalk: Enhancing Social Network Security via Adaptive Random Walks

Extensive research has been carried out to protect the privacy of trust relationships between any pair of users (link privacy) [19, 20, 50, 54, 33, 27]. The challenge of preserving link privacy lies in causing no significant losses on the utility of applications that leverage the social trust relationships. Specifically, link privacy is preserved by adding extra noise to the local structure of a social network. At the same time, global structural characteristics are maintained to ensure that the utility of the social network is not severely reduced. This can be implemented by replacing a real link between two users with a fake link generated by a random walk[33]. Link privacy/utility trade-off. Mittal et al. in [33] considered that the length of random walks for all nodes has a fixed value. As the length increases (more noise), the perturbed social graph converges to a random graph and its utility declines drastically. Our key insight is that instead of adding identical amount of noise to all users, perturbation can be unevenly distributed according to the local mixing time such that privacy can be protected with less perturbation on average. In other words, we can perform nodeadaptive random walks rather than random walks with a fixed length for every user when generating fake links. [33] P. Mittal, C. Papamanthou, and D. Song. Preserving link privacy in social network based systems. In NDSS, 2013

5 Related Work

[23] [17] [16] [6] [20] [18] [15] [21] [7]

Sun et al. proposed PathSelClus [18] that uses limited guidance from users in the form of seeds in some of the clusters and automatically learn the best weights for each meta-path in the clustering process.

The concept of temporal smoothness has been used in evolutionary clustering [36] and link prediction in a dynamic network [23].

There exist many embedding methods for static networks, however very few considered dynamic networks. Zhu et al. [23] attempt dynamic link prediction by adding a temporal-smoothing regularization term to a non-negative matrix factorization objective. Their goal is to reconstruct the adjacency matrix of different time-stamps of a graph. They use a Block-Coordinate Gradient Descent (BCGD) algorithm to perform non-negative factorization. Their formulation is almost identical to the algorithm of Chi et al. [2], who perform evolutionary spectral clustering that captures temporal smoothness. Because matrix factorization provides embedding vectors of the nodes for each time-stamp, the factorization by-product from this work can be considered as dynamic network embeddings.

6 Conclusions and Future Work

TBA.

References

1. Cantador, I., Brusilovsky, P., Kuflik, T.: 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In: Proceedings of the 5th ACM conference on Recommender systems. RecSys 2011, ACM, New York, NY, USA (2011)
2. Chi, Y., Song, X., Zhou, D., Hino, K., Tseng, B.L.: Evolutionary spectral clustering by incorporating temporal smoothness. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 153–162. KDD '07, ACM, New York, NY, USA (2007). https://doi.org/10.1145/1281192.1281212, http://doi.acm.org/10.1145/1281192.1281212
3. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd international conference on Machine learning. pp. 233–240. ACM (2006)
4. Denny, J.C.: Mining electronic health records in the genomics era. PLoS computational biology **8**(12), e1002823 (2012)
5. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of machine learning research **9**(Aug), 1871–1874 (2008)
6. Huang, Z., Zheng, Y., Cheng, R., Sun, Y., Mamoulis, N., Li, X.: Meta structure: Computing relevance in large heterogeneous information networks. In: KDD'16 (2016)
7. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. journal of the Association for Information Science and Technology **58**(7), 1019–1031 (2007)
8. Menon, A.K., Elkan, C.: Link prediction via matrix factorization. In: Joint european conference on machine learning and knowledge discovery in databases. pp. 437–452. Springer (2011)
9. Milani Fard, A., Wang, K.: Neighborhood randomization for link privacy in social network analysis. World Wide Web **18**(1), 9–32 (2015)
10. Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1105–1114. ACM (2016)
11. Sarkar, P., Chakrabarti, D., Jordan, M.I.: Nonparametric link prediction in dynamic networks. In: Proceedings of the 29th International Conference on Machine Learning. pp. 1897–1904. ICML'12, Omnipress, USA (2012)
12. Sarkar, P., Moore, A.W.: Dynamic social network analysis using latent space models. ACM SIGKDD Explorations Newsletter **7**(2), 31–40 (2005)
13. Shi, C., Kong, X., Huang, Y., Philip, S.Y., Wu, B.: Hetesim: A general framework for relevance measure in heterogeneous networks. IEEE Trans. Knowl. Data Eng. **26**(10), 2479–2492 (2014)
14. Shi, C., Li, Y., Zhang, J., Sun, Y., Philip, S.Y.: A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering **29**(1), 17–37 (2017)
15. Sun, Y., Barber, R., Gupta, M., Aggarwal, C.C., Han, J.: Co-author relationship prediction in heterogeneous bibliographic networks. In: Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining. pp. 121–128. ASONAM '11, IEEE Computer Society (2011)

16. Sun, Y., Han, J., Aggarwal, C.C., Chawla, N.V.: When will it happen?: Relationship prediction in heterogeneous information networks. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining. pp. 663–672. WSDM '12, ACM, New York, NY, USA (2012)
17. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In: Proceedings of the VLDB Endowment 4 (11). pp. 992–1003. VLDB Endowment (2011)
18. Sun, Y., Norick, B., Han, J., Yan, X., Yu, P.S., Yu, X.: Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. ACM Transactions on Knowledge Discovery from Data (TKDD) **7**(3), 11 (2013)
19. Wang, C., Song, Y., El-Kishky, A., Roth, D., Zhang, M., Han, J.: Incorporating world knowledge to document clustering via heterogeneous information networks. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1215–1224. ACM (2015)
20. Wang, C., Sun, Y., Song, Y., Han, J., Song, Y., Wang, L., Zhang, M.: Relsim: Relation similarity search in schema-rich heterogeneous information networks (2016)
21. Yang, Y., Chawla, N., Sun, Y., Hani, J.: Predicting links in multi-relational and heterogeneous networks. In: 2012 IEEE 12th International Conference on Data Mining. pp. 755–764 (Dec 2012)
22. Zhang, J., Wang, C., Wang, J., Yu, J.X.: Inferring continuous dynamic social influence and personal preference for temporal behavior prediction. Proceedings of the VLDB Endowment **8**(3), 269–280 (2014)
23. Zhu, L., Guo, D., Yin, J., Steeg, G.V., Galstyan, A.: Scalable temporal latent space inference for link prediction in dynamic social networks. IEEE Transactions on Knowledge and Data Engineering (TKDE) **28**(10), 2765–2777 (Oct 2016)