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 cannot be directly applied on heterogeneous ones. On the other hand, works that investigate link prediction in heterogeneous networks, do not necessarily consider network dynamism 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 experiment results on two real-world information network datasets show up to X% improvement in prediction accuracy compared to the state of the art techniques.

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 [18] is to estimate the likelihood of future relationship between two nodes based on the observed graph. Predicting such connections in a network have multiple applications such as recommendation systems [3, 29, 20, 17, 11], network reconstruction [10], node classification [9], or biomedical applications such as predicting protein-protein interactions [15]. Traditional link prediction techniques, such as [18], consider networks to be homogeneous, i.e., graphs with only one type of nodes and edges. However, most real-world networks, such as social networks, scholar networks, patient networks [6] and knowledge graphs [36] are heterogeneous information networks (HINs) [28] 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 writes, cites and publishes.

In a HIN relations between different entities carry different semantics. For instance the relationship between two authors are different in meaning when they are co-authors compared to the case that one cites another's paper. Thus techniques for homogeneous networks [18, 35, 19, 16, 1] cannot be directly applied on heterogeneous ones. A few works such as [32, 30] investigated the problem of link/relationship prediction in HINs, however, they do not consider the dynamism of networks and overlook the potential benefits of analyzing a heterogeneous graph as a sequence of network snapshots. To this end, existing work has already shown that in homogenous networks incorporating temporal changes improves link prediction accuracy [44]. Previous work on temporal link prediction scarcely studied HINs and to the best of our knowledge, the problem of predicting relationships in dynamic heterogeneous networks has not been studied before. A dynamic heterogeneous information network (DHIN) is a HIN, where links are associated with timestamps.

In this work we study the problem of relationship prediction in a DHIN that is stated as: *Given a DHIN graph G at t consecutive time intervals, we are interested in predicting the existence of a particular relationship/path between two given nodes at time* $t+1$. The major challenge in relationship prediction in DHINs is how to effectively combine the HIN topology features and inferred latent features that incorporate temporal changes in order to exhibit the best performance. Also, the prediction technique should be computationally efficient for large-scale networks. To this end, the main contributions of our work include:

- We propose the problem of relationship prediction in a DHIN, and draw a contrast between this problem and existing link prediction techniques that have been proposed for dynamic and heterogeneous networks;
- We present a simple yet effective technique, called MetaDynaMix, that leverages topological meta path-based and latent features to predict a target relationship between two nodes in a DHIN;
- We empirically evaluate the efficacy and accuracy of our proposed work on two real-world datasets, and the results show up to X% improvement compared to the state of the art baselines.

In the rest of the paper, we introduce the preliminaries and problem statement in Section 2, discuss our solutions to the relationship prediction problem in Section 3, explain the details of our empirical experimentation and findings in Section 4, review the related work in Section 5, and finally conclude the paper.

2 Problem Statement

Our work is focused on heterogeneous information networks (graphs) that can change and evolve over time. As such, we first formally define the concept of *Dynamic Heterogeneous Information Networks*, as follows:

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 networkis an example of a DHIN, containing different types of nodes such as papers, authors, topics, and publication venues, with

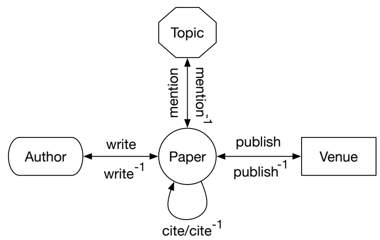


Fig. 1. Network schema for DBLP network.

publication links associated with date. 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 the 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 exists between an author and a paper she wrote, and an indirect link exists between her and her co-authors through the paper, which they wrote together. In order to better understand different types of nodes and their relation in a network, the concept of *network schema* [32] is used. A network schema is a meta graph structure graph that summarizes a HIN and is formally defined as follows:

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.* $\square$

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* [32] provides a meta structure for paths between different node types in the network.

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

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 $\mathcal{P} = A_1 - A_2 ... - A_{n+1}$ if p follows $\mathcal{P}$ in the corresponding HIN, i.e.,

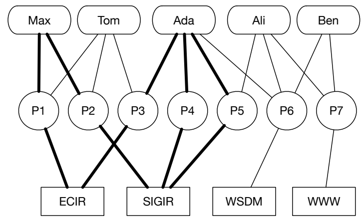


Fig. 2. An example of A–P–V–P–A meta paths between two authors Max and Ada.

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–ECIR–P3–Ada*. Each meta path carries different semantics and defines a unique topology representing a special relation.

Meta Path-based Similarity Measures. Given a meta path $\mathcal{P} = (A_i, A_j)$ and a pair of nodes a and b such that $\phi(a) = A_i$ and $\phi(b) = A_j$, several *similarity measures* can be defined between a and b based on the path instances of $\mathcal{P}$. Examples of such similarity or proximity measures in a HIN are *path count* [32, 30], *PathSim* [32] or *normalized path count* [30], *random walk* [30], *HeteSim* [27], and *KnowSim* [37]. Without loss of generality, in this work we use path count as the default similarity 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 $\mathcal{P}(A_i, A_j)$ between nodes of type A_i and A_j, we aim to predict the existence of a path instance of $\mathcal{P}$ between two given nodes of types A_i and A_j at time $t+1$.* □

3 Proposed Relationship Prediction Approach

Given a DHIN graph $G = (V, E)$, we first decompose G to a sequence of t HIN graphs $G_1, .., G_t$ based on links with associated timestamps and then 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 $\mathcal{P}(A_i, A_j)$ between nodes of type A_i and A_j, an augmented reduced graph $G^{\mathcal{P}} = (V^{\mathcal{P}}, E^{\mathcal{P}})$ is a graph, where $V^{\mathcal{P}} \subseteq V$ and nodes in $V^{\mathcal{P}}$ are of type A_i and A_j, and edges in $E^{\mathcal{P}}$ indicate relationships of type $\mathcal{P}$ in G.* □

For example, an augmented reduced graph for the network in Figure 2 and target meta path $\mathcal{P}(A, A)$=A–P–V–P–A is a graph whose nodes are of type *Author* and whose edges represent *publishing in the same venue*.

3.1 Homogenized link prediction

Once the given DHIN graph $G = (V, E)$ is decomposed to t HIN graphs $G_1, .., G_t$, one solution to the relationship prediction problem (Definition 4) is to build an augmented reduced graph $G_i^{\mathcal{P}}$ for each G_i with respect to the given target meta path $\mathcal{P}$ and then predict a link in $G_i^{\mathcal{P}}$ instead of a path in G_i. In other words, we generate a homogenized version of a graph snapshot and apply a link prediction method. The intuition behind considering different snapshots, i.e., a dynamic network, rather than a single snapshot for link prediction is that we can incorporate network evolution patterns to increase prediction accuracy. Our hypothesis is that the estimated graph $\hat{G}_{i+1}^{\mathcal{P}}$ dependents on $\hat{G}_i^{\mathcal{P}}$. Inspired by Zhu et al. [44], we formulate our problem as follows: Given a sequence of augmented reduced graphs $G_1^{\mathcal{P}}, .., G_t^{\mathcal{P}}$, we aim to infer a low rank k-dimensional latent space matrix Z_i for each adjacency matrix $G_i^{\mathcal{P}}$ at time i by minimizing

$$\begin{aligned} &\underset{Z_1,..,Z_t}{\operatorname{argmin}} \sum_{i=1}^{t} \left(\left\| G_i^{\mathcal{P}} - Z_i Z_i^T \right\|_F^2 + \lambda \sum_{x \in V^{\mathcal{P}}} (1 - Z_i(x) Z_{i-1}(x)^T) \right) \\ &\text{subject to} : \forall x \in V^{\mathcal{P}}, i, Z_i \geq 0, Z_i(x) Z_i(x)^T = 1 \end{aligned} \tag{1}$$

where $Z_i(x)$ is a temporal latent vector for node x at time i, $Z_i(x, j)$ indicates the position of x in the j-th dimension, λ is a regularization parameter, and $1 - Z_i(x)Z_{i-1}(x)^T$ penalizes sudden latent position changes for x. This optimization problem can be solved using a gradient descent technique. The intuition behind the above formulation is that nodes with similar latent space representations are more likely to connect, and also nodes move smoothly in the latent space over time and it is less likely to have abrupt moves [42]. The matrix $G_{t+1}^{\mathcal{P}}$ can be then estimated by $\Phi(f(Z_1, ...Z_t))$, where Φ and f are link and temporal functions, or simply by $Z_t Z_t^T$.

Algorithm 1 takes as input a DHIN graph G, the number of graph snapshots t, a target relation meta path $\mathcal{P}(A, B)$, the latent space dimension k, and the link to predict (a, b) at $t + 1$. It 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 $G_i^{\mathcal{P}}$ is generated (lines 2-7) for which nodes are of type a and b (beginning and end of target meta path $\mathcal{P}$). For example given $\mathcal{P}(A, A)$=A–P–A, each $G_i^{\mathcal{P}}$ represents the co-authorship graph at time i. Finally by optimizing the Equation (1) it infers latent spaces $Z_1, ..., Z_t$ (line 8) and estimate $G_{t+1}^{\mathcal{P}}$ by $Z_t Z_t^T$ (line 9). Note that Z_i depends on Z_{i-1} as used in the temporal regularization term in Equation (1).

3.2 Dynamic meta path-based relationship prediction

The above homogenized approach does not consider different semantics of meta paths between the source and destination nodes and assumes that the probability

Algorithm. 1 Homogenized Link Prediction

Input: A DHIN graph G, the number of snapshots t, a target meta path $\mathcal{P}(A, B)$, the latent space dimension k, the link to predict (a, b) at $t + 1$
Output: The probability of existence of link (a, b) in $G^{\mathcal{P}}_{t+1}$
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 $\mathcal{P}$ 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 $G^{\mathcal{P}}_i$
6:     **end for**
7: **end for**
8: $\{Z_1, .., Z_t\} \leftarrow MatrixFactorization(G^{\mathcal{P}}_1, .., G^{\mathcal{P}}_t, k)$
9: Return $Pr((a, b) \in E^{\mathcal{P}}_{t+1}) \leftarrow \sum_{i=1}^{k} Z_t(a, i)Z_t(b, i)$

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 $\mathcal{P}(A, B)$, 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 $G^{\mathcal{P}}_{t+1}$
1: $\{G_1, .., G_t\} \leftarrow DecomposeGraph(G, t)$
2: Generate target augmented reduced graphs $G^{\mathcal{P}}_1, .., G^{\mathcal{P}}_t$ following Algorithm 1 lines 2-7
3: $\{\mathcal{P}_1, .., \mathcal{P}_n\} \leftarrow GenerateMetaPaths(S, \mathcal{P}(A, B), l)$
4: $\{Z_1, .., Z_t\} \leftarrow MatrixFactorization(G^{\mathcal{P}}_1, .., G^{\mathcal{P}}_t, k)$
5: **for** each pair (x, y), where $x \in V^{\mathcal{P}}_{t-1}$ and $y \in N(x)$ is a nearby neighbor of x in $G^{\mathcal{P}}_{t-1}$ **do**
6:     Add feature vector $\langle f^{\mathcal{P}_i}_{t-1}(x, y)$ for $i = 1..n$, $Z_{t-1}(x, j)Z_{t-1}(y, j)$ for $j = 1..k)\rangle$ to the training set T with *label*=1 if (x, y) is a new link in $E^{\mathcal{P}}_t$ otherwise *label*=0.
7: **end for**
8: $model \leftarrow Train(T)$
9: Return $Pr((a, b) \in E^{\mathcal{P}}_{t+1}) \leftarrow Test(model, \langle f^{\mathcal{P}_i}_t(a, b)$ for $i = 1..n$, $Z_t(a, j)Z_t(b, j)$ for $j = 1..k\rangle)$

of a link between nodes depends only on their latent features. Our intuition is that leveraging meta path-based features along with latent space features can help to boost prediction accuracy. To this end, we combine latent space features with topological meta path-based features in our model.

Algorithm 2 takes as input a DHIN graph G, the number of graph snapshots t, a network schema S, a target relation meta path $\mathcal{P}(A, B)$, the maximum length of a meta path l, the latent space dimension k, and the link to predict (a, b) at $t + 1$. Similar to Algorithm 1, it decomposes G into a sequence of graphs (line 1). Next it generates augmented reduced graphs $G^{\mathcal{P}}_i$s from G_is based on $\mathcal{P}$ for nodes which are of type A and B (beginning and end of meta path $\mathcal{P}$) (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 $\mathcal{P}(A, B)$ (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 matrix factorization to find latent space matrices Z_i (line 4). The last steps create a training dataset for sample pairs (x, y) with feature set containing meta path-based measures $f^{\mathcal{P}_i}_t(x, y)$ for each meta path $\mathcal{P}_i$, and latent features $Z_t(a, j)Z_t(b, j)$ for $j = 1..k$ at time t, and *label*=1 if (x, y) is a new link in $G^{\mathcal{P}}_{t+1}$ otherwise *label*=0 (lines 5-7), subsequently training the predictive model (line 8), generating features for the given pair (a, b) and testing it using the trained model (line 9). 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 prediction accuracy as we can learn latent features that fit the residual of meta path-based features. One way to combine these features is to incorporate meta path measures in Equation (1) by changing the loss function and regularization term as

$$\begin{aligned} &\underset{\boldsymbol{\theta_i}, Z_i}{\operatorname{argmin}} \sum_{i=1}^{t} \left\| G_i^{\mathcal{P}} - (Z_i Z_i^T + \sum_{i=1}^{n} \theta_{i_{i-1}} \mathcal{F}_{i-1}^{\mathcal{P}_i}) \right\|_F^2 + \\ &\lambda \sum_{i=1}^{t} \left(\sum_{x \in V^{\mathcal{P}}} (1 - Z_i(x) Z_{i-1}(x)^T) + \sum_{i=1}^{n} \theta_{i_i}^2 \right) \end{aligned} \tag{2}$$

where n is the number of meta path-based features, $\mathcal{F}^{\mathcal{P}_i}$ is the i-th meta path-based feature matrix defined on G_i, and θ_i is the weight for feature f_i. Although we can use a fast block-coordinate gradient descent [44] to infer Z_is, it cannot be efficiently applied to the above changed loss function. This is because it requires computing meta paths for all possible pairs of nodes in $\mathcal{F}^{\mathcal{P}_i}$ 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 interpolated 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 time $t + 1$ from node a to b as $Pr(label = 1 | a, b; \boldsymbol{\theta}) = \frac{1}{e^{-z}+1}$, where $z = \sum_{i=1}^{n} \theta_i f_t^{\mathcal{P}_i}(a, b) + \sum_{j=1}^{k} \theta_{n+j} Z_t(a, j) Z_t(b, j)$, 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 time t 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

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{l} -logPr(label | a_i, b_i; \boldsymbol{\theta}) + \lambda \sum_{j=1}^{n+k} \theta_j^2 \tag{3}$$

We preferred to combine 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 the training dataset is scalable compared to the matrix factorization technique. Moreover, similar to [30], in order to avoid excessive computation 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 $G_i^{\mathcal{P}}$, we choose target nodes that are within 2-hop of x but not in 1-hop, i.e, are not connected to x in $G_i^{\mathcal{P}}$. We first find all target nodes that make a new relationship with x in $G_{i+1}^{\mathcal{P}}$ 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, in order 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 for the target link. The output probability can be later interpreted as a binary value based on a user defined cut-off threshold.

3.3 Implementation

We use the implementation of matrix factorization for inferring temporal latent spaces of a sequence of graph snapshots presented in [44]. For the classification part, we use the efficient LIBLINEAR [7] package and set the type of solver to L2-regularized logistic regression (primal).

4 Experiments

To assess the efficacy of our proposed technique, we have conduct experiments to address the following research question: *Does combining latent and meta path-based topological features improve relationship prediction accuracy in DHINs?*

4.1 Experiment Setup

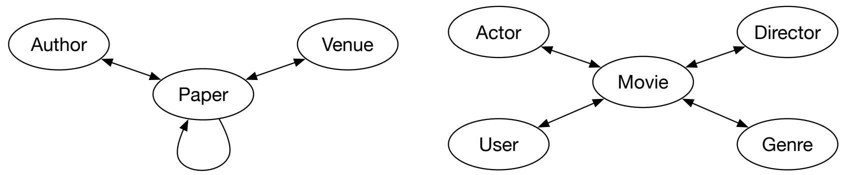
Dataset. We conduct our experiments on two real-world network datasets that have different characteristics and evolution behaviour.

Publications dataset: The AMiner citation dataset [34] version 8 (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 1936 to 2016. Each paper is associated with abstract, authors, year, venue, and title. We confined our experiments on papers published since 1996, which includes 2,935,679 papers. Similar to [30], we also generate another dataset by considering only authors with at least 5 papers.

Movies dataset: The RecSys HetRec movie dataset [2] is an extension of MovieLens10M dataset, published by GroupLeans research group that links the movies of MovieLens dataset with their corresponding web pages at IMDB and Rotten Tomatoes 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, and 855,598 ratings (avg. 404.92 ratings per user, and avg. 84.64 ratings per movie).

Experiment Settings. We describe meta paths and target relationships, comparable methods, and different parameter settings.

Meta paths and target relationships. Figure 3 depicts network schemas for the two datasets. Note that we consider a simplified version and ignore nodes



(a) Publications Network (b) Movies Network

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

Table 1. Meta paths for publications dataset with V={Author, Paper, Venue} and movies dataset with V={User, Movie, Actor, Director, Genre}.

Network	Meta path	Meaning
Publications	*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
Movies	*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

such as topic for papers or tag for movies. Table 1 presents a number of meta paths that we consider in our experiments, where target meta path relations are *co-authorship* and *watching*. We also limit our meta path-based measures to only path count for calculations efficiency, as the results in [30] suggest that except for the case of hybrid features, using the path count measure is not considerably different than the normalized path count or the random walk.

For the publications network we consider meta paths *A-P-V-P-A*, *A-P-A-P-A*, and *A-P-P-P-A*, as the study in [30] shows that shared co-authors, shared venues, and co-cited papers for two authors significantly contribute to their future collaborations. There are two major differences between target relation types in our datasets. First, unlike a new co-authorship relation that happens at a particular time, users can watch/rate a movie once it is released. In other words each paper is published once and a new co-authorship is made at that time whereas users create new watching relations to an existing movie. Second, the target relation for the publications dataset, i.e., *A-P-A*, has the same node type at both ends, while the target meta path for the movie dataset, i.e., *U-M*, considers two different node types. Note that $G_i^{\mathcal{P}}$s in Equation 1 are square adjacency matrices. For the case of having target relations with two types of nodes at ends, we consider 0 value for the relationships of the same type in case no such relation actually exists in the network.

Comparable methods. We compare the following 5 methods: (1) The original *PathPredict* that considers only 3 intervals [30], (2) homogenized link prediction (Section 3.1), denotes by *HLP*, (3) logistic regression on HLP latent features, denoted by *LRHLP*, (4) PathPredict applied on different time intervals, denotes by *PathPredict2*, and (5) our combined meta path and latent features, denoted by *MetaDynaMix*. Sun et al. [30] showed that *PathPredict* outperforms traditional link prediction approaches that use topological features defined in homogeneous networks such as common neighbors or Katzβ, and thus we do not compare our technique with them.

Parameters. We set number of snapshots t=3, 5, and 7 to evaluate the effect of dynamic analysis of different time intervals. Note that t=3 refers to the default case for many link prediction algorithms that learns based on one interval and test based on another interval. More specifically T1 is used for feature calculation and T2 for labels (training) and T2 for feature calculation and T3 for labels (testing). We also set the number of latent features k to 10, 15, and 20.

Evaluation Metrics. To asses the link prediction accuracy, we use Area Under Curves (both Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves), termed as AUCROC and AUCPR [5]. We also present the prediction accuracy for the 5-fold cross validation in training phase as well as the testing phase. Furthermore we perform the non-parametric McNemar's test [21] to assess the statistical significance of the difference between the classifiers accuracy of different techniques.

4.2 Results and Findings

179,607 authors had no co-author in 1996-2016. 78,635 authors had no co-author (about 4%). ——— 100,972 (those who published in 1930-1996)?

1,752,443 (total) - 100,972 = 1,651,471 (those who published in 1996-2016)?

1,544,408 authors had no co-author in 1930-1996 78,635 authors had no co-author (about 4%). ——— 1,465,773 (those who published in 1996-2016)?

1,752,443 (total) -1,465,773 = 300,000 (those who published in between)?

78,635 authors had no co-author (about 4%)

Adding more features to our Logistic Regression model will increase the training accuracy because model has to consider more data to fit the logistic regression. But testing accuracy increases if feature is found to be significant

The null hypothesis of the McNemar's states that the same population proportion of links will be correctly classified by the two methods. However the test result gives a p-value < 0.0001 and hence we reject the null hypothesis of equal classifier performance.

$\boxed{\text{Amin}}$ ▶*One reason that A–P–V–P–A is better with intervals is that one may publish in ECIR but there are so many publishing there....*◀

5 Discussion

Applications. Our proposed technique can also be used in other applications. For example link recommendation and predicting missing edges in graphs.

Vertex Recommendation similar to [26]

Combining topological and latent features. Our hypothesis is that combining latent with topological features can increase 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.

Some latent features may be already covered by topological features .... this may affect the accuracy to some extent and that can be done by feature ebgineewring such as bakward...

As shown in [22] and [44], latent features are more predictive of linking behaviour compared to unsupervised scoring techniques such as Katz, Prefferentail Attachemnet, and Adamic.

Experiments in [22] shows combining the latent structure and side-information increases the prediction accuracy.

In this work we modelled the predicted graph $\hat{G}_t(i,j)$ 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 [22], 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 $\hat{G}_t(i,j) = \Phi(z_i^T z_j + f_D(z_{i,j}; w)$

Link privacy concern. Connection to link privacy research such as [23]

While link prediction techniques has a number of useful applications, it may 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. [8] presented a link reconstruction attack, in which the attacker uses 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. 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.

There is an increasing concern about privacy issues since more and more personal information could be obtained by others online. Many algorithms have been developed for protecting the privacy of users, such as identity, relationship and attributes, from different situations in which different public information was exposed to adversaries [17-20]. In this paper, the focus is on preserving link privacy in social networks.

In retrospect, Zheleva et al. [43] proposed the concept of link re-identification attack, which refers to inferring sensitive relationships from anonymized network data. If the sensitive links can be identified by the released data, then this means privacy breach. Link perturbation is a common technique to preserve sensitive links. Zheleva et al. [43] assumed that the adversary has an accurate probabilistic model for link prediction, and they proposed several heuristic approaches to anonymizing network data. Ying et al. [40] investigated the relationship between the level of link randomization and the possibility to infer the presence of a link in a network. Further, Ying et al. [41] investigated the effect of link randomization on protecting privacy of sensitive links, and they found that similarity indices can be utilized by adversaries to significantly improve the accuracy in predicting sensitive links.

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 [23]. It should be noted that both subnetwork-wise perturbation and neighborhood randomization perturb every link in the network based on a certain probability.

To avoid revealing the sensitive information about users, social relationships, link privacy preserving systems provide a delicately perturbed social graph to these applications by adding extra noise to the local structure of a social network. e.g. [12, 24, 40, 43]. The challenge of preserving link privacy lies in causing no significant losses on the utility of applications that leverage the social trust relationships.

6 Related Work

[44] [32] [31] [13] [38] [33] [30] [39] [18]

Thus techniques for homogeneous networks [18, 35, 19, 16, 1] cannot be directly applied on heterogeneous ones. A few works such as [32, 30] investigated the problem of link/relationship prediction in HINs, however, they do not consider the dynamism of networks and overlook the potential benefits of analyzing a heterogeneous graph as a sequence of network snapshots.

Sun et al. [30] showed that *PathPredict* outperforms traditional link prediction approaches that use topological features defined in homogeneous networks such as common neighbors [25], preferential attachment [25], Jaccard's coefficient [18], and Katzβ [14]

Different from link prediction problem, Sun et al. [31] proposed the problem of predicting relationship building time in HINs.

Matrix factorization technique [22] has been used for link prediction... the effectiveness of the structural link prediction problem, inspired by their success in collaborative filtering... as learning latent features from the data, and why it can be expected to be more predictive than popular unsupervised scores.

Such methods are homogeneous and non-temporal. The number of common neighbors [25], preferential attachment [25], Jaccard's coefficient [18], and Katzβ [14], are amongst frequently used topological features defined in homogeneous networks.

Sun et al. proposed PathSelClus [33] 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 [44].

While many graph embedding methods exist for static networks, few considered dynamic networks. Zhu et al. [44] attempt dynamic link prediction by adding a temporal-smoothing regularization term to a non-negative matrix factorization objective. They use a block-coordinate gradient descent algorithm to perform non-negative factorization. Their formulation is almost identical to the algorithm of Chi et al. [4], 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.

7 Conclusions and Future Work

TBA.

References

1. Al Hasan, M., Chaoji, V., Salem, S., Zaki, M.: Link prediction using supervised learning. In: SDM06: workshop on link analysis, counter-terrorism and security (2006)
2. 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), `http://www.grouplens.org`
3. Chen, H., Li, X., Huang, Z.: Link prediction approach to collaborative filtering. In: Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on. pp. 141–142. IEEE (2005)
4. 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)
5. 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)
6. Denny, J.C.: Mining electronic health records in the genomics era. PLoS computational biology **8**(12), e1002823 (2012)
7. 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), `https://github.com/cjlin1/liblinear`
8. Fire, M., Katz, G., Rokach, L., Elovici, Y.: Links reconstruction attack. In: Security and Privacy in Social Networks, pp. 181–196. Springer (2013)

9. Gallagher, B., Tong, H., Eliassi-Rad, T., Faloutsos, C.: Using ghost edges for classification in sparsely labeled networks. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 256–264. ACM (2008)
10. Guimerà, R., Sales-Pardo, M.: Missing and spurious interactions and the reconstruction of complex networks. Proceedings of the National Academy of Sciences **106**(52), 22073–22078 (2009)
11. Guy, I.: Social recommender systems. In: Recommender Systems Handbook, pp. 511–543. Springer (2015)
12. Hay, M., Miklau, G., Jensen, D., Towsley, D., Weis, P.: Resisting structural reidentification in anonymized social networks. Proceedings of the VLDB Endowment **1**(1), 102–114 (2008)
13. 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)
14. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
15. Lei, C., Ruan, J.: A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity. Bioinformatics **29**(3), 355–364 (2012)
16. Leroy, V., Cambazoglu, B.B., Bonchi, F.: Cold start link prediction. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 393–402. ACM (2010)
17. Li, X., Chen, H.: Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. Decision Support Systems **54**(2), 880–890 (2013)
18. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American society for information science and technology **58**(7), 1019–1031 (2007)
19. Lichtenwalter, R.N., Lussier, J.T., Chawla, N.V.: New perspectives and methods in link prediction. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 243–252. ACM (2010)
20. Lü, L., Medo, M., Yeung, C.H., Zhang, Y.C., Zhang, Z.K., Zhou, T.: Recommender systems. Physics Reports **519**(1), 1–49 (2012)
21. McNemar, Q.: Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika **12**(2), 153–157 (1947)
22. 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)
23. Milani Fard, A., Wang, K.: Neighborhood randomization for link privacy in social network analysis. World Wide Web **18**(1), 9–32 (2015)
24. Mittal, P., Papamanthou, C., Song, D.X.: Preserving link privacy in social network based systems. In: 20th Annual Network and Distributed System Security Symposium, NDSS (2013)
25. Newman, M.E.: Clustering and preferential attachment in growing networks. Physical review E **64**(2), 025102 (2001)
26. 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)

27. 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)
28. 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)
29. Song, H.H., Cho, T.W., Dave, V., Zhang, Y., Qiu, L.: Scalable proximity estimation and link prediction in online social networks. In: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement. pp. 322–335. ACM (2009)
30. 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)
31. 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)
32. 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)
33. 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)
34. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: Extraction and mining of academic social networks. In: KDD'08. pp. 990–998 (2008), `https://aminer.org/citation`
35. Wang, C., Satuluri, V., Parthasarathy, S.: Local probabilistic models for link prediction. In: icdm. pp. 322–331. IEEE (2007)
36. 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)
37. Wang, C., Song, Y., Li, H., Zhang, M., Han, J.: Text classification with heterogeneous information network kernels. In: AAAI. pp. 2130–2136 (2016)
38. 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)
39. 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)
40. Ying, X., Wu, X.: Randomizing social networks: a spectrum preserving approach. In: proceedings of the 2008 SIAM International Conference on Data Mining. pp. 739–750. SIAM (2008)
41. Ying, X., Wu, X.: On link privacy in randomizing social networks. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 28–39. Springer (2009)
42. 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)
43. Zheleva, E., Getoor, L.: Preserving the privacy of sensitive relationships in graph data. In: Privacy, security, and trust in KDD, pp. 153–171. Springer (2008)

44. 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 (2016), https://github.com/linhongseba/Temporal-Network-Embedding