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 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 social network graph [5] is to estimate the likelihood of the relationship between two nodes in future, based on the observed network. 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. Amin ▶*more citations*◀

Traditional link prediction techniques [5] Amin ▶*more citations*◀ consider networks to be homogeneous, i.e., graphs with only one type of edges and nodes. However, most real-world information networks, such as Twitter, Facebook, and DBLP, are heterogeneous and have multiple relation and node types. For example, in a bibliographic network there are nodes of types authors, papers, and venues, and edges of types write, cite and publish. There are limited number of works that focused on link prediction in heterogeneous information networks (HINs) such as [11]. However, those techniques do not consider the dynamics of social networks and ignore sequence of network snapshots. Amin ▶*make sure about these citations*◀ [16] [11] [10] [4] [13] [12] [9] [14] [5]

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?*

1.1 Motivation

Amin ►*Motivate the problem by a real example and show the points made in this para.*◄

The link prediction problem for homogeneous networks has been studied in the past [5] Amin ►*more citations*◄ . However most real social networks are heterogeneous and 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 [11, 9], investigated this problem, however, they do not consider the dynamics of social networks and ignore ignore sequence of network snapshots. On the other hand, it has ben shown that for homogenous network link prediction, incorporating temporal changes helps in a more accurate prediction [16]. Previous work on temporal link prediction scarcely studied heterogeneousness of social networks and to the best of our knowledge, the problem of relationship prediction for dynamic heterogeneous networks was not studied before.

1.2 Contributions

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.

2 Problem Statement

Definition 1 (Dynamic heterogeneous information network). *A dynamic heterogeneous information network (DHIN) is defined as a directed graph $G = (V, E)$, where V and E are set of nodes and edges of different types, and edges have timestamps.* □

with an object type mapping function $\phi : V \to A$ and a link type mapping function $\psi : E \to R$, where each object $v \in V$ belongs to one particular object type $\phi(v) \in A$, and each link $e \in E$ belongs to a particular relation $\phi(e) \in R$

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

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

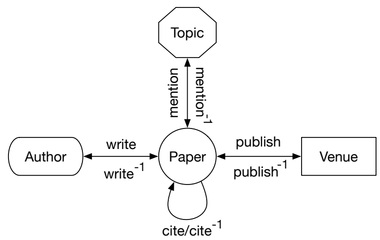


Fig. 1. Network schema for DBLP network.

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. Note that the length of a sequence can be 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.

Definition 2 (Relationship prediction problem). *Given a DHIN graph G at time t, and a target relation type R, we aim to predict the existence of a relation of type R between two given nodes at time $t+1$.* □

In order to better understand different types of nodes and their relation in a network, the concept of *network schema* [11] is used. The network schema is a meta structure graph that summarizes a heterogeneous information network and is formally defined as bellow.

Definition 3 (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.

2.1 Meta path-based topology

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

Definition 4 (Meta path [11]). *A meta path $\mathcal{P}$ is a path in the network schema graph $S_G = (\mathcal{A}, \mathcal{R})$, denoted in the form of $\mathcal{P} = A_1 \xrightarrow{R_1} A_2 ... \xrightarrow{R_n} A_{n+1}$, 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}.* □

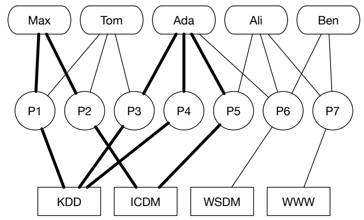


Fig. 2. An example of A–P–V–P–A meta paths between two authors Max and Ada.

In the DBLP network, the co-author relationphip can be described with the meta path $A \xrightarrow{write} P \xrightarrow{write^{-1}} A$ or in short form 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 for a path connecting two nodes, and defines a unique topology representing a special relation. For instance 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).

Table 1. Notations and their description. Amin ▶*update w.r.t the latest version.*◀ Amin ▶*This table will be later removed to to page space limit.*◀

Notation	Description
$\mathcal{P}$, p	Meta path and path
k	The dimension of latent features
G_τ	Graph G at time τ
$\hat{G}_\tau$	Predicted graph G at time τ
Z_τ	low rank k-dimensional latent space representation matrix for G_τ

3 Relationship Prediction Approach

Given a DHIN graph $G = (V, E)$, and number of graph snapshots t, we first decompose G to a sequence of t heterogeneous graphs $G_1, .., G_t$ with respect to its associated timestamps. We then apply our techniques to predict G_{t+1}. As mentioned in Definition 2, in this work we intend to predict *existence of a given type of relationship* (target relation) between two nodes in a heterogeneous network. Therefore we define a new type of graph, called *augmented reduced graph*, that is generated based on a given heterogeneous graph and a target relation meta path.

Definition 5 (Augmented reduced graph). *Given a heterogeneous graph $G = (V, E)$ and a target relation meta path $P(a, b)$ between nodes of type a and b, 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 and b, and edges in E^P indicates relationships of type P in G.* □

An augmented reduced graph for the network in Figure 2 and target relation meta path $P(Author, Author)$=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 another meta path $P(Author, Author)$=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* relationship.

3.1 Homogenize link prediction

Zhu et al. [16] 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}. The authors presented a matrix factorization with block-coordinate gradient descent (BCGD) technique that for each G_τ at time τ infers a low rank k-dimensional latent space representation matrix Z_τ that minimizes the quadratic loss with temporal regularization

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

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 $Z_\tau(u, c)$ indicates the position of u in the c-th dimension at Z. The intuition behind their prediction model is that 1) nodes move smoothly in the latent space over time and it is less likely to have large moves [8, 15], and 2) user interactions are more likely to occur between similar users in a latent space representation. Thus it needs Z_{t+1} to predict the future graph G_{t+1}. They assume Z_{t+1} can be approximated by $Z_1, ..., Z_t$. Finally they use $Z_t Z_t^T$ to predict G_{t+1}. Amin ▶*However, the authors mentioned that the predicted graph at t + 1 can be formulated as $\Phi(f(Z_1, ...Z_t))$, where Phi is the link function and f is the temporal function. Learning or selecting the best link function Phi and temporal function f is beyond the scope of this work. For example, we could apply nonparametric approaches [25] to automatically learn the link function F.*◀

An immediate adaptation of the above BCGD technique is to consider a sequence of augmented reduced graphs G_i^P as input, i.e., graphs with source and destination of a target relationship type with edges between them if the target relation exists in the original graph G_i. This changes equation (1) by replacing G_τ with G_τ^P.

Algorithm 1 gets as an input a DHIN graph G, number of graph snapshots t, a target relation meta path $P(a, b)$, and latent space dimension k. The algorithm first decomposes G into a sequence of graphs $\{G_1, .., G_t\}$ (line 1) by considering the associated timestamps on edges. Next from each graph snapshot G_i, a

Algorithm. 1 Homogenize Link Prediction

Input: A DHIN graph G, number of graph snapshots t, a target relation meta path $P(a,b)$, latent space dimension k
Output: The predicted graph G^P at time $t+1$ based on the given target relation meta path P
1: $\{G_1,..,G_t\} \leftarrow DecomposeGraph(G,t)$
2: **for** each graph G_i **do**
3: Let a and b be the node types of beginning and end of P
4: **for** each node $x \in V_i$ of type a in G_i **do**
5: follow P to reach a node y of type b in G_i
6: add edge (x,y) to augmented reduced graph G_i^P
7: **end for**
8: **end for**
9: Infer temporal latent spaces $Z_1,..,Z_t$ using $BCGD$
10: $G_{t+1}^P \leftarrow Z_t Z_t^T$
11: return G_{t+1}^P

corresponding augmented reduced graph G_i^P is generated (lines 2-8) for which nodes are of type a and b (beginning and end of target relation meta path P). Finally the BCGD technique in [16] is applied (lines 9-10) to infer k-dimensional temporal latent spaces $Z_1,...,Z_t$ and estimate G_{t+1}^P by $Z_t Z_t^T$. Amin ▶*How Z_{t+1} depends on $Z_1,...,Z_t$ from the algorithm? Explain assumptions in [16].*◀

3.2 Meta path-based relationship prediction

The homogenize approach, however, does not consider different semantics of meta paths between the source and destination nodes. In fact, Zhu et al. [16] assume that the probability of a link between nodes depends only on their latent positions. However, we also consider meta path-based features in our prediction model. Our intuition is that leveraging meta path-based features [11] helps to boost prediction accuracy besides using latent space feature space. In other word, we combine latent space features with topological meta path-based features.

We define a set of meta paths [11] on a given network schema and a target relation meta path P, such as co-authorship, to generate an augmented reduced graph (Definition 5) G_i^P from G_i based on P. We then leverage the technique in [16] to predict G_{t+1}^P given $G_1^P,...,G_t^P$, by inferring the temporal latent space representation for nodes at time $t+1$. Amin ▶*this para seems wrong!!!*◀

Algorithm 2 gets as an input a DHIN graph G, number of graph snapshots t, a network schema S, target relation meta path $P(a,b)$ between node types a and b, maximum length of a meta path l, and latent space dimension k. Same as Algorithm 1, it decomposes G into a sequence of graphs (line 1). It then produces the set of all meta paths between a node type a and type b defined in the given target relation $P(a,b)$ (line 2). This is done by traversing the network schema S (for instance through a BFS traversal) and generating meta paths with maximum length of l .

Next from each graph snapshot G_i, a corresponding augmented reduced graph G_i^P is generated (lines 3-9) for which nodes are of type a and b (beginning and end of meta path P) and edges have weight between 0 and 1, based on a similarity measure. For example PathCount[11], PathSim [11], or random walk are measures for the relation of two nodes given link type and meta path,

Algorithm. 2 Meta path-based Link Prediction

Input: A dynamic heterogeneous graph G, number of graph snapshots t, network schema S, a target relation meta path $P^*(a,b)$, maximum length of a meta path l, latent space dimension k

Output: The predicted graph G^{P^*} at time $t+1$ based on the given target relation P^*

```
{G_1, .., G_t} ← DecomposeGraph(G, t)
{P_1, .., P_n} ← GenerateMetaPaths(S, P*(a, b), l)
for each meta path P_j(a, b) do
    for each graph G_i do
        for each node x ∈ V_i of type a in G_i do
            follow P_j to reach a node y of type b in G_i
            w_xy ← MeasureSimilarity(x, y, G_i)
            add edge (x, y) with weight w_xy to augmented reduced graph G_i^{P_j}
        end for
    end for
    Infer temporal latent spaces Z_1, .., Z_t using BCGD
    G_{t+1}^{P_j} ← Z_t Z_t^T
end for
G_t^{P*} ← LastKnownTargetGraph(G, P*(a, b), t)
∀(a, b ∈ N(a)) ∈ G_t^{P*}, add a feature vector to the training set with w_ab in G_{t−1}^{P_j} for each meta
    paths P_j, and label with 1 if (a, b) ∈ E_t^{P*} otherwise 0.
Learn the model and apply it to the feature vector of G_{t+1}^{P_j} with different meta path P_j.
Build G_{t+1}^{P*} based on the cut-off values for the output of prediction model.
return G_{t+1}^{P*}
```

such as co-authorship. Amin ▶*Note that values of G_i^P depends on meta-path P*◀ . Once the sequence of augmented reduced graphs $\{G_1^P, ..., G_t^P\}$ are generated, we apply the matrix factorization with the BCGD technique [16] to find a low rank k-dimensional latent space representation matrix Z_τ for nodes at time τ.

The predictive model. Given the training pairs of nodes and their corresponding meta path-based features (similarity weights w), we build a prediction model to learn the weights associated with these features by apply logistic regression. We define the probability of existence of a link between nodes a and b as $Pr(label = 1|a, b; \boldsymbol{\theta}) = \frac{e^z}{e^z+1}$ where $z = \sum_{i=1}^{n} \theta_i.w_i$ for n meta paths, and θ_i is a normalized weight value for w_i (meta path-based feature). We use logistic regression with L_2 regularization to estimate the optimal θ. $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_i logPr(y_i = 1|a_i, b_i; \boldsymbol{\theta}) - \alpha \sum_{j=1}^{N} \theta_j^2$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_i logPr(y_i = 1|a_i, b_i; \boldsymbol{\theta}) - \alpha \sum_{j=1}^{N} \theta_j^2$$

We derive $\hat{\theta}$ which maximizes the likelihood of all the training pairs, using maximum likelihood estimation.

In the training phase, for each pair of nodes (a, b) in $G_t^{P^*}$, where $b \in N(a)$, we add a feature vector to the training set with corresponding w_{ab} in $G_{t-1}^{P_j}$ for each meta paths P_j, and with label 1 if $(a, b) \in E_t^{P^*}$ otherwise label 0. We then perform logestic regression to learn the model. Finally we apply the model to the feature vector of predicted graphs $G_{t+1}^{P_j}$ with different meta path P_j. Finally it builds $G_{t+1}^{P^*}$ based on the cut-off values for the output of prediction model.

3.3 Combining latent and meta path-based features

Amin ►*ISSUES TO DISCUSS*◄

Zhu et al. [16] 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}. The authors presented a matrix factorization with block-coordinate gradient descent (BCGD)

Although the gradient descent algorithm used in the BCGD technique to infer latent space matrices is fast, we cannot apply it to meta paths efficiently since computing meta path values for all pairs is not scalable. Thus we restrict it to those links that make new connections in future or negative samples and use logistic regression.

- After p-value analysis some latent features might be correlated with meta paths. Removing may increase the accuracy or we can remove those with lower p-value. This needs careful analysis as it might be dependent to number of intervals or the size of latent feature.

- $G_{ij} \approx \sum_{l=1}^{m} \theta_l w_l(i,j) + \sum_{l=1}^{k} \theta_l Z_{il} Z_{jl}$

$$\underset{Z}{\operatorname{argmin}} \sum_{(i,j)\in O} \left\| G_\tau(i,j) - \Phi(z_i^T z_j + f_D(z_{i,j}; w) \right\|_F^2$$

- Augmenting latent with explicit features helps to combine latent features with the results of any other link prediction model. Suppose scores w_{ij} is the meta path feature between nodes i and j. Then, we can treat this as being a dyadic feature $z_i j$ in the above framework, and learn latent features that fit the residual of these scores. In general then, the latent feature approach has a natural mechanism by which any predictive signal can be incorporated, whether it is an explicit feature vector or model predictions. However, a caveat is in order: it is not necessary that combining latent features with another model will improve performance on test data. If the latent features learn similar structure to the other model, then combining the two cannot be expected to yield better results.

As a final remark, we note that the linear combination of latent and explicit features is not the only way to incorporate side information. This issue has been studied in the context of the cold-start problem [29] in collaborative filtering. Recent advances in this literature are based on inferring reasonable values of latent features by falling back to the side information as a prior [2,12]. However, unlike most collaborative filtering applications, in link prediction we are mostly interested in using side information to improve predictions, rather than dealing with cold-start nodes. Therefore, we expect it to be most useful to directly augment the latent feature prediction with one based on side information.

3.4 Implementation

We use the implementation of temporal latent space inference for a sequence of dynamic graph snapshots [2][16]. For the classification part, we use the efficient LIBLINEAR [3] package[3] and set the type of solver to L2-regularized logistic regression (primal) that solves $min_w w^T w/2 + C \sum log(1 + exp(-y_i w^T x_i))$, where w is the generated weight vector as the model for a given set of instance-label pairs (x_i, y_i), $i = 1, \ldots, l$, $x_i \in R^n$, $y_i \in \{-1, +1\}$, $w^T w/2$ is the regularization term, and $C > 0$ is a penalty parameter. In the testing phase, we predict a data point x as positive if $w^T x > 0$, and negative otherwise. We performed 5-fold cross validation for the training phase.

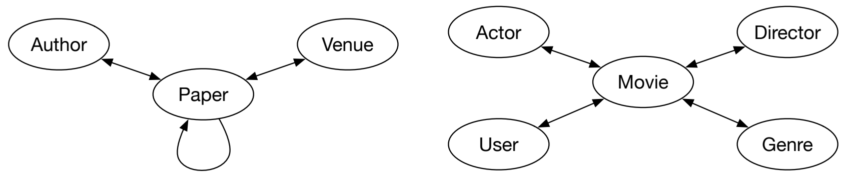
4 Experiments

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 [9] 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 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).

[2] https://github.com/linhongseba/Temporal-Network-Embedding
[3] https://github.com/cjlin1/liblinear
[4] https://aminer.org/citation
[5] http://www.grouplens.org
[6] http://www.imdb.com
[7] http://www.rottentomatoes.com



(a) DBLP (b) IMDB

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

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 [9] 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 2 shows meta paths between authors under length 4 for the publications dataset.

Table 2. 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

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 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 3. 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 [2]. 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 [7]

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 [], 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 [] and [16], latent features are more predictive of linking behaviour compared to unsupervised scoring techniques such as Katz, PrefferentailAttachemnet, and Adamic.

Experiments in [] shows that although we cannot certainly infer that latent structure are more predictive than side-information, combining the two increases the prediction accuracy.

* Link prediction via matrix factorization

Connection to link privacy research such as [6]

5 Related Work

[16] [11] [10] [4] [13] [12] [9] [14] [5]

PathSelClus [22] utilizes 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. [22] Y. Sun, B. Norick, J. Han, X. Yan, P. Yu, X. Yu. Integrating Meta-Path Selection with User-Guided Object Clustering in Heterogeneous Information Networks. In KDD, 2012.

Some earlier works have used temporal smoothness for evolutionary clustering [36] and link prediction in a dynamic network [16].

There exist many embedding methods for static networks, however very few considered dynamic networks. Zhu et al. [16] 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. [36], 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.

[36] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, ?Evolutionary spectral clustering by incorporating temporal smoothness,? in Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2007, pp. 153?162.

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. 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)
3. 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)
4. 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)
5. 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)
6. Milani Fard, A., Wang, K.: Neighborhood randomization for link privacy in social network analysis. World Wide Web **18**(1), 9–32 (2015)
7. 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)
8. Sarkar, P., Moore, A.W.: Dynamic social network analysis using latent space models. ACM SIGKDD Explorations Newsletter **7**(2), 31–40 (2005)
9. Sun, Y., Barber, R., Gupta, M., Aggarwal, C.C., Han, J.: Co-author relationship prediction in heterogeneous bibliographic networks. In: 2011 International Conference on Advances in Social Networks Analysis and Mining. pp. 121–128 (July 2011). https://doi.org/10.1109/ASONAM.2011.112
10. 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). https://doi.org/10.1145/2124295.2124373, `http://doi.acm.org/10.1145/2124295.2124373`
11. 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)
12. 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)
13. 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)
14. 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). https://doi.org/10.1109/ICDM.2012.144
15. 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)

16. 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). https://doi.org/10.1109/TKDE.2016.2591009