Relationship Prediction in Dynamic Heterogeneous Information Networks

Amin Milani Fard[1], Ebrahim Bagheri[2], and Ke Wang[3]

[1] New York Institute of Technology, Vancouver, Canada
amilanif@nyit.edu
[2] Ryerson University, Toronto, Canada
bagheri@ee.ryerson.ca
[3] Simon Fraser University, Burnaby, Canada
wangk@cs.sfu.ca

Abstract. Most real-world information networks, such as social networks, are heterogeneous and as such, relationships in these networks can be of different types and hence carry differing semantics. 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 features to predict a target relationship between two nodes in a dynamic heterogeneous information network. Our technique, called Meta-DynaMix, effectively combines meta path-based topology features and inferred latent features that incorporate temporal network changes in order to capture network (1) heterogeneity and (2) temporal evolution, when making link predictions. Our experiment results on two real-world datasets show statistically significant improvement over AUCROC and prediction accuracy compared to the state of the art techniques.

1 Introduction

The goal of link prediction [18] is to estimate the likelihood of a future relationship between two nodes based on the observed network graph. Predicting such relationships in a network can be applied in different contexts such as recommendation systems [4, 29, 20, 17, 13], network reconstruction [12], node classification [11], 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 [35] 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 is different in meaning when they

are co-authors compared to the case when one cites another's paper. Thus techniques for homogeneous networks [18, 34, 19, 16, 1] cannot be directly applied on heterogeneous ones. A few works such as [30, 31] investigated the problem of link 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. Previous work on temporal link prediction scarcely studied HINs and to the best of our knowledge, the problem of predicting relationships in dynamic heterogeneous information networks (DHINs) has not been studied before. In this work we study the problem of relationship prediction in a DHIN, which can be stated as: *Given a DHIN graph G at t consecutive time intervals, the objective is to predict the existence of a particular relationship between two given nodes at time $t+1$.* In the context of this problem, the main contributions of our work can be enumerated as follows:

- We propose the problem of relationship prediction in a DHIN, and draw contrast between this problem and existing link prediction techniques that have been proposed for dynamic and/or 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 performance of our work on two real-world datasets, and the results show statistically significant improvement over AUCROC and prediction accuracy compared to the state of the art techniques.

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 $\phi : V \rightarrow \mathcal{A}$ and a link type mapping function $\psi : E \rightarrow \mathcal{R}$, where V, E, $\mathcal{A}$, and $\mathcal{R}$ denote sets of nodes, links, node types, and relation types, respectively. 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 $e = (u, v, t)$ connects two vertices u and v with a timestamp t.*□

The DBLP bibliographic network is an example of a DHIN, containing different types of nodes such as papers, authors, topics, and publication venues, with publication links associated with a 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 considered to be indirectly connected through a set of intermediary links. In this work, we use the terms *relationship prediction* and *link prediction* interchangeably referring to

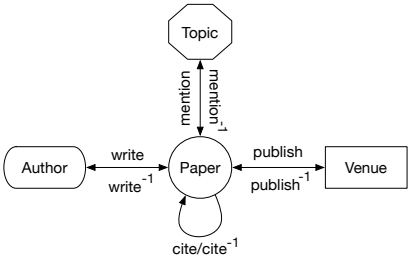


Fig. 1. Network schema for DBLP network.

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 capture 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 that summarizes a HIN and is formally defined as follows:

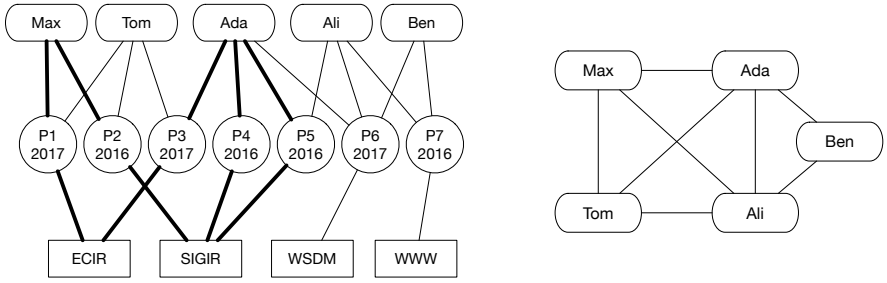
Definition 2 (Network schema). *For a heterogeneous network $G = (V, E)$, the network schema $S_G = (\mathcal{A}, \mathcal{R})$ is a directed meta graph where $\mathcal{A}$ is the set of node types in V and $\mathcal{R}$ is the set of relation types in E.* □

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 $A_i \in \mathcal{A}$ and $R_i \in \mathcal{R}$.* □

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 $\mathcal{P} = A_1 - A_2 ... - A_{n+1}$ if p follows $\mathcal{P}$ 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(a) 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.



(a) An example of A–P–V–P–A meta paths between two authors Max and Ada.

(b) The augmented reduced graph based on $\mathcal{P}(A, A)$=A–P–V–P–A

Fig. 2. An example of a publications network. Link formation time is shown below the paper ID.

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* [36]. Without loss of generality, in this work, we use Path Count (PC) as the default similarity measure. For example, given the meta path A–P–V–P–A and the HIN in Figure 2(a), $PC(Max, Ada)$=3 and $PC(Tom, Ada)$=4. 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 decompose G into 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 an input 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(a) and target meta path $\mathcal{P}(A, A)$=A–P–V–P–A is a graph shown in Figure 2(b)

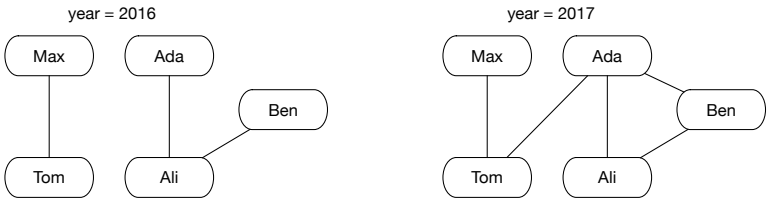


Fig. 3. Augmented reduced graphs for the network in Figure 2(a) with respect to the target meta path *A–P–A* (co-authorship) in 2016 and 2017.

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 into 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. Figure 3 shows examples of such graphs at different time intervals. 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 $\hat{G}_{i+1}^{\mathcal{P}}$ is dependent on $\hat{G}_i^{\mathcal{P}}$.

Recent research in link prediction has focused on network latent space inference [41, 37, 25, 7, 22] with the assumption that the probability of a link between two nodes depends on their positions in the latent space. Each dimension of the latent space characterizes an attribute, and the more two nodes share such attributes, the more likely they are to connect (also known as homophily). Amongst such graph embedding methods, a few [7, 41] considered dynamic networks. Inspired by Zhu et al. [41], 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, λ is a regularization parameter, and $1 - Z_i(x)Z_{i-1}(x)^T$ penalizes sudden changes for x in the latent space. This optimization problem can be solved using gradient descent. The intuition behind the above formulation is two fold: (1) nodes with similar latent space representation are more likely to connect with each other, and (2)

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 $Pr((a, b) \in E^{\mathcal{P}}_{t+1}) \leftarrow \sum_{i=1}^{k} Z_t(a, i)Z_t(b, i)$

nodes typically evolve slowly over time and abrupt changes in their connection network are less likely to happen [39]. The matrix $G^{\mathcal{P}}_{t+1}$ can be estimated by $\Phi(f(Z_1, ...Z_t))$, where Φ and f are link and temporal functions, or simply by $Z_t Z_t^T$. Note that Z_i depends on Z_{i-1} as used in the temporal regularization term in Equation (1).

Algorithm 1 presents a concrete implementation of Equation 1 for relation prediction. It 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 $G^{\mathcal{P}}_i$ 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 $G^{\mathcal{P}}_i$ represents the co-authorship graph at time i. Finally by optimizing Equation (1), it infers latent spaces $Z_1, ..., Z_t$ (line 8) and estimates $G^{\mathcal{P}}_{t+1}$ using $Z_t Z_t^T$ (line 9).

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 of a link between nodes depends only on their latent features. For instance, as depicted in Figure 3, *Tom* and *Ada* became co-authors in 2017 that can be due to publishing at the same venue in 2016, i.e., having two paths between them that passes through *SIGIR*, as shown in Figure 2. Similarly *Ben* and *Ada* who published with a same author, *Ali* in 2016, became co-authors in 2017.

We would like to further hypothesize that combining latent and 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:

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

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 [41] 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 Path Count or PathSim, can be very costly. For example computing path counts for the *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 interpolation of topological and latent features. 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 prefer 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 two hops 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 equivalent 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 cut-off threshold.

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 $j=1..k)\rangle)$

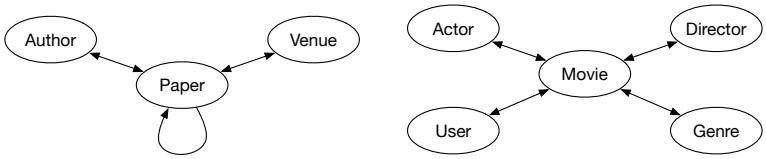
We describe steps for building and applying our predictive model, called *MetaDynaMix*, in Algorithm 2. The algorithm takes as input a DHIN graph G, the number of graph snapshots t, a network schema S, a target relation meta path $\mathcal{P}(A,B)$, the maximum length of a meta path l, the latent space 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 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). Next it creates 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 the algorithm trains the predictive model (line 8), generates features for the given pair (a,b), and tests it using the trained model (line 9).

4 Experiments

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 [33] 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 an abstract, authors, year, venue, and title. We confined our experiments to papers published since 1996, which includes 2,935,679 papers. Similar to [30], we considered only authors with at least 5 papers.



(a) Publications Network (b) Movies Network

Fig. 4. 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

Movies dataset: The RecSys HetRec movie dataset [3] is an extension of MovieLens10M published by the GroupLens research group that links the movies of MovieLens dataset with their corresponding web pages on IMDB and Rotten Tomatoes. 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. Here, we describe meta paths and target relationships, baseline methods, and different parameter settings that have been used in our experiments.

Meta Paths and Target Relationships. Figure 4 depicts network schemas for the two datasets. Note that we consider a simplified version and ignore nodes such as topic for papers or tag for movies. Table 1 presents a number of meta paths that we employed in our experiments where target meta path relations are *co-authorship* and *watching*. Note that in the publications network, each paper is published only once and authorship relationships are formed at the time of publication whereas in the movies network, users can watch/rate a movie at any given point in time and hence user-movie relations are not as rigid as the authorship relations in the publication dataset.

Baseline Methods. Sun et al. [30] proposed a supervised learning framework for link prediction in HINs, called PathPredict, that learns coefficients associated with meta path-based features by maximizing the likelihood of new relationship formation. Their model is learned based on one past interval and does not consider temporal changes in different intervals. Since to our knowledge there is no

baseline for relationship prediction in DHINs, we perform comparative analysis of our work, denoted as MetaDynaMix, with four techniques: (1) The original PathPredict [30] that considers only 3 intervals, (2) PathPredict applied on different time intervals, denoted as PathPredict+, (3) homogenized link prediction (Section 3.1) by applying [41], denoted as HLP, and (4) logistic regression on HLP latent features, denoted as LRHLP. Note that PathPredict [30] was shown to outperform traditional link prediction approaches that use topological features defined in homogeneous networks such as common neighbors or Katzβ, and thus we do not include these techniques in our experiments.

Parameters. We set the 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 learn based on one interval and test based on another. More specifically in the training phase, features are extracted based on T1 and labels are determined based on T2, and for the testing phase, features are calculated based on T2 and labels are derived from T3. In our experiments we did not observe a considerable change in prediction performance by setting the number of latent features k to 5, 10, and 20, and thus all presented results are based on setting k to 20.

Implementation. We use the implementation of matrix factorization for inferring temporal latent spaces of a sequence of graph snapshots presented in [41]. We use all the default settings such as the number of latent features k to be 20, and the optimization algorithm to be the local block-coordinate gradient descent. For the classification part, we use the efficient LIBLINEAR [8] package and set the type of solver to L2-regularized logistic regression (primal).

Evaluation Metrics. To assess link prediction performance, we use Area Under Curves (AUC) for Receiver Operating Characteristic (ROC) [5] and accuracy (ACC). We also perform the McNemar's test [21] to assess the statistical significance of the difference between classification techniques.

4.2 Results and Findings

Link Prediction Accuracy. We now compare the prediction accuracy of different methods. The results shown in Figure 5 are based on setting the number of time intervals t to 7 for dynamic methods and 3 intervals for PathPredict. Table 2 shows more details considering different intervals. These results show the statistically significant improvement provided by the proposed MetaDynaMix prediction method compared to the baselines. The authors in [22, 41] showed that latent features are more predictive compared to unsupervised scoring techniques such as Katz, or Adamic. In our experiments we observed that combining latent features with meta path-based features (MetaDynaMix) can increase prediction accuracy. However, if latent features learn similar structure as topological features do, then mixing them may not be beneficial. In such cases feature engineering techniques could be applied.

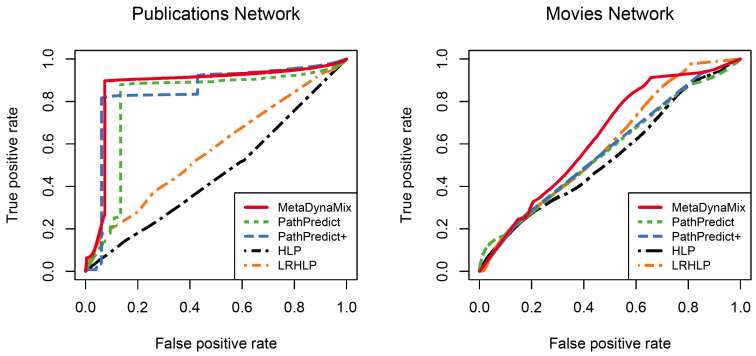


Fig. 5. The ROC curves for different methods and datasets.

Table 2. Relationship prediction accuracy comparison. Bold values are determined to be statistically significant compared to the baselines based on McNemar's test.

		Publications Network			Movies Network		
Method	**Metric**	t=3	t=5	t=7	t=3	t=5	t=7
PathPredict	ROC	0.78	–	–	0.56	–	–
	ACC	0.55	–	–	0.54	–	–
PathPredict+	ROC	0.78	0.80	0.83	0.56	0.57	0.57
	ACC	0.55	0.58	0.60	0.54	0.54	0.55
HLP	ROC	0.42	0.43	0.46	0.51	0.53	0.54
	ACC	0.50	0.50	0.50	0.51	0.52	0.53
LRHLP	ROC	0.49	0.50	0.52	0.52	0.56	0.59
	ACC	0.47	0.50	0.51	0.52	0.56	0.58
MetaDynaMix	ROC	0.85	0.87	**0.87**	0.57	0.59	**0.63**
	ACC	0.78	0.80	**0.82**	0.56	0.60	**0.62**

We also observe that PathPredict+ performs better than LRHLP in predicting links for the publications network but LRHLP offers more accurate predictions on the movies network. This implies that unlike the publications network, our meta path-based features for the movies network are not as predictive as latent features. However, in both cases combining the two set of features gives better performance than either model individually.

Significance of Improvement. McNemar's test, also called within-subjects χ^2 test, is used to compare statistically significant difference between the accuracy of two predictive models based on the contingency table of their predictions. The null hypothesis assumes that the performances of the two models are equal. We compare MetaDynaMix with the other four baselines and the test results show a p-value < 0.0001 for all cases and hence we reject the null hypothesis.

The Effect of Time Intervals. We set the number of time intervals t to 3, 5, and 7 and assess its impact on prediction performance. As presented in Table 2, accuracy increases with the number of snapshots. The intuition is that shorter time intervals result in less changes in the graph and thus leads to more reliable predictions. For example considering a meta path A–P–V–P–A, with smaller number of intervals, i.e., longer time intervals, we have more distinct authors who have published in a venue in different years and thus more similar path

count values. However, by considering more intervals fewer authors will have such relations and more diverse path counts can contribute to a more accurate prediction for the next time interval.

5 Related Work

The problem of link prediction in static and homogeneous networks has been extensively studied in the past [18, 34, 19, 16, 1, 2], for which the probability of forming a link between two nodes is generally considered as a function of their topological similarity. However, such techniques cannot be directly applied to heterogeneous networks. A few works such as [30, 31] investigated the problem of link prediction in HINs. Sun et al. [30] showed that *PathPredict* outperforms traditional link prediction approaches that use topological features defined on homogeneous networks such as common neighbors, preferential attachment, and Katzβ. Different from the original link prediction problem, Sun et al. [31] studied the problem of predicting the time of relationship building in HINs. These works, however, do not consider the dynamism of networks and overlook the potential benefits of analyzing a HIN as a sequence of network snapshots.

Research works on static latent space inference of networks [26, 22, 38, 25, 37] have assumed that the latent positions of nodes are fixed, and only few graph embedding methods [10, 7, 41] have considered dynamic networks. Dunlavy et al. [7] developed a tensor-based latent space modeling technique to predict temporal links. Zhu et al. [41] added a temporal-smoothing regularization term to a non-negative matrix factorization objective to penalize abrupt large changes in the latent positions.These works do not consider heterogeneity of network structure.

6 Conclusions and Future Work

We have studied the problem of relationship prediction in DHINs and proposed a supervised learning framework based on a combined set of latent and topological meta path-based features. Our results show that the proposed technique significantly improves prediction accuracy compared to the baseline methods. As a part of future work and given the major computational bottleneck of methods that rely on meta-paths, such as our approach, is calculating meta path-based measures, we would like to investigate approximation techniques to make the prediction process scalable. Furthermore, we are interested in enhancing the matrix factorization technique based on a loss function that does not require the full topological features matrix. Another interesting direction to investigate is the effectiveness of our proposed approach in other application domains such as predicting user interests in a social network that is both temporally dynamic and heterogeneous by nature. Link prediction techniques may also increase the risk of link disclosure, such as through link reconstruction and re-identification attacks [40, 9], and thus increase privacy concern. It is interesting to study the effect of our technique in performance of link privacy preserving methods, such as [14, 40, 24, 23], and propose suggestions for improvement.

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. Al Hasan, M., Zaki, M.J.: A survey of link prediction in social networks. In: Social network data analytics, pp. 243–275. Springer (2011)
3. 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`
4. 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)
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. Dunlavy, D.M., Kolda, T.G., Acar, E.: Temporal link prediction using matrix and tensor factorizations. ACM Transactions on Knowledge Discovery from Data (TKDD) **5**(2), 10 (2011)
8. 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`
9. Fire, M., Katz, G., Rokach, L., Elovici, Y.: Links reconstruction attack. In: Security and Privacy in Social Networks, pp. 181–196. Springer (2013)
10. Fu, W., Song, L., Xing, E.P.: Dynamic mixed membership blockmodel for evolving networks. In: Proceedings of the 26th annual international conference on machine learning. pp. 329–336. ACM (2009)
11. 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)
12. 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)
13. Guy, I.: Social recommender systems. In: Recommender Systems Handbook, pp. 511–543. Springer (2015)
14. Hay, M., Miklau, G., Jensen, D., Towsley, D., Weis, P.: Resisting structural re-identification in anonymized social networks. Proceedings of the VLDB Endowment **1**(1), 102–114 (2008)
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. Milani Fard, A., Wang, K., Yu, P.S.: Limiting link disclosure in social network analysis through subgraph-wise perturbation. In: Proceedings of the International Conference on Extending Database Technology (EDBT). pp. 109–119. ACM (2012)
25. Qi, G.J., Aggarwal, C.C., Huang, T.: Link prediction across networks by biased cross-network sampling. In: Data Engineering (ICDE), 2013 IEEE 29th International Conference on. pp. 793–804. IEEE (2013)
26. Sarkar, P., Moore, A.W.: Dynamic social network analysis using latent space models. ACM SIGKDD Explorations Newsletter **7**(2), 31–40 (2005)
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. 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`
34. Wang, C., Satuluri, V., Parthasarathy, S.: Local probabilistic models for link prediction. In: icdm. pp. 322–331. IEEE (2007)
35. 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)
36. Wang, C., Song, Y., Li, H., Zhang, M., Han, J.: Text classification with heterogeneous information network kernels. In: AAAI. pp. 2130–2136 (2016)
37. Ye, J., Cheng, H., Zhu, Z., Chen, M.: Predicting positive and negative links in signed social networks by transfer learning. In: Proceedings of the 22nd international conference on World Wide Web. pp. 1477–1488. ACM (2013)
38. Yin, J., Ho, Q., Xing, E.P.: A scalable approach to probabilistic latent space inference of large-scale networks. In: Advances in neural information processing systems. pp. 422–430 (2013)
39. 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)
40. 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)
41. 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`