Temporal Link Prediction in Dynamic Networks

Hemant Kasat
hemant.kasat@research.iiit.ac.in
IIIT-Hyderabad

Sanket Markan
sanket.markan@students.iiit.ac.in
IIIT-Hyderabad

Manish Gupta
manish.gupta@iiit.ac.in
IIIT-Hyderabad

Vikram Pudi
vikram@iiit.ac.in
IIIT-Hyderabad

ABSTRACT

Link Prediction is an important task for evolutionary analysis of dynamic networks where the goal is to predict links over time based on historical evolution of the network. Given a sequence of previous snapshots we propose a temporal link function, **SiameseLSTM**, to predict the probability of link formation for any pair of nodes in the near future. We assume that nodes lie in a temporal latent space, gradually move as the network evolves over time and co-evolve with their neighbors in the near future. The proposed model uses LSTM to project the temporal latent space embeddings ([7, 41]) of nodes to a **hidden state latent space** optimized for the downstream link prediction task. We use a Siamese adaptation of LSTM for the hidden state embeddings to follow Structural Homophily: two nodes which are close to each other in the latent space interact with one another more frequently than two faraway nodes.

We empirically show that our model outperforms state of the art algorithms for link prediction when evaluated on real world dynamic networks. We also show how varying the number of previous snapshots used to exploit historical information affects the performance of the model to predict future links.

KEYWORDS

Dynamic Networks, Temporal Information , Link Prediction, Siamese Network, LSTM, Latent Space Model

1 INTRODUCTION

Dynamic networks arise in many real life applications such as web social networks, citation networks, communication networks, cyber-physical systems, biological networks etc [3]. Link Prediction is an important task for evolutionary analysis of such dynamic networks, where we infer potential links in future based on series of observed network snapshots in the past [1, 7, 9, 13, 27, 39, 41].

In this work we focus on temporal link prediction problem: Given a sequence of graph snapshots $G_1, ..., G_t$ from time 1 to t, how do we predict links in future time $t + 1$? For this, one needs to construct model for link probabilities between pair of nodes. Recent research in link prediction has focused on latent space modeling of networks. That is, given the observed interactions between nodes the goal is to infer the position of each node in some latent space so that the probability of a link between two nodes depends on their positions in that space. Various approaches, including Bayesian inference [15, 37], Multidimensional scaling (MDS) [29], matrix factorization [11, 13, 24, 26, 36, 40] have been proposed to infer the static latent space representations of the observed network. Most of these approaches have focused on static graphs, where the latent positions of the nodes are fixed.

To extend the static latent space models for dynamic networks, several approaches propose to learn a temporal latent space such that for each snapshot, nodes that are more likely to have a link are closer in the temporal latent space. Some methods have exploited topological structural information, which implies that two nodes close to each other are more likely to form a link in the near future [7, 9, 10, 13, 20, 21, 34, 38, 39, 41]. However, since real-life networks are often very sparse with limited observed links, methods considering structural information alone may have poor performance. Recently, many studies have also exploited temporal information, which reveals the relationships between the current state of a network and its recent history [7, 9, 27, 29, 38, 39]. They assume the principle of temporal smoothness, which states that the current state of a network should not change drastically from its most recent history [8].

A very recent study LIST [39] has exploited structural and temporal information and characterized network evolution using a time function. Another recent study STEP [7] has also exploited both structural and temporal information, but characterized network evolution using a global transition matrix to reflect different types of evolutionary patterns which was difficult using time function used in LIST [39] as the time function should be general enough to reflect the evolutionary patterns. STEP also preserves the deep network structure by considering the higher-order proximity among nodes.

All temporal latent space models formulate link prediction problem in dynamic networks as an indirect regularized optimization problem where the main cost is to infer proximity (or high order proximity) among node pairs along with structural and temporal evolution constraints costs. These methods use a link function (temporal in nature) to predict the probability of link formation between nodes in future. However, as the network dynamics is non-linear in nature having temporal dependencies, we need a parametric temporal link function with the parameter being optimized directly on the downstream link prediction task.

Given temporal latent spaces $Z_1, ..., Z_t$ for graph snapshot sequence $G_1, ..., G_t$, we propose a temporal link function $F(Z_1, ..., Z_t) \mapsto H_1, ..., H_t$ where H_τ for $\tau \in \{1, .., t\}$ represent hidden latent space such that the probability of link formation between two nodes depends on their position in the hidden state space.

We propose a recurrent neural network architecture to model the temporal function F. Recurrent neural networks (RNN), especially the Long Short-Term Memory (LSTM) Network [14] have been particularly successful in sequence representation task [25, 31, 32].

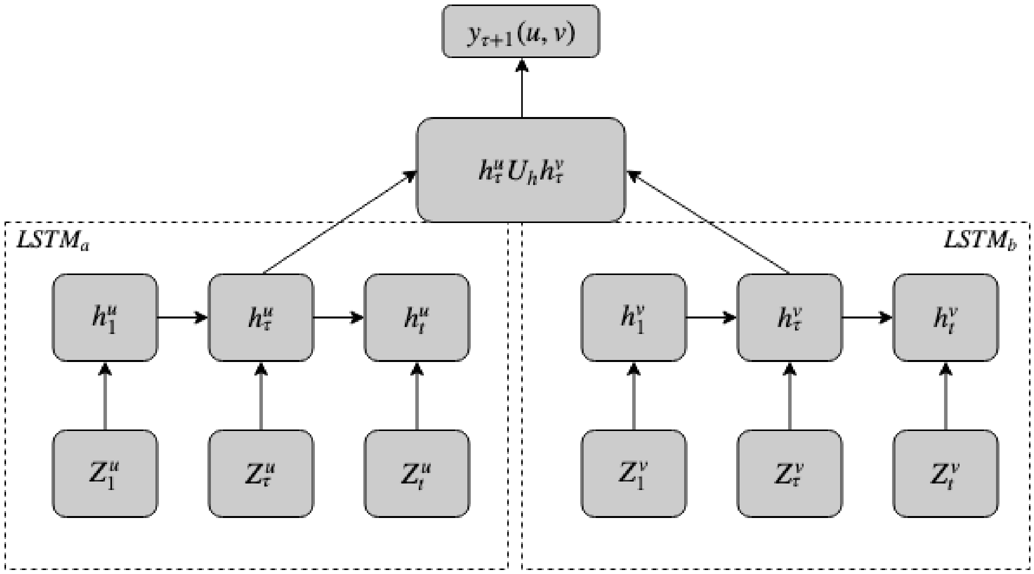

Figure 1: Our model uses LSTM to read in temporal node embedding sequence for pair of nodes and employs hidden states to predict whether they will have link or not at different time steps.

In this work we show that the Siamese adaptation of LSTM, SiameseLSTM, can be trained using observed links over sequence of graph snapshots. Specifically for every node pair u, v in graph G_τ, for each $\tau \in \{1, ..., t\}$, probability $y_\tau(u,v)$ that they will have link in graph $G_{\tau+1}$ is given by :

$$y = F(Z_1^u, .., Z_\tau^u) U_h F(Z_1^v, ..., Z_\tau^v)$$

where Z_τ^u represents node $u's$ latent position at time-point τ.

SiameseLSTM uses temporal latent space embedding of nodes in a principled manner to learn from historical evolution patterns from sequence of graph snapshots to better predict the probability of two nodes forming link in future.

Our major contributions are summarized as follows:

- Given a temporal latent space for sequence of graph snapshots we define the proposed temporal link function SiameseLSTM. Further we define a regularized optimization problem to leverage historical information present in past graph snapshots.
- We briefly define two recent temporal latent space models, BCGD (Block-Coordinate Gradient Descent) [41] and STEP (Structural and Temporal Evolution in Link Prediction) [7] which serve as input to the proposed model. The key idea of both models is to collectively leverage structural and temporal information to better infer a low-rank temporal latent space.
- We conduct extensive experiments on several real-life datasets to validate the performance of the proposed temporal link function when used with two different temporal latent space models BCGD and STEP. We also show the effect of different hyper parameters on the performance of the model.

The rest of the paper is organized as follows. We formulate the temporal link prediction problem in Section 2. Section 3 defines the proposed link function along with its optimization formulation. In section 4 we briefly define the temporal latent space models BCGD and STEP. Experimental results and effect of parameters on performance is given in Section 5. Section 6 describes several related works. We conclude the paper in Section 7.

2 PROBLEM FORMULATION

Let $G(V, E)$ denote a network where V is set of nodes and $E \subseteq V \times V$ is the set of links. We denote individual nodes by u and v and timestamps by τ. t is the total number of time snapshots.

Dynamic networks evolve over time generating sequence of snapshots denoted by $G_1(V, E_1), G_2(V, E_2), ..., G_t(V, E_t)$. V can vary as the network evolves but one can always modify the network to include all observed nodes from all snapshots. Let Z_τ be the low-rank r-dimensional temporal latent space embeddings for set V.

We aim to learn the temporal latent space exploiting structural and temporal information based on two key ideas described below.

- Network structural evolution: Two nodes that are close to each other in the network in terms of network distance, are also close to each other in temporal latent space [6]. Also individual node is more likely to co-evolve with its neighbors in the near future [7].
- Network temporal smoothness: Network evolves smoothly, i.e., current state of the network shouldn't change dramatically from its most recent history [8].

We use BCGD and STEP as our temporal latent space models which are described in Section 4. We define link prediction problem as predicting the emergence of new links and deletion of existing links in graph G_{t+1} [41]. Here the network can by symmetric or asymmetric and weighted or unweighted.

3 TEMPORAL LINK FUNCTION: SIAMESELSTM

Let $Z_\tau \in R^{n \times r}$ denote a temporal latent space, for each $\tau \in \{1, ..., t\}$, where n is total number of nodes and r is the temporal latent space dimension. If the link function is F then we model F using a Long Short Term Memory network. Practically LSTM is superior to basic RNNs for learning long temporal dependencies through its use of memory cell units that can store/access information across lengthy input sequences. Like RNNs, LSTM sequentially updates a hidden-state representation, but these steps also rely on a memory cell containing four components (which are real-valued vectors): a memory state c_τ, an output gate o_τ that determines how the memory state affects other units, as well as an input (and forget) gate i_τ (and f_τ) that controls what gets stored in (and omitted from) memory based on each new input and the current state. For each node u sequence $(Z_1^u, ..., Z_t^u)$ serve as the input to the LSTM.

Below are the updates performed at each timestamp $\tau \in \{1, ..., t\}$ in an LSTM parameterized by weight matrices W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o and bias-vectors b_i, b_f, b_c, b_o.

$$
\begin{aligned}
i_\tau &= sigmoid(W_i Z_\tau^u + U_i h_{\tau-1} + b_i) \\
f_\tau &= sigmoid(W_f Z_\tau^u + U_f h_{\tau-1} + b_f) \\
\hat{c_\tau} &= sigmoid(W_c Z_\tau^u + U_c h_{\tau-1} + b_c) \\
c_\tau &= i_\tau \odot \hat{c_\tau} + f_\tau \odot c_{\tau-1} \\
o_\tau &= sigmoid(W_o Z_\tau^u + U_o h_{\tau-1} + b_o) \\
h_\tau &= o_\tau \odot tanh(c_\tau)
\end{aligned}
\tag{1}
$$

By following updates given by Equation 1, we get hidden state embedding for each node u given by h_τ^u for each $\tau \in \{1, ..., t\}$.

A simple Siamese adaptation of the LSTM can be used to predict the probability $y_{\tau+1}(u, v)$ that a node pair u, v will have new link at time step $\tau + 1$, and is given by:

$$y_{\tau+1}(u, v) = h_\tau(u) U_h h_\tau(v)$$

where $U_h \in R^{d \times d}$ is hidden state space interaction matrix and can incorporate directed and undirected natures of graphs and d represents hidden state space dimension.

The proposed SiameseLSTM is outlined in Figure 1 and has two sub-networks $LSTM_a$ and $LSTM_b$ which each process one node for a given node pair u, v. We choose the Siamese architecture with tied weights such that $LSTM_a = LSTM_b$.

Now we define an optimization formulation to optimally learn LSTM weights and hidden state space interaction matrix U_h. Formally $(Z_1^u, ..., Z_\tau^u)$ and $(Z_1^v, ..., Z_\tau^v)$ represent node embedding sequences for nodes u and v respectively for each $\tau \in \{1, ..., t\}$. $Y_{\tau+1}(u, v)$ is defined as follows.

$$Y_{\tau+1}(u, v) = \begin{cases} 1 & \text{if } u \text{ and } v \text{ have new link from } \tau \text{ to } \tau + 1 \\ 0 & \text{otherwise} \end{cases}$$

It represents the interaction measure between the node pair from τ to $\tau + 1$. Then we define a minimization optimization problem as follows.

$$\min J = \sum_{\tau=1}^{t-1} \sum_{(u,v) \in V \times V} \|Y_{\tau+1}(u, v) - h_\tau^u U_h h_\tau^v\|_F^2 + \beta \|U_h\|_F^2 \tag{2}$$

Optimization formulation given by Equation 2 uses observed links across snapshots to learn evolutionary pattern.

Hidden state embedding h_τ^u for any node u is calculated using node embedding sequence $Z_{\tau-w+1}^u, ..., Z_\tau^u$ transformed using LSTM where w is the window-size and represents how many previous snapshots to consider to calculate hidden state latent space. Optimizing Equation 2 is computationally expensive due to the large number of possible node pairs ($O(V \times V)$) at each graph snapshot.

We define a sampling approach to reduce the number of node pairs at each $\tau \in 1, ..., t - 1$. We find all node pairs which have link in future given current time-stamp τ and call them set Ps. Now we randomly sample same number of node pairs which did not have link in future and call them set Ns [41]. The modified optimization problem now can be written as follows.

$$\min J = \sum_{\tau=1}^{t-1} \sum_{(u,v) \in Ps+Ns} \|Y_{\tau+1}(u, v) - h_\tau^u U_h h_\tau^v\|_F^2 + \beta \|U_h\|_F^2 \tag{3}$$

The sampled node pairs Ps and Ns at each time step $\tau \in \{1, ..., t\}$ are training pairs for the SiameseLSTM architecture. We allow $|Ps| = |Ns|$ to be ≤ 10000 [41].

4 TEMPORAL LATENT SPACE MODELS: BCGD AND STEP

In this section we briefly present two latent space models BCGD [41] and STEP [7] which we use as input for our temporal link function.

4.1 STEP

STEP aims to jointly optimize for higher order proximity matrices factorization at all time steps in addition with structural evolution and temporal smoothness constraints optimization. The overall optimization formulation is given as follows.

$$
\begin{aligned}
\min J = &\sum_{\tau=1}^{t} \|f(A_\tau) - Z_\tau U Z_\tau^T\|_F^2 + \alpha \sum_{\tau=1}^{t} \|Z_\tau - D_\tau^{-1} f(A_\tau) Z_\tau\|_F^2 \\
&+ (1 - \alpha) \sum_{\tau=2}^{t} \|Z_\tau - P Z_{\tau-1}\|_F^2 + \beta \|U\|_F^2 \\
&s.t. \quad Z_\tau^T Z_\tau = I,\ Z_\tau \geq 0, \quad for \quad \tau = 1, 2, ..., t \\
&\quad P1 = 1, P \geq 0
\end{aligned}
\tag{4}
$$

where $f(A_t)$ is a higher order proximity matrix for graph G_t and given by:

$$f(A) = \log \frac{\hat{A} + \hat{A}^2 + ... + \hat{A}^k}{k}$$

where $\hat{A}$ is the row-normalized adjacency matrix A, i.e., each row of $\hat{A}$ sums up to one. k is an integer number representing the order of proximity, which usually controls the trade-off between computational speed and accuracy. $D_\tau \in R^{n \times n}$ is the diagonal matrix for

Dataset	#Nodes	#Edges
CollegeMsg	1899	59835
Hep-Ph	28093	4,596,803
Digg	30398	87,627

Table 1: Statistics of Dynamic networks

$f(A_\tau)$. 1 is the n-dimensional column vector of all ones. $\|.\|_F$ is the the Frobenius norm, $Z_t \in R^{n\times r}$ is low rank embedding for nodes, n is number of nodes and r is the matrix rank (embedding dimension). $U \in R^{rxr}$ is attribute interaction matrix and has different meaning for directed and undirected networks. It models link transitivity between nodes. $P \in R^{nxn}$ is a stochastic transition matrix to approximate the change of low rank representation matrices from time $\tau - 1$ to time τ. $Z_\tau = PZ_{\tau-1} + \epsilon_\tau$.

To solve the optimization problem in Equation 4, STEP proposes an efficient block coordinate gradient descent approach, where each group of variables naturally forms a 'block'. To be specific, the objective function is alternately minimized with respect to one group of variables while fixing the rest.

Low-rank embeddings Z_τ are updated as described in the following. Gradient of $J(Z_\tau)$ (w.r.t Z_τ) when optimization formulation, Equation 4, only includes terms related to Z_τ can be written as follows.

$$\begin{aligned}\nabla J(Z_\tau) = &-2(f(A_\tau)Z_\tau U_T + f(A_\tau)^T Z_\tau U) + 2Z_\tau \hat{U} + 2\alpha W_\tau Z_\tau \\ &+ 2(1-\alpha)(\hat{P}Z_\tau - PZ_{\tau-1} - P^T Z_{\tau+1}) \\ &where \\ &W_\tau = (I - D_\tau^{-1} f(A_\tau))^T (I - D_\tau^{-1} f(A_\tau)) \\ &\hat{U} = UU^T + U^T U \quad and \quad \hat{P} = I + P^T P \end{aligned} \tag{5}$$

As the gradient $\nabla J(Z_\tau)$ is Lipschitz continuous with Lipschitz constant given by:

$$L = 4\|f(A_\tau)\|_F^2 \|U\|_F^2 + 2\|\hat{U}\|_F^2 + 2\|\alpha W_\tau + 2(1-\alpha)\hat{P}\|_F^2$$

STEP employ Nesterov's [5] descent to solve for Z_τ which is scalable for large datasets. STEP gets optimal Z_τ by constructing two sequences Z_τ^k and Y_τ^k and alternately updates them in each iteration round. At k-th iteration we have

$$\begin{aligned} Z_\tau^k &= \rho_+(Y^{k-1} - 1/L\nabla J(Y^{k-1})) \\ \gamma_k &= \frac{1 + \sqrt{4\gamma_{k-1}^2 + 1}}{2} \\ Y^k &= Z_\tau^k + \frac{\gamma_{k-1} - 1}{\gamma_k}(Z_\tau^k - Z_\tau^{k-1}) \end{aligned} \tag{6}$$

where the projection $(\rho_+(A))_{ij} = max(0, A_{ij})$. Also, $\nabla J(Y^{k-1})$ can be calculated using Equation 5 and γ_k is the acceleration coefficient for updating Y^k. Initialization is done as follows: $Y^0 = Z_\tau$ and $\gamma_0 = 1$. We can continuously generate two sequences Z_τ^k and Y_τ^k by updating according to Equation 6. As the sequences converge, the optimal solution for optimization formulation can be determined by the final value of Z_τ^k.

The attribute interaction matrix U is updated as described in the following. Only the following terms in the objective function in Equation 4 have U in them.

$$\min_U J(U) = \sum_{\tau=1}^{t} \|f(A_\tau) - Z_\tau U Z_\tau^T\|_F^2 + \beta\|U\|_F^2$$

By setting the derivative of the above objective function $\nabla J(U) = 0$, we can obtain the optimal solution as follows.

$$U = \frac{\sum_{\tau=1}^{t} Z_\tau^T f(A_\tau) Z_\tau}{t + \beta}$$

The global transition matrix P is updated as follows. The global transition matrix can be solved by minimizing $J(P)$, defined as follows.

$$\min_{P1=1, P\geq 0} J(P) = \sum_{\tau=2}^{t} \|Z_\tau - PZ_{\tau-1}\|_F^2$$

This equation can be further decomposed into several independent sub-problems w.r.t. each row of P as follows.

$$\min_{P(i,:)1=1, P(i,:)\geq 0} J(P(i,:)) = \sum_{\tau=2}^{t} \|Z_\tau(i,:) - P(i,:)Z_{\tau-1}\|_F^2$$

Because this objective function $J(P(i,:))$ is convex and continuously differentiable over a compact convex set, STEP applies simple but efficient Frank-Wolfe [17] method to solve it.

4.2 BCGD

The Temporal Latent space model, Block Coordinate Gradient Descent (BCGD), optimizes for structural and temporal information by formulating a optimization problem given by

$$\begin{aligned} \min J(Z_1, .., Z_t) = &\sum_{\tau=1}^{t} \sum_{(u,v)\in E_\tau} (G_\tau(u,v) - Z_\tau(u)Z_\tau(v)^T)^2 \\ &+ \sum_{\tau=1}^{t} \sum_{(u,v)\notin E_\tau} (Z_\tau(u)Z_\tau(v)^T)^2 \\ &+ \lambda \sum_{\tau=1}^{t} \sum_{u} (1 - Z_\tau(u)Z_{\tau-1}(u)^T) \\ &\text{subject to } \forall u, \tau, Z_\tau \geq 0 \text{ and } Z_\tau(u)Z_\tau(u)^T = 1 \end{aligned} \tag{7}$$

Equation 7 incorporates temporal information by constraining latent position of nodes to move gradually without any drastic jump.

We use Incremental BCGD Algorithm [41] to solve this optimization problem to infer temporal latent space.

5 EXPERIMENTS

In this section, we present the experimental evaluation of proposed temporal link function SiameseLSTM with input from BCGD and STEP temporal latent space models and compare them with recent state of the art methods.

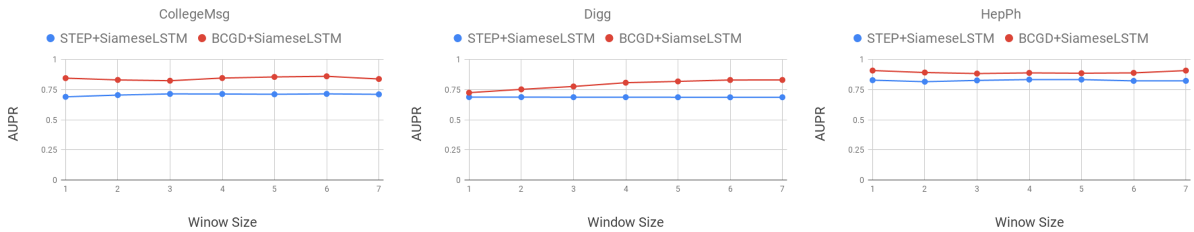

Figure 2: Window size vs AUPR

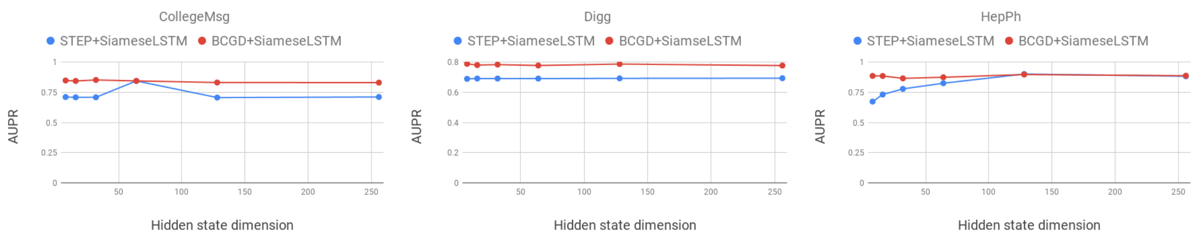

Figure 3: Hidden state dimension vs AUPR

	CollegeMsg	Hep-Ph	Digg
AA	0.5057	0.5607	0.5003
BCGD	0.5432	0.5567	0.6177
STEP	0.7205	0.8211	0.6852
STEP+SiameseLSTM	0.7278	0.8841	0.6879
BCGD+SiameseLSTM	**0.8246**	**0.8870**	**0.7769**

Table 2: New links prediction accuracy: Area under Precision-Recall curve comparison

5.1 Datasets

We briefly describe the datasets in the following.

- Hep-Ph (undirected) : Hep-Ph dataset is a collaboration network from the arXiv's High Energy Physics-Phenomenology section. Nodes are authors and an edge between two authors denotes a common publication. Timestamps denote the date of publication.
- Digg (directed) : The dataset contains the reply networks of the social news website Digg. Each node in the network is a user of the website, and each directed edge denotes that a user replied to another user.
- CollegeMsg (directed) : This dataset is comprised of private messages sent on an online social network at the University of California, Irvine. Users could search the network for others and then initiate conversation based on profile information. Every edge (u,v) has a time-stamp denoting the time when a user u sent a private message to user v.

The statistics of the three datasets are shown in Table 1. The number of edges in each network is the total number of edges over all the snapshots. We create t number of snapshots for each network by dividing the whole time-stamp interval from first edge time-stamp to the last edge time-stamp into equal time interval width. In the experiments if A_τ is adjacency matrix for snapshot τ then element a_{ij} represents the link weight between vertices i and j at time-stamp τ for each $\tau \in \{1, ..., t\}$. In this work we set $t = 8$.

5.2 Evaluation metric and baseline methods

For dynamic networks, it is generally more meaningful to assess an approach in terms of its predictive power for future network behavior. Towards this goal, for each dataset we use $A_1, ..., A_{t-1}$ as training data and A_t as test data. To systematically evaluate the performance of different methods we use precision-recall curve as Receiver Operating Characteristic (ROC) curve can be deceptive and precision-recall curve yields better precision in evaluating the performance of link prediction [35]. We report the area under the precision-recall curve (AUPR) to qualify the performance of an approach.

We compare our method's performance with the following baseline methods.

- Adamic-Adar (AA): AA assumes that two nodes are more likely to be linked together if they share many common neighbors. AA assigns each common neighbor a weight to reflect its contribution. Although AA approach is commonly used for static networks, it can also be applied to time-varying networks by aggregating all edges from different timestamps to one network.
- BCGD: BCGD is scalable temporal latent space model approach for link prediction, which assumes two nodes are more likely to form a link if they are close to each other in the latent low-rank space. The incremental inference algorithm is chosen and all other parameters (e.g., latent dimension $r = 20, \lambda = 0.0001$) are set to values same as the ones in [41]. The model learns temporal latent space $(Z_1, ..., Z_{t-1})$ and

defines a link function for prediction of A_t given by

$$A_t \approx Z_{t-1} Z_{t-1}^T$$

- STEP: STEP uses temporal latent space embeddings learned by optimization formulation to predict future graphs. It implements a simple weighted decaying model to uncover time-varying process among massive set of time series data. The model specifies that A_t depends on its previous series of data $Z_{t-w+1}, ..., Z_{t-1}$ and defines a temporal link function which can then be written as

$$A_t \approx \sum_{\tau=t-w+1}^{t-1} D(\tau) P^{t-\tau+1} Z_\tau U (P^{t-\tau+1} Z_\tau)^T$$

  where w is the window size of historical data, and $D(\tau)$ is decay factor and can be selected as an exponential decay function with parameter $\theta > 0$ such as $D(\tau) = e^{-\theta(t-\tau)}$. We set the model parameters $\alpha = 0.5,\ \beta = 0.1,\ \theta = 0.8,\ w = 3$. We set $k = 3$ while constructing the high order proximity matrix $f(A_\tau)$ same as the ones suggested in [7].

For each dataset, we use $\{A_1, A_2, ..., A_7\}$ as the training set and try to predict the potential links for A_8. Specifically we select all node pairs (u, v) such that $A_8(u, v) \neq 0$ but $A_7(u, v) = 0$ and call them test links. We randomly generate an equal size of non-links i.e. node pairs (u, v) such that $A_7(u, v) = A_8(u, v) = 0$ and call them test non-links. The equal number of test links and test non-links are then served as the test set in the evaluation process. We carry out prediction on five different randomly generated test set for each model and report average AUPR values.

The original BCGD method is only for undirected graphs, we transfer the adjacency matrix of directed graph to undirected one by using $(A + A^T)/2$ when implementing BCGD on directed graphs.

Optimization of SiameseLSTM paramers is done using Adam optimizer. We employ early stopping based on a validation set containing 25% of training node pairs.

5.3 Experimental Results

In this section we present results for the proposed SiameseLSTM model with different temporal latent spaces as inputs, i.e., BCGD (BCGD+SiameseLSTM) and STEP (STEP+SiameseLSTM) and other baseline methods.

For comparison with baselines we empirically set parameters for SiameseLSTM link function and present effect of varying the parameters in the next subsection. We set hidden state space dimension $d = 64$. For BCGD+SiameseLSTM we set latent space dimension $r_{BCGD} = 20$. For STEP+SiameseLSTM we set latent space dimension $r_{STEP} = 128$. Window size $w = 3$ is used for all datasets with regularization parameter $\beta = 0.5$. We randomly generate five different sets for previously observed links to be used as training pairs for SiameseLSTM and report average AUPR for each of the five randomly generated test sets. Table 2 shows comparison of different methods for new link prediction task. There are several interesting observations. First, SiameseLSTM with BCGD temporal latent space outperforms all other methods. Although STEP is better than BCGD but when using with SiameseLSTM, BCGD+SiameseLSTM outperforms STEP+SiameseLSTM. Note that performance of both latent space models, BCGD and STEP, increases when SiameseLSTM is used as the temporal function.

5.4 Parameter Studies

Now we study the impact of varying different parameters of SiameseLSTM like window-size (w), hidden state space dimension (d) and regularization parameter (β).

We plot window size vs AUPR for both BCGD+Siamese- LSTM (Red) and STEP+SiameseLSTM (Blue) for all the datasets in Figure 2 keeping other parameters constant ($\beta = 0.5$, $r_{BCGD} = 20$, $r_{STEP} = 128$, $d = 64$). We vary window size from 1 to 7 (as total number of training snapshots = 7 for total of $t = 8$ snapshots).

We also show impact of β on performance of SiameseLSTM with both the latent spaces in Figure 4 keeping other parameters constant ($w = 3$, $r_{BCGD} = 20$, $r_{STEP} = 128$, $d = 64$). β is varied as $\{0, 0.2, 0.4, ..., 1\}$. β characterizes the hidden state interaction matrix. As we can see SiameseLSTM is very robust for all datasets with respect to parameter β.

We also vary hidden state space dimension d as $\{8, 16, 32, 64, 128, 256\}$ and its effect on AUPR is plotted in Figure 3. Parameters other than d are kept constant ($\beta = 0.5$, $r_{BCGD} = 20$, $r_{STEP} = 128$, $w = 3$). For both BCGD+SiameseLSTM and STEP+SiameseLSTM, AUPR increases up to a certain value of d then drops a little bit and then saturates.

6 RELATED WORK

In this section we give a brief overview of previous works related to link prediction in static as well as dynamic networks.

Link prediction problem specifically in static networks has been extensively studied. We refer readers to recent survey papers [4, 23] for an exhaustive introduction to this field. Many approaches have used graph-based heuristics to model link probability between two nodes. Graph-based heuristics [2, 18, 19, 21, 33] model the link probability between two nodes as a function of their topological similarity. Common Neighbors [2], triad closures [33] are examples of local proximity heuristics whereas weighted shortest paths [19] is a global proximity metric.

The problem is quite different for dynamic networks because of their evolutionary nature [3]. Many approaches have used network topological information, which model link probabilities based on how close the nodes are in the network [7, 9, 10, 13, 20, 21, 34, 38, 39, 41]. For example, an auto regressive integrated moving average model was built to predict links in the next period based on previous time series data [16]. A local structural similarity based non-parametric model [28] was presented to predict link probabilities of node-pairs. Another method proposed stochastic block transition models that combined an extended Kalman filter with a local search algorithm to track dynamic networks [34].

Many approaches have relied on latent space modeling [1, 7, 9, 10, 12, 15, 30, 38, 39, 41], where the main idea is to learn a latent low-dimensional vector representation for each node such that nodes close to each other in the low-rank space will have link in future with higher probability that the ones that are far away. For example [12] extended the mixed membership block model to allow a linear Gaussian trend in the model parameters (DMMSB). [30] embedded longitudinal network data as trajectories in a latent

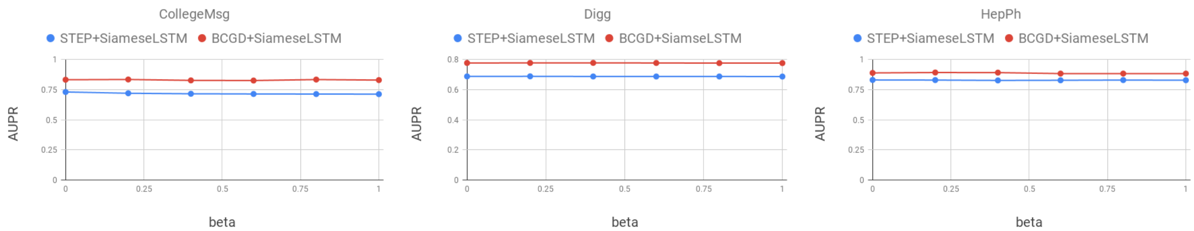

Figure 4: Hidden state interaction matrix (β) vs AUPR

Euclidean space. [9] did latent space modeling of road networks for analysis of traffic patterns and their evolution over time. In addition to network structures, [7–9, 22, 27, 29, 38, 39, 41] have also exploited temporal information as real-life networks are often very sparse and methods considering structural information alone may give poor performance.

Temporal information is extremely valuable to reveal the relationships between the current state of the network and its recent history. They assume the principle of temporal smoothness, which states that the current state of a network should not change drastically from its most recent history. For example [27] developed a dynamic mix-membership model to analyze the role transitions of nodes from a sequences of earlier time-stamps. Yet another approach embedded networks into a latent space in which the positions of nodes at consecutive time-stamps were constrained such that dramatic changes were unlikely to occur [29]. Similarly, a temporal latent space model was proposed to allow nodes in networks to move gradually as the network structures evolved over time[41].

Two very recent studies [7, 39] have also modeled temporal latent space utilizing both structural and temporal information. [39] has characterized network evolution directly as a function of time whereas [7] has characterized network evolution using a global transition matrix to reflect different types of evolutionary patterns which was difficult using time function used in [39] as the time function should be general enough to reflect the patterns [7]. STEP also preserves the deep network structure by considering the higher-order proximity among nodes.

All temporal latent space models uses a link function which maps node pairs to probability of link formation between them in future. Generally the link function used is a non parametric similarity measure between nodes and does not address temporal non-linear dependencies present in the evolution of networks. To the best of our knowledge, this is the first work which proposes a trainable temporal link function, which aims to learn temporal dependencies over sequence of previous graph snapshots for future link prediction.

7 CONCLUSION

In this paper, we propose a temporal link function, SiameseLSTM, for temporal link prediction in dynamic networks. The proposed link function learns from historical graph snapshots to better infer the probability that any two nodes will have link in future. To exploit Structural and Temporal evolution we use pre-trained latent space models and learn non-linear temporal dependencies for link prediction using SiameseLSTM to increase the performance of latent space models. We evaluate our method on three real-world datasets and show that SiameseLSTM link function when used in conjunction with latent space model gives better performance on link prediction task. We conclude that SiameseLSTM can be used with different latent space models with different optimization formulations to improve the predictive power for future link prediction task.

REFERENCES

[1] Evrim Acar, Daniel M Dunlavy, and Tamara G Kolda. 2009. Link prediction on evolving data using matrix and tensor factorizations. In *2009 IEEE International conference on data mining workshops*. IEEE, 262–269.
[2] Lada A Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social networks* 25, 3 (2003), 211–230.
[3] Charu Aggarwal and Karthik Subbian. 2014. Evolutionary Network Analysis: A Survey. *ACM Comput. Surv.* 47, 1, Article 10 (May 2014), 36 pages. https://doi.org/10.1145/2601412
[4] Mohammad Al Hasan and Mohammed J Zaki. 2011. A survey of link prediction in social networks. In *Social network data analytics*. Springer, 243–275.
[5] Amir Beck and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* 2, 1 (2009), 183–202.
[6] Tanya Berger-Wolf and David Kempe. 2007. A framework for community identification in dynamic social networks. Citeseer.
[7] Huiyuan Chen and Jing Li. 2018. Exploiting Structural and Temporal Evolution in Dynamic Link Prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. ACM, New York, NY, USA, 427–436. https://doi.org/10.1145/3269206.3271740
[8] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L. Tseng. 2007. Evolutionary Spectral Clustering by Incorporating Temporal Smoothness. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*. ACM, New York, NY, USA, 153–162. https://doi.org/10.1145/1281192.1281212
[9] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. 2016. Latent space model for road networks to predict time-varying traffic. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1525–1534.
[10] Daniel M Dunlavy, Tamara G Kolda, and Evrim Acar. 2011. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 5, 2 (2011), 10.
[11] Dóra Erdős, Rainer Gemulla, and Evimaria Terzi. 2014. Reconstructing graphs from neighborhood data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 8, 4 (2014), 23.
[12] Wenjie Fu, Le Song, and Eric P Xing. 2009. Dynamic mixed membership blockmodel for evolving networks. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 329–336.
[13] Sheng Gao, Ludovic Denoyer, and Patrick Gallinari. 2011. Temporal link prediction by integrating content and structure information. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 1169–1174.
[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[15] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. 2002. Latent space approaches to social network analysis. *Journal of the american Statistical association* 97, 460 (2002), 1090–1098.

[16] Zan Huang and Dennis KJ Lin. 2009. The time-series link prediction problem with applications in communication surveillance. *INFORMS Journal on Computing* 21, 2 (2009), 286–303.

[17] Martin Jaggi. [n. d.]. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization.

[18] Glen Jeh and Jennifer Widom. 2002. SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 538–543.

[19] Leo Katz. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18, 1 (1953), 39–43.

[20] Xiaoyi Li, Nan Du, Hui Li, Kang Li, Jing Gao, and Aidong Zhang. 2014. A deep learning approach to link prediction in dynamic networks. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 289–297.

[21] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology* 58, 7 (2007), 1019–1031.

[22] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L Tseng. 2008. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 685–694.

[23] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. 2017. A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)* 49, 4 (2017), 69.

[24] Aditya Krishna Menon and Charles Elkan. 2011. Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 437–452.

[25] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[26] Guo-Jun Qi, Charu C Aggarwal, and Thomas Huang. 2013. Link prediction across networks by biased cross-network sampling. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. IEEE, 793–804.

[27] Ryan A Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. 2013. Modeling dynamic behavior in large evolving graphs. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 667–676.

[28] Purnamrita Sarkar, Deepayan Chakrabarti, and Michael Jordan. 2012. Nonparametric link prediction in dynamic networks. *arXiv preprint arXiv:1206.6394* (2012).

[29] Purnamrita Sarkar and Andrew W Moore. 2006. Dynamic social network analysis using latent space models. In *Advances in Neural Information Processing Systems*. 1145–1152.

[30] Daniel K Sewell and Yuguo Chen. 2015. Latent space models for dynamic networks. *J. Amer. Statist. Assoc.* 110, 512 (2015), 1646–1657.

[31] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*. 843–852.

[32] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[33] Panagiotis Symeonidis, Eleftherios Tiakas, and Yannis Manolopoulos. 2010. Transitive node similarity for link prediction in social networks with positive and negative links. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 183–190.

[34] Kevin S Xu and Alfred O Hero. 2014. Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal of Selected Topics in Signal Processing* 8, 4 (2014), 552–562.

[35] Yang Yang, Ryan N Lichtenwalter, and Nitesh V Chawla. 2015. Evaluating link prediction methods. *Knowledge and Information Systems* 45, 3 (2015), 751–782.

[36] Jihang Ye, Hong Cheng, Zhe Zhu, and Minghua Chen. 2013. Predicting positive and negative links in signed social networks by transfer learning. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 1477–1488.

[37] Junming Yin, Qirong Ho, and Eric P Xing. 2013. A scalable approach to probabilistic latent space inference of large-scale networks. In *Advances in neural information processing systems*. 422–430.

[38] Wenchao Yu, Charu C Aggarwal, and Wei Wang. 2017. Temporally factorized network modeling for evolutionary network analysis. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 455–464.

[39] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Haifeng Chen, and Wei Wang. [n. d.]. Link Prediction with Spatial and Temporal Consistency in Dynamic Networks.

[40] Yongfeng Zhang, Min Zhang, Yiqun Liu, Shaoping Ma, and Shi Feng. 2013. Localized matrix factorization for recommendation based on matrix block diagonal forms. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 1511–1520.

[41] Linhong Zhu, Dong Guo, Junming Yin, Greg Ver Steeg, and Aram Galstyan. 2016. Scalable temporal latent space inference for link prediction in dynamic social networks. *IEEE Transactions on Knowledge and Data Engineering* 28, 10 (2016), 2765–2777.