TNE: a general time-aware network representation learning framework for temporal applications

TNE: A General Time-aware Network Representation Learning Framework for Temporal Applications

Huizhi Liang*, Thanet Markchom

Department of Computer Science, University of Reading, United Kindom

Abstract

Temporal dynamics such as short term and long term effects, recency effects, periodic and seasonal temporal factors in information networks are of great importance for many real-world applications. However, existing network embedding learning approaches mainly focus on semantic information or temporal phenomenon such as recency or dynamic process. They failed to have the capability of incorporating multiple temporal factors/phenomenon in information networks. To bridge the gap, this paper proposes a general time-aware network representation learning framework TNE for temporal applications. TNE contains a temporally annotated network TAN, a temporally annotated meta-path based random walk method, and a self-supervised embedding learning approach. We introduce temporal nodes and relations to existing information networks to construct TAN that can incorporate multiple temporal factors. We propose a temporally annotated meta-path based random walk approach to form a time-aware hybrid neighbourhood context that considers both semantic and temporal factors. Based on the time-aware context, self-supervised representation learning approaches are used to simultaneously preserve both semantic and temporal factors in embeddings. Extensive experiments of two large scale real-life datasets show that the proposed framework is effective in various temporal applications such as temporal similarity search and temporal recommendations.

Keywords: Representation Learning, Heterogeneous Information Network, User Profiling, Network Embeddings, Temporal Dynamics

1. Introduction

Temporal dynamics are natural and common in information networks, which includes recency, short term and long term effects, periodic, seasonal and other effects. Recency effect is an important temporal phenomenon in information networks. For example, people in social media tend to discuss more recent topics than topics 7 days ago [34]. Long term and short term effects are common phenomenon for human users' information needs and topic preferences [59]. Commonly, a user has long term preferences/interests in certain topics that seldom change or remain a long time period [25]. A user is also affected by his/her short-term interests due to breaking news events such as pandemic diseases or special personal occasions/changes such as birthday or having babies. Other temporal dynamics include periodic effects such as a user's active time in a day/month/year and seasonal effects [59]. Temporal dynamics are of great importance for many real-world temporal applications [25] such as similarity search [19], recommendations [25], link prediction [67], influence modelling [30], community detection [4, 21, 63], relation reasoning [53], and other network analysis tasks [54].

Heterogeneous information network have been widely used to model multi-typed entities and their interactions. For example, in a recommender system, there are multiple entities including users, items, and affiliated entities such as tags and item categories. These entities and relations form a heterogeneous information network [8]. Network embedding learning maps nodes in a network to low-dimensional embedding vector representations and can effectively preserve the network structure [46] and other information such as properties [29] and attributes [15]. The embedding vectors can be directly used as node features in various network processing and analysis tasks [18] such as node classification, node clustering, similarity search [19], recommendation [60], and link prediction [67]. The embedding learning approaches can be categories as self-supervised learning methods such as skip-gram [41] model, transformer model [17], and end-end network embedding learning approaches such as graph convolutional neural network [24] and graph attention network [56]. Self-supervised approaches can work on unlabelled data. They have been popularly used to learn network representations in many application areas [13, 18, 17].

Static network embedding learning approaches are the main stream of embedding learning approaches [8]. For heterogeneous information networks, the semantics of the networks are very important [8]. The existing heterogeneous information network representation learning approaches mainly focus on the semantic factors of information networks. For example, using meta-path based approaches to form heterogeneous neighborhood nodes context and learn nodes embeddings [51, 13,

*Corresponding author

Email addresses: huizhi.liang@reading.ac.uk (Huizhi Liang), thanet.mar@gmail.com (Thanet Markchom)

14]. Recently, some work explored the problem of learning dynamic node embeddings from temporal networks [50]. These approaches focus on the dynamic evolution of network [50], the sequence order of edges [42], and the change and updates of edges and nodes [66]. The existing static and dynamic network representation learning approaches fail to incorporate or model the multiple temporal factors in information networks to learn network representations. The major challenges of modeling temporal dynamics and learning time-aware embeddings in information networks include:

First, how to represent various kinds of temporal factors in existing information networks? In heterogeneous information networks, each type of node might have different kinds of temporal factors. For example, the temporal factors of item nodes are usually different with those of user and tag nodes. Even for the same type of nodes, one node might have different temporal factors with another. For example, some item nodes have stronger seasonal effect than other item nodes. Some temporal factors are global phenomenon for the whole network while some are local effects that are only applicable to part of the network [59]. All these increase the complexity of this problem.

Second, how to consider both semantic and temporal factors simultaneously to form neighborhood context and learn network embeddings? Real-life applications usually need to consider multiple factors in information networks, including both semantic and temporal factors. For example, a user wants to find a comedy movie and suitable to watch in weekend. This query requires the consideration of those items with the semantic meaning “comedy” and temporal factor “weekend”. Modeling the interaction of various types of factors in information networks and preserve these factors in network embeddings brings challenges to existing network embedding learning approaches.

To address these challenges, we introduce a general time-aware network representation learning framework TNE for temporal applications. TNE contains three components: 1) a temporally annotated network TAN; 2) a temporally annotated meta-path based random walk method; 3) a self-supervised approach for embedding learning. In this framework, we introduce temporal nodes and relations to existing information networks to construct TAN that can incorporate multiple temporal factors. The nodes and edges of information networks are classified into two types: semantic entity nodes and relations, and temporal nodes and relations. The semantic entity nodes and relations are traditional nodes and relations in information networks. The proposed framework defines the concept of temporally annotated meta-path. Following a random walk approach based on temporally annotated meta-paths, we can consider both semantic and temporal factors to find neighbor nodes and thus form a time-aware hybrid neighborhood context. The time-aware hybrid context will be used to facilitate self-supervised representation learning approaches to learn network embeddings that simultaneously preserve both semantic and temporal factors. The contribution of this paper is as below:

- This work proposes a general framework to learn time-aware embeddings for temporal applications.
- This work introduces temporal nodes and temporal relations to information networks to construct temporally annotated information networks (TAN), for the purpose of incorporating temporal dynamics in information networks.
- This work proposes temporally annotated meta-paths guided random walks to form profound hybrid context that considers temporal and semantic factor simultaneously to learn static time-aware network embeddings.
- This work conducted extensive experiments on real-life datasets for various kinds of temporal applications including temporal similarity search and temporal recommendation tasks.

The rest of the paper is organised as follows. In Section 2, the related work will be briefly reviewed. Then the proposed framework will be discussed in details in Section 3. In this section, the definition and construction of temporally annotated network (TAN) will be discussed first. Then, we will define temporally annotated meta-path and discuss the time-aware hybrid neighborhood forming approach following temporally annotated meta-path random walk. In Section 4, the temporal applications including temporal similarity search and temporal recommendation will be discussed. The experiments and results will be discussed in Section 5. The conclusions will be given in Section 6.

2. Related Work

Modelling time drifting data is a central problem in data mining [25]. Temporal analysis contains a wide range of analytic techniques. For example, computing correlations between time series [12, 64], detecting global and local temporal dynamics in user behaviors [25], identifying lagged effects [9], recency phenomenon [34], periodic or seasonal effects [59], recognizing user engagement patterns [28] and temporal patterns for patient treatment [5, 27]. Temporal dynamics have been discussed in various applications including similarity search [19], recommender systems [25], link predictions [67], dynamic community structure analysis [45], and attack detection [7]. Similarity search and recommender systems and are two important applications. Accuracy is one important evaluation metrics. Precision and Recall are commonly used to evaluate the accuracy of Top-N recommendation task, while Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are widely used to measure that of rating prediction task of recommender systems. Besides accuracy, diversity metrics such as Coverage and Novelty are popularly used for information retrieval and Top-N recommendation tasks [26, 20, 10].

Modelling temporal dynamics is one important research direction for these applications. Some time-aware similarity search have been proposed. For example, Daud et al. [11] proposed a method to incorporate temporal data in topic models for temporal expert search. He et al. proposed to exploit transitive similarity and temporal dynamics for similarity search in heterogeneous information networks [19]. Some temporal Collaborative filtering based approaches have been proposed to make recommendations [62, 60, 25, 38, 33] and attack detection [7].

For example, Koren et al. proposed a matrix factorisation approach to consider the temporal dynamics of users and items to improve recommender systems [25]. Cai et al. proposed an approach to detecting shilling attacks in recommender systems based on analysis of the temporal dynamics of user rating behavior [7]. Other temporal dynamic modeling approaches include recency-aware [34], time interval-aware [31], long term and short term-aware [59], time-weighted meta-path [36] recommendation approaches. These approaches are neither from the perspective of learning network embeddings nor capable of incorporating various kinds of temporal factors in their models.

Network embedding learning is an emerging network analysis paradigm. It assigns nodes in a network to low-dimensional embedding vector representations and effectively preserves the network structure and other information [18]. The embedding vectors can be directly used as node features in various network processing and analysis tasks such as node classification, node clustering, network visualisation [18, 13, 66], social influence analysis [30], similarity search [19], recommendation [60, 33], and link prediction [67]. Existing network embedding learning approaches include self-supervised approaches and supervised end-end embedding learning approaches. self-supervised approaches such as node2vec [18] and deepWalk [46] were inspired by the skip-gram model of Word2Vec [41] in natural language processing area. End-end network embeddings learning models include graph convolutional neural network [24], graph attention network [56], network embedding based recommender systems [49]. They are usually supervised approaches and learn embeddings from prediction tasks directly such as node classification task.

The majority of research in this area focuses on static network embedding learning approaches. These approaches can be categorised as structure preserving [46], property preserving [29], and attributes preserving [15] approaches. Network embedding usually refer to node embeddings. There are some research focuses on learning the latent representations for edges or relations [58, 35]. The node embedding learning approaches can be classified as homogeneous network embedding learning approaches and heterogeneous information network learning approaches. For a homogeneous network with only one type of entity and relation, these models capture the co-occurrence phenomenon of nodes in a hop-n random walk context [18, 46]. For heterogeneous information networks with multiple type of entities and relations, meta-paths based approaches are popularly used to make recommendations [8, 51, 32]. The meta-path similarity measure framework [51] of heterogenous information networks provides a powerful mechanism for a user to measure the possibility of an unobserved user-item interaction in the information network under different semantic assumptions. Dong et al. proposed a metapath2vec[13] approach to learn network embeddings for heterogeneous information network. Recently, Fu et al. proposed a metapath aggregated graph neural network for heterogeneous graph embeddings [14]. These approaches fail to model or discuss the temporal factors in the information network. How to incorporate temporal factors in heterogeneous information networks and learn the latent network representations still need to be explored.

Recently, some work explored the problem of learning dynamic node embeddings from temporal networks [54, 37, 52, 44, 39, 16, 48, 23]. For example, some work proposed to approximate the dynamic network as a sequence of discrete static snapshot graphs [50]. The work of Nguyen et al. [42] considers the temporal order of neighbour edges to learn continuous time dynamic network embeddings. These approaches mainly model the dynamic evolution of network [65], the sequence order of edges [42], and the change and updates of edges and nodes and weights [66]. However, these dynamic graph embedding methods are mainly focusing on network dynamics or the time-evolving phenomenon of nodes (e.g., adding, deletion, updating of nodes), which is different with our approach that focuses on temporal dynamics such as recency, long term, short term, and periodic/seasonal phenomenon.

Different with these dynamic embedding learning approaches, the proposed approach is a static network embedding learning approach but focus on preserving both semantic and temporal factors of the network in embeddings. In summary, the existing approaches mainly focus on semantic factor or one type of temporal factor. They lack the capability of incorporating and modeling multiple temporal factors in information networks to learn network representations. To bridge the gap, the proposed approach is a general framework that can incorporate multiple temporal factors and simultaneously consider both semantic and temporal factors to learn the latent network representations for heterogeneous information networks.

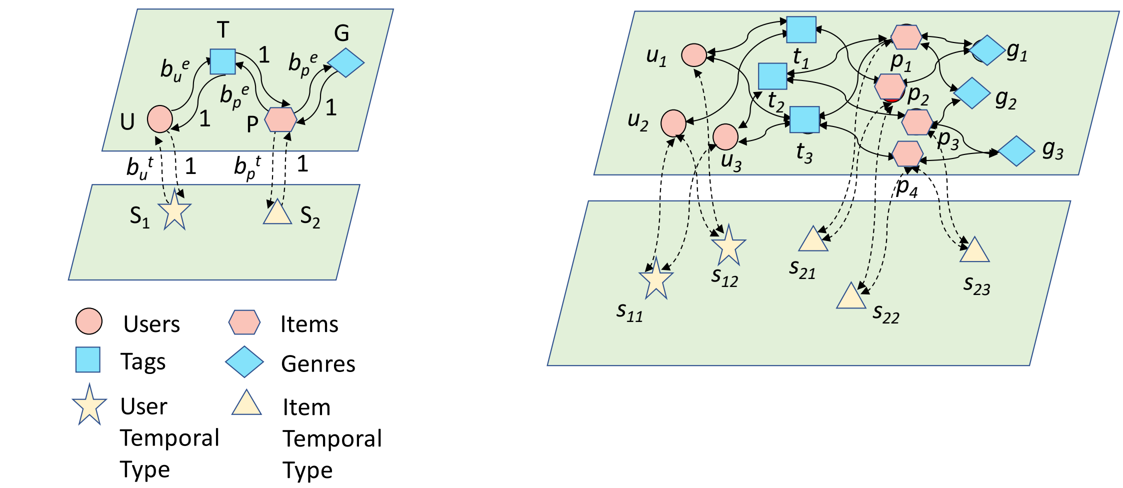
3. TNE: A General Time-aware Network Representation Learning Framework

In this section, we discuss the proposed framework. The goal of this framework is to maximise the likelihood of preserving both semantic and temporal factors of a given information network. We first discuss how to construct temporally annotated networks TAN and how to design the weight functions. Then we discuss how to form time-aware hybrid neighbourhood context that considers both semantic and temporal factors in TAN. Based on the hybrid context, we discuss how to learn time-aware network representations. Note this paper do not differentiate the term “network” and “graph”. The summary of notations is shown in Table 1.

3.1. The construction of TAN

To make an information network capable to incorporate multiple temporal factors, we introduce the concept of temporal nodes and temporal relations to information networks. Their definitions are given below.

Definition 1: Temporal nodes and **Temporal relations**. Let f be a temporal analysis function, v and v' are two nodes in a graph. If the nodes have the same value $f(v)$=$f(v')$= n, then nodes v and v' can be put into the same temporal bin/category with the same value n. We call n a *temporal node*. The relation between nodes v and n is called a *temporal relation*.The same with other semantic relations, temporal relations have directions. The relation between nodes v' and n is another temporal relation. The function f is call a *temporal type*.



(a) Temporally Annotated Network Schema (b) Temporally Annotated Network

Figure 1: Example Temporally Annotated Network and Schema

Table 1: Summary of notations

Notations	Descriptions	Notations	Descriptions
$\mathbb{C}$	schema	$\mathcal{G}$	graph or network
$\mathbb{N}$	type nodes set	$\mathcal{V}$	nodes set
$\mathbb{R}$	relation type edges set	$\mathcal{L}$	edges set
$\mathbb{W}$	weight functions for relation type edges	$\mathcal{W}$	weight functions for relation edges
$\mathcal{N}_i$	a type node	v	a node
$\mathcal{R}_{ij}$	hop-1 relation of type $\mathcal{N}_i$ and $\mathcal{N}_j$	$l_{vv'}$	relation between nodes v and v'
$\mathcal{R}_{ij}^{-1}$	reversed relation of $\mathcal{R}_{ij}$	$\mathcal{Z}$	function of obtaining the relation type of an edge
W_{ij}	weight of relation $\mathcal{R}_{ij}$	$\mathcal{W}_{vv'}$	weight for relation $l_{vv'}$
$\mathbb{E}$	semantic entity type nodes set	$\mathcal{E}$	semantic entity nodes set
$\mathbb{T}$	temporal type nodes set	$\mathcal{T}$	temporal nodes set
$\mathbb{E}^{\tau}$	temporally annotated entity types set	$\mathcal{E}^{\tau}$	temporally annotated entity nodes set
b_i^e	semantic entity relation weight	O_v	outgoing edges of node v
b_i^t	temporal relation weight	O_v^e	outgoing edges with semantic relations of node v
b_i	ratio of b_i^e over b_i^t	O_v^t	outgoing edges with temporal relations of node v
b_d	temporal decay weight	$\mathcal{M}$	a meta-path
$T(l)$	timestamp of edge l	$\mathscr{P}$	random walk transition probability
t	timestamp	ρ	random walk out-degree impact parameter
t_m	minimum timestamp of all edges of of O_v	β	prior weight for starting node
$\mathcal{C}$	context	U	users set
$\mathcal{C}_t$	temporal hybrid context	u_i	a user in U
$\mathcal{V}_{\mathcal{C}}$	nodes in the context $\mathcal{C}$	P	items set
d	size of embedding vector or latent space	p_j	an item in P
$\mathbf{W}$	nodes weight matrix	$\mathcal{O}_i$	observed items of u_i
$\mathbf{V}$	d-dimensional latent space	$\tilde{N}(v)$	top k similar nodes of v
$\mathbf{R}$	non-negative real values	$\mathcal{L}_j$	set of users that have rated p_j
Z^{+}	positive integer values	k	neighborhood size

If the temporal function is only related to one node type, we call it a global temporal type. It does not dependent or conditioned on other node types. Otherwise, we call it a local temporal type. For example, the "session" concept is a local temporal type. Those items that have the same time span and accessed by the same user will be put in the same time bin or connect to the same temporal node.

Example 1: Temporal nodes and relations. Let $\mathscr{T}$ denote the time series of user rating behavior of an item. For two given items v and v', $\mathscr{T}(v)$={19-8-2018, 24-12-2019, 20-4-2020}, $\mathscr{T}(v')$={19-8-2015, 25-12-2018, 18-4-2020}. Assume the temporal analysis function f is to get the "year" of the most recent time stamp of these items. $f(\mathscr{T}(v))$=$f(\mathscr{T}(v'))$="2020". "2020" is a temporal node, the temporal relation includes the edge between "2020" and v, and the edge between "2020" and v'. Nodes v and v' are neighbor nodes and connected with each other via the temporal function f. This function is a global temporal type.

We can use various temporal feature engineering functions or approaches to extract the temporal features of nodes. Temporal type can be a simple or complicated temporal analysis function, for example, tendency [6] and periodic analysis function [6], temporal distance measure function such as Dynamic Time Warping Distance [22], Longest Common Sub-sequence [1]. It can be an Allen's interval algebra function such as "overlap", "meets", "during", "starts", "finishes" [3], time interval distribution function [47], burst pattern detection function [43], outlier detection function [61]. It also can be a time series modeling function such as moving average, auto-regression, and ARIMA model [6].

A network schema is a graph that defines node types and relation types of a network. It defines the semantics of a network. For example, for a network schema of a movie heterogeneous information network, the nodes of network schema represent semantic entity types such as Users, Items, and Tags. The edges represent semantic relation types of two types of nodes such as User-Item relation and Item-Tag Relation types. The temporal nodes and relations are used to annotate the semantic entity nodes, we call this network *Temporally Annotated Networks (TAN)*. To be general, a TAN network schema is a weighted and directed graph. It is defined as follows:

Definition 2: TAN schema. Let $\mathbb{C}$ denote a TAN schema, $\mathbb{C}$ =($\mathbb{N}, \mathbb{R}, \mathbb{W}$), consists of type nodes $\mathbb{N} = \{\mathcal{N}_1, \mathcal{N}_2, ..., \mathcal{N}_m\}$, a set of relation type edges $\mathbb{R}$, and a weight function $\mathbb{W} : \mathbb{R} \to \mathbf{R}$ that denotes a non-negative weight function to map each relation type edge to a real value $\mathbf{R}$. Type nodes include both semantic entity types $\mathbb{E}$ and temporal types $\mathbb{T}$. $\mathbb{R}$ includes relations among type nodes, i.e. $\mathbb{R} = \{\mathcal{R}_{ij} | i = 1...|\mathbb{N}|, j = 1...|\mathbb{N}|, i \neq j\}$, where $\mathcal{R}_{ij}$ denotes the hop-1 connection relation of type $\mathcal{N}_i$ and $\mathcal{N}_j$. $\mathcal{R}_{ij}^{-1}$ denotes the reversed relation of $\mathcal{R}_{ij}$. W_{ij} denote the weight of relation $\mathcal{R}_{ij}$. Relation types include entity relation types that connect two entity type nodes and temporal relation types.

Example 2: TAN Schema. Figure 1 (a) shows an example TAN schema. It has entity types Users U, Items P, Tags T, Genres G. Entity relation types include $\mathcal{R}_{GP}$, $\mathcal{R}_{GP}^{-1}$, $\mathcal{R}_{TU}$, $\mathcal{R}_{TU}^{-1}$, $\mathcal{R}_{TP}$, $\mathcal{R}_{TP}^{-1}$. It has user temporal types S_1 for user nodes and item temporal types S_2 for item nodes. S_1 is defined to get the latest active "year" of users. S_2 is defined to get the "month" of the most recent time stamp of items. Temporal Relation types include $\mathcal{R}_{S_1U}$, $\mathcal{R}_{S_1U}^{-1}$, $\mathcal{R}_{S_2P}$, $\mathcal{R}_{S_2P}^{-1}$.

Based on the schema $\mathbb{C}$, we can define a network. The definition of TAN is as below:

Definition 3: TAN. It is defined as a weighted and directed graph $\mathcal{G}$ =($\mathcal{V}, \mathcal{L}, \mathcal{W}$) with schema $\mathbb{C}$. Each node $v \in \mathcal{V}$ belongs to one particular node type of $\mathbb{N}$. Each edge $l \in \mathcal{L}$ belongs to a particular type of relation of $\mathbb{R}$. Let $\mathcal{E}$ denote the set of entity nodes, $\mathcal{T}$ denotes the set of temporal nodes, $\mathcal{V} = \mathcal{E} \cup \mathcal{T}$. Given a node type $\mathcal{N}_i \in \mathbb{N}$, $\mathcal{E}_i$ denotes a set of nodes belong to node type $\mathcal{N}_i$. Let $\mathcal{L}_e$ denote the set of entity relations of $\mathcal{G}$, $\mathcal{L}_t$ denotes the set of temporal relations of $\mathcal{G}$, $\mathbb{L} = \mathcal{L}_e \cup \mathcal{L}_t$. $\mathcal{W} : \mathcal{L} \to \mathbf{R}$ denotes a non-negative weight function for relations $\mathcal{L}$. There is a mapping between the weight function $\mathcal{W}$ of $\mathcal{G}$ and the weight function $\mathbb{W}$ of schema $\mathbb{C}$. For two nodes v and v' of graph $\mathcal{G}$, assume v is a node of type $\mathcal{N}_i$ and v' is a node of type $\mathcal{N}_j$. Let $l_{vv'}$ be a relation of type $\mathbb{R}_{ij}$, $\mathcal{W}_{vv'}$ be the weight for relation $l_{vv'}$, $\mathcal{W}_{vv'} = W_{ij}$.

Example 3: TAN. Figure 1 (b) shows an example TAN defined by schema $\mathbb{C}$ in Figure 1 (a). It has users $U = \{u_1, u_2, u_3\}$, tags $T = \{t_1, t_2, t_3\}$, items $P = \{p_1, p_2, p_3, p_4\}$, and genre nodes $G = \{g_1, g_2, g_3\}$. It also has temporal nodes $S_1 = \{s_{11}, s_{12}\}$, and $S_2 = \{s_{21}, s_{22}, s_{23}\}$.

The problem of **Network Representation Learning** is defined as: given a temporally annotated information network TAN $\mathcal{G}$, the task is to learn the d-dimensional latent representations $\mathbf{V} \in \mathbf{R}^{|\mathcal{E}|*d}$ for nodes $\mathcal{V}$ that are able to capture the semantic and temporal relations among them, $d \ll |\mathcal{E}|$.

3.2. The Design of Weight Functions

In this section, we discuss how to design weight functions. In a TAN, assuming that temporal factors have the same importance with the semantic factors is not always true in many scenarios. To be able to model the different importance level of temporal factors more precisely, we use a weight function $\mathbb{W}$ to assign weights to different type of relations at the schema level. Also, to be able to differentiate the importance of those edges of the same type of relations (e.g., temporal decay), we use a weight function $\mathcal{W}$ to assign weights to different edges at the graph level.

For easy discussion, we introduce the concept *temporally annotated entity type* and *typical entity type*. If a semantic entity type is directly connected to one or more temporal types, it is called a *temporally annotated entity type*. Otherwise, it is called a *typical entity type*. Let $\mathbb{E}^\tau$ denote the set of *temporally annotated entity types* at schema $\mathbb{C}$, $\mathcal{E}^\tau$ denote the set of *temporally annotated entity nodes* at graph $\mathcal{G}$. For example, in Figure 1 (a), item type P is a *temporally annotated entity type*, $P \in \mathbb{E}^\tau$, and item nodes p_1 is a *temporally annotated entity node*, $p_1 \in \mathcal{E}^\tau$. For a given edge $l_{vv'} \in \mathcal{L}$ of $\mathcal{G}$, let $\mathcal{N}_i$ be the node type of v, $\mathcal{N}_j$ be the node type of v', $\mathcal{R}_{ij}$ is the edge that connects type $\mathcal{N}_i$ and $\mathcal{N}_j$ at schema level $\mathbb{C}$. We discuss how to assign weight W_{ij} for $\mathcal{R}_{ij} \in \mathbb{R}$ at the schema level and $\mathcal{W}_{vv'} \in \mathcal{L}$ at the graph level. We discuss the weight setting for both semantic and temporal relations.

3.2.1. Semantic Relation

Semantic relations connect two semantic entities nodes. Both the source and target nodes can be either a *typical entity type node* or a *temporally annotated entity type node*. Based on the source node types, we discuss the following two cases:

- i) The source type $\mathcal{N}_i$ is a *typical entity type node*, $\mathcal{N}_i \in \mathbb{E} - \mathbb{E}^\tau$. Based on the definition of *typical entity type* , the target node $\mathcal{N}_j$ only can be an semantic entity type (either a *typical entity type* or a *temporally annotated entity type*, $\mathcal{N}_j \in \mathbb{E}$). For this case, $\mathcal{R}_{ij}$ is an entity relation and l_{ij} connects two semantic entity types. Although the weight for each relation type can be set differently by domain experts, assuming equal weight of each entity relation is a common setting in heterogeneous information networks [8]. For simplicity, we follow the traditional heterogeneous information networks setting, let weight $W_{ij} = 1$ and $\mathcal{W}_{vv'} = 1$ for this case. For example, in Figure 1 (a), Genres G and Tags T are *typical semantic entity types*, the weights of W_{GP} ,W_{TU}, and W_{TP} are set to 1.
- ii) The source type $\mathcal{N}_i$ is a *temporally annotated entity type*, $\mathcal{N}_i \in \mathbb{E}^\tau$. The target node $\mathcal{N}_j$ can be either a temporal type or a semantic entity type. We discuss the case of $\mathcal{N}_j$ is a semantic entity type here while the case of $\mathcal{N}_j$ is a temporal type in the following subsection 3.2.2. In the case of $\mathcal{N}_j$ is a semantic entity type, $\mathcal{R}_{ij}$ is an entity relation and l_{ij} connects two semantic entity types. To differentiate *temporal relations* and *semantic relations*, we can assign different weights for them. At the schema level, the source type $\mathcal{N}_i$ has two types of outgoing relations. Let b_i^e be the weight for the outgoing semantic entity relation, $0 \leq b_i^e \leq 1$. If $\mathcal{N}_i$ has multiple outgoing semantic entity types, we can assign different weights to different types of outgoing semantic entity relations and set their weight summation to b_i^e. If the target $\mathcal{N}_j$ is a semantic entity type, $W_{ij} = b_i^e$ and $\mathcal{W}_{vv'} = b_i^e$. For example, in Figure 1 (a), P and U are temporally annotated types, G and T are semantic entity types, the weights for $W_{GP}^{-1} = b_p^e$, $W_{TU}^{-1} = b_u^e$, $W_{TP}^{-1} = b_p^e$.

3.2.2. Temporal Relation

Temporal relations connect semantic entities types and temporal types. Based on the source node types, we discuss the following two cases:

- i) Source type $\mathcal{N}_i$ is a *temporally annotated entity type* $\mathcal{N}_i \in \mathbb{E}^\tau$. The target node $\mathcal{N}_j$ can be either a temporal type or a semantic entity type. We discuss the case of $\mathcal{N}_j$ is a temporal node here. For this case, the relation is a *temporal relation*. At the schema level, the source type $\mathcal{N}_i$ has two types of outgoing relations. Let b_i^t be the weight for the temporal relation, $0 \leq b_i^t \leq 1$. $0 \leq b_i^t \leq 1$. Similarly, if $\mathcal{N}_i$ has multiple outgoing temporal relation types, we can give different weights to different types of outgoing temporal relations and let their summation to b_i^t. For the purpose of normalising the total outgoing relation weights of source type $\mathcal{N}_i$, we set $b_i^t + b_i^e = 1$. $\mathcal{N}_j \in \mathbb{T}$, $W_{ij} = b_i^t$ and $\mathcal{W}_{vv'} = b_i^t$. For example, in Figure 1 (a), U and P are temporally annotated nodes, S_1 and S_2 are temporal nodes, $W_{S_1U}^{-1} = b_u^t$, $W_{S_2P}^{-1} = b_p^t$.
- ii) Source type $\mathcal{N}_i$ is a temporal type $\mathcal{N}_i \in \mathbb{T}$. Based on the definition of *temporally annotated entity type*, the target node $\mathcal{N}_j$ only can be a *temporally annotated entity type* $\mathcal{N}_j \in \mathbb{E}^\tau$. As a temporal type can connect to one or multiple semantic entity types, we can set different weights for different types of temporal relations by domain experts. For simplicity and fitting the temporal relations to the same weighting system as entity relations, we set $W_{ij} = 1$ at the schema level. For example, in Figure 1 (a), S_1 and S_2 are temporal nodes, U and P are temporally annotate entity types, $W_{S_1U} = 1$, $W_{S_2P} = 1$.

To sum up, the weight function W_{ij} for $\mathcal{R}_{ij} \in \mathbb{R}$ at schema level is defined as:

$$W_{ij} = \begin{cases} 1 & \text{if } (\mathcal{N}_i \in \mathbb{E} - \mathbb{E}^\tau, \mathcal{N}_j \in \mathbb{E}) \text{ or} \\ & (\mathcal{N}_i \in \mathbb{T}, \mathcal{N}_j \in \mathbb{E}^\tau\ ) \\ b_i^e & \text{if } (\mathcal{N}_i \in \mathbb{E}^\tau, \mathcal{N}_j \in \mathbb{E}) \\ b_i^t & (\text{if } \mathcal{N}_i \in \mathbb{E}^\tau, \mathcal{N}_j \in \mathbb{T}) \end{cases} \tag{1}$$

Note, the weight $W_{ij} = 1$ can be replaced by weights given by experts. The weight function on the schema level can be easily mapped to the weight function at network level. Different with the weight setting of temporal relations at the schema level, the weight of temporal relations at graph level should consider the time decay effect.

3.2.3. Temporal decay

For a given edge $l_{vv'}$ of $\mathcal{G}$, we discuss how to set up temporal decay effects at graph level generally. We discuss three kinds of temporal decay methods:

Unbiased decay. This kind of decay ignored the temporal differences between different outgoing nodes. Every outgoing semantic entity nodes of source node v is equally important. There is no temporal decay for outgoing temporal relations. For simplicity and fitting the temporal relations to the same weighting system as semantic relations, we set $\mathcal{W}_{vv'} = 1$.

Exponential decay. We can incorporate different temporal decay effects for outgoing semantic entity nodes with different times tamps. For example, items have the same time function value (i.e., connect with the same temporal node in TAN $\mathcal{G}$), but they have different time stamps. Thus we can use decay functions to assigning higher weights for most recent items while those old items will get lower weights. Let l be an edge between a temporal node v of type $\mathcal{T}_i$ ($v \in \mathcal{T}_i$) and a temporally annotated semantic entity node v' of type $\mathcal{N}_j$ ($v' \in \mathcal{E}_j^\tau$), O_v denote all the out edges of node v of relation type $\mathcal{R}_{ij}$, $T\left(l\right)$ is the timestamp of edge l, t_m is a minimum timestamp of all edges of O_v. The weight can be calculated as:

$$\mathcal{W}_{vv'} = \frac{exp\left(T\left(l\right) - t_m\right)}{\sum_{l' \in O_v} exp\left(T\left(l'\right) - t_m\right)} \tag{2}$$

This favours the edges come late in time. In other words, recency matters.

Linear decay. When the time difference between two timewise consecutive edges is large, it can sometimes be helpful to map the edges to discrete time steps. Let $g : \mathcal{L} \rightarrow Z^+$ be a function that sorts the edges in ascending order by time in the graph. g maps each edge to an index with $g(l)$=1 for the earliest edge l. $g(l')$=2, if l' is a second earliest edge in $\mathcal{L}$. In this case, each edge $l \in g(O_v)$ will be assigned the probability:

$$\mathcal{W}_{vv'} = \frac{g\left(l\right)}{\sum_{l' \in O_v} g\left(l'\right)} \tag{3}$$

Let b_d denote the temporal decay weight. After considering the temporal decay effect of temporal relations, the weight $\mathcal{W}_{vv'}$ for edge $l_{vv'} \in \mathcal{L}$ at graph level is defined as below:

$$\mathcal{W}_{vv'} = \begin{cases} 1 & \text{if} \, (v \in \mathcal{E}_i - \mathcal{E}_i^{\tau}, v' \in \mathcal{E}_j) \\ b_d & \text{if} \, (v \in \mathcal{T}_i, v' \in \mathcal{E}_j^{\tau}) \\ b_i^e & \text{if} \, (v \in \mathcal{E}_i^{\tau}, v' \in \mathcal{E}_j) \\ b_i^t & \text{if} \, (v \in \mathcal{E}_i^{\tau}, v' \in \mathcal{T}_j) \end{cases} \tag{4}$$

$$b_d = \begin{cases} 1 \\ \frac{exp(T(l) - t_m)}{\sum_{l' \in O_v} exp(T(l') - t_m)} \\ \frac{g(l)}{\sum_{l' \in O_v} g(l')} \end{cases} \tag{5}$$

Note, the same with W_{ij} on the schema level, the weight $\mathcal{W}_{ij} = 1$ also can be replaced by weights given by experts.

3.3. Hybrid Neighbourhood Context Forming

For self-supervised representation learning models, "context" is a key concept that defines under which condition or circumstances, a set of components are co-occurred together or connect with each other[17, 41]. Context also defines the input and output of a model. Usually, the input is one or some components of the context and the output is the other components of the same context. For example, for language skip-gram model [41], the context is a window of words in a sentence. For homogeneous information network, the context of a given node of a graph is the hop-1 neighbours of that node [18]. For heterogeneous information network, the context usually has semantic meaning [8, 32]. For example, meta-path ($UPUP$) can find items that are used by those users that at least shared one common item with a given user, which forms a user based collaborative filtering neighbourhood [32]. Meta-pah (UTP) forms a content based neighbourhood, while the items in the context has the same tag with a given user [32].

Meta-path based random walk is popularly used to form context for heterogeneous information networks[8, 32]. For a TAN, as the nodes with the same temporal effect can be considered as similar nodes in terms as temporal effect, we can consider temporal effect as one type of context to capture their co-occurrence phenomenon and find neighbour nodes. To consider both semantic and temporal factors in the same context, we can use an "or" operation in a meta-path to find a hybrid context. For a TAN, we define the context as a hybrid context that includes both semantic and temporal factors. To better describe the hybrid context, we define the concept of temporally annotated meta-path. We introduce symbol "$\oplus$" to represent an "or" operation of a temporally annotated type that has both outgoing semantic entity relations and temporal relations.

Definition 4: Temporally Annotated Meta-path. *A temporally annotated meta-path $\mathcal{M} = (i\{b_i^e x \oplus b_i^t y\}l...j)$ is a sequence of node types and relation types in schema $\mathbb{C}$ of a TAN. It contains at least one temporally annotated type and one temporal relation type. For simplicity, we ignore the weights in the annotation and use $(i\{x \oplus y\}l...j)$ to denote $\mathcal{M}$.*

$$\mathcal{M} = \left(i\{b_i^e x \oplus b_i^t y\}l...j\right) = \mathcal{N}_i \left\{ \begin{array}{c} \xrightarrow[b_i^e]{\mathcal{R}_{ix}} \mathcal{N}_x \xrightarrow[W_{xl}]{\mathcal{R}_{xl}} \\ \xrightarrow[b_i^t]{\mathcal{R}_{iy}} \mathcal{N}_y \xrightarrow[W_{yl}]{\mathcal{R}_{yl}} \end{array} \right\} \mathcal{N}_l \xrightarrow[W_{l.}]{\mathcal{R}_{l.}} ... \mathcal{N}_j. \tag{6}$$

Example 4: Temporally Annotated Meta-path. In Figure 1, $\mathcal{M} = (P\{T \oplus S_2\}P)$ is a temporally annotated meta-path. It forms a hybrid neighbourhood. In this context, items are either have the same temporal factor value or have the same tags with each other. For item p_1, following this meta-path, item p_2 that has the same temporal factor value with p_1 has the probability of b_p^t being put in the same context with p_1, while item p_3 that has the same tag with p_1 has the probability of b_p^e being put in the same context with p_1.

Starting from one source node, we can walk along a *temporally annotated meta path* to get a set of neighbour nodes. Different with those papers only consider symmetric meta paths, in this paper, we do not restrict the meta paths be symmetric, as the connectivity of different types of nodes are important in many applications. For a given temporally annotated meta-path $\mathcal{M} = (i\{b_i^e x \oplus b_i^t y\}l...j)$, starting from source node v with type $\mathcal{N}_i$, it can walk along the path and reach target nodes with type $\mathcal{N}_j$. The walk transition probability from node v to another node v' with type $\mathcal{N}_l$ following $\mathcal{R}_{il}$ can be calculated based on the ratio of the weight of the edge $l_{vv'}$ over the total number of weights of all outgoing edges of v with relation $\mathcal{R}_{il}$. This paper assumes each connection (i.e., edge) between any two nodes with the same type of relation without temporal decay is equally important. For example, in Figure 1, $l_{p_1 g_1}$ and $l_{p_1 g_2}$ have the same weight because they have the same type of relation $\mathcal{R}_{PG}$. Let $\mathcal{Z}(l_{vv'})$ be the function of obtaining the relation type of edge $l_{vv'}$, $\mathcal{Z}(l_{vv'})$ = $\mathcal{R}_{ij}$. Let O_v be the set of outgoing edges of v with relation $\mathcal{R}_{ij}$ of meta path $\mathcal{M}$, $O_v = \{v'' \in \mathcal{V} : l_{vv''} \in \mathcal{L}, \mathcal{Z}(l_{vv'}) = \mathcal{R}_{il}\}$. The walk transition probability from node v to another v' following $\mathcal{R}_{il}$ can be calculated as:

$$\mathscr{P}\left(v, v'\right) = \left(\frac{\mathcal{W}_{vv'}}{\sum_{v'' \in O_v} \mathcal{W}_{vv''}}\right)^{\rho} \tag{7}$$

Where $\rho \in$ [0,1] is a parameter to tune the impact of the out-degree in the random walk propagation process. Node v' will get lower incoming preference if node v has larger out-degree. Temporal relations can be decayed with time, which will be further discussed in section 3.2.2. For easy control of the importance of semantic and temporal factor, we introduce parameter $b_i = \frac{b_i^e}{b_i^t}$ to control the ratio of semantic entity nodes to temporal nodes. The outgoing edge set O_v can be categorized

Algorithm 1: Meta-path based Neighbourhood Context Forming

```
Input:
- Graph G, Graph schema ℂ, Meta Path 𝓜
- Parameter b_i, b_d, ρ
Output:
- context set D_𝒞

D_𝒞 ← {}    // Initialize context set
Repeat N times:
    𝒞 ← {}    // Initialize context
    k = 0    //k is the index of the sequence of 𝓜
    Randomly draw node v ∈ V_k, V_k has the node type 𝒯_k in 𝓜
    𝒞 ← 𝒞 ∪ {v}
    For k =1 to |𝓜|-1:
        Draw next node v' ∈ V_k with 𝒫(v, v') based on Equation 8
        𝒞 ← 𝒞 ∪ {v'}
    D_𝒞 ← D_𝒞 ∪ 𝒞
```

into semantic entity relations and temporal relations. Let O_v^e be the sub set of outgoing edges of semantic entity relations. O_v^t be the sub set of outgoing edges of temporal relations. $O_v = O_v^t \cup O_v^e$. By applying edge weight definition, Equation 7 can be rewritten as:

$$\mathscr{P}\left(v, v'\right) = \begin{cases} \left(\frac{1}{|O_v|}\right)^{\rho} & \text{if } v \in \mathcal{E}_i - \mathcal{E}_j^{\mathcal{T}}, v' \in \mathcal{E}_j \\ \left(\frac{b_d}{|O_v|}\right)^{\rho} & \text{if } v \in \mathcal{T}_i, v' \in \mathcal{E}_j^{\mathcal{T}} \\ \left(\frac{1}{|O_v^e| + \frac{1}{b_i} * |O_v^t|}\right)^{\rho} & \text{if } v \in \mathcal{E}_i^{\mathcal{T}}, v' \in \mathcal{E}_j \\ \left(\frac{1}{|O_v^t| + \frac{1}{b_i} * |O_v^e|}\right)^{\rho} & \text{if } v \in \mathcal{E}_i^{\mathcal{T}}, v' \in \mathcal{T}_j \end{cases} \tag{8}$$

Parameter b_i affects the importance of temporal and semantic factors respectively on measuring the similarity of two nodes with temporal annotations. If $b_i = 0$, only semantic factor will be considered. If $b_i = \infty$, only temporal factor will be considered for the value of $\mathscr{P}\left(v, v'\right)$. $\mathscr{P}\left(v, v'\right)$ can be considered a propagation function that measures how much preference of v is propagated to its succeed node v'.

Injecting weights for starting nodes A Meta-path can start with a temporal node or a semantic entity node. The weight propagates from the starting node to the target node along the meta-path. To control the influence of different types of starting nodes, we can set up a parameter for the weight of the starting node. Let $\mathcal{B}(v_i)$ denote the function to inject prior preference weight for a starting node, following a meta-path $\mathcal{M}$ =($il...kj$), the transition probability from source node v_i to target node v_j can be calculated by the following equation:

$$\mathscr{P}\left(v_i, v_j | \mathcal{M}\right) = \mathcal{B}\left(v_i\right) * \mathscr{P}\left(v_i, v_l\right) * ... * \mathscr{P}\left(v_k, v_j\right) \tag{9}$$

Let β denote a prior weight, $0 \leq \beta \leq 1$. β is a parameter used to tune the ratio of injected preferences on the entity node against the temporal node. $\beta = 0$ means no preferences are injected into the starting entity node; while $\beta = 1$ means all preferences are injected into the starting temporal node. If v_i is a semantic entity node, $v_i \in \mathcal{E}_i$, $\mathcal{B}\left(v_i\right) = \beta$; if v_i is a temporal node, $v_i \in \mathcal{T}$, $\mathcal{B}\left(v_i\right) = 1 - \beta$.

3.4. Embedding Learning Models

We can use self-supervised representation learning models to learn node embeddings. As skip-gram model has been popularly used in heterogeneous information networks for the task of node embedding learning [13], we used the skip-gram model as the embedding learning model in this paper. For a given context $\mathcal{C}$, let $\mathcal{V}_{\mathcal{C}} \subseteq \mathcal{V}$ denote the nodes in the context $\mathcal{C}$. The objective is then to maximise the average log probability: $\frac{1}{|\mathcal{V}_{\mathcal{C}}|}\sum_{v \in \mathcal{V}_{\mathcal{C}}} \sum_{v' \in \mathcal{V}_{\mathcal{C}} \setminus \{v\}} \log \Pr\left(v'|v\right)$ The input is the vector representation of a node $v \in \mathcal{V}_{\mathcal{C}}$ with random initialisation, while the output is every other node in the same context $v' \in \mathcal{V}_{\mathcal{C}} \setminus \{v\}$. Let $\mathbf{W}$ be the weight matrix between the input layer and output layer; $\mathbf{W}$ is randomly initialised and shared across all contexts. Let y denote the un-normalised log-probability for each output node v_l, which can be computed as:

$$y = \mathbf{W} \cdot \mathbf{V} \tag{10}$$

We use a multi-class classifier (e.g., softmax) to conduct the prediction task:

$$\Pr\left(v'|v\right) = \frac{e^{y_{v'}}}{\sum_{l=1}^{|\mathcal{V}_{\mathcal{C}}|} e^{y_{v_l}}} \tag{11}$$

We use the squared difference error as the loss function, stochastic gradient descent to train the model, and back propagation to obtain the gradient to update each input latent vector of node v and weight matrix $\mathbf{W}$. With this approach, we can learn the representations of each node, including source nodes, intermediate nodes, and target nodes of given paths.

4. Temporal Applications

We can apply the TNE framework to downstream temporal applications, for example temporal recommendation and similarity search. Figure 2 illustrates the application framework of TNE. In these application scenario, users and items are two basic node types in information networks $\mathcal{G}$. **Users:** $U = \{u_1, u_2, ..., u_k\}$ is the set of all users in an online community. **Items** (e.g., Products or Businesses): $P = \{p_1, p_2, ..., p_z\}$ is the set of all items that have interactions with users in U. In this section, we discuss two important temporal applications: temporal similarity search and temporal recommendations.

4.1. Temporal Similarity Search

Similarly search is popularly used in various kinds of scenarios, for example, information retrieval, clustering, and entity resolution [19]. Temporal similarity search can help to answer temporal dynamics related queries in temporal applications. Temporal similarity search is defined as given a node v and a temporal or hybrid context C_t, find a set of nodes that are similar to v. The distance or similarity measure can be calculated through various kinds of proximity computing approaches such as cosine similarity or Pearson's correlation. Cosine is used to measure the similarity of two nodes in this paper. The similarity of two nodes v and v' can be measured by the similarity of their node representations following temporally annotated

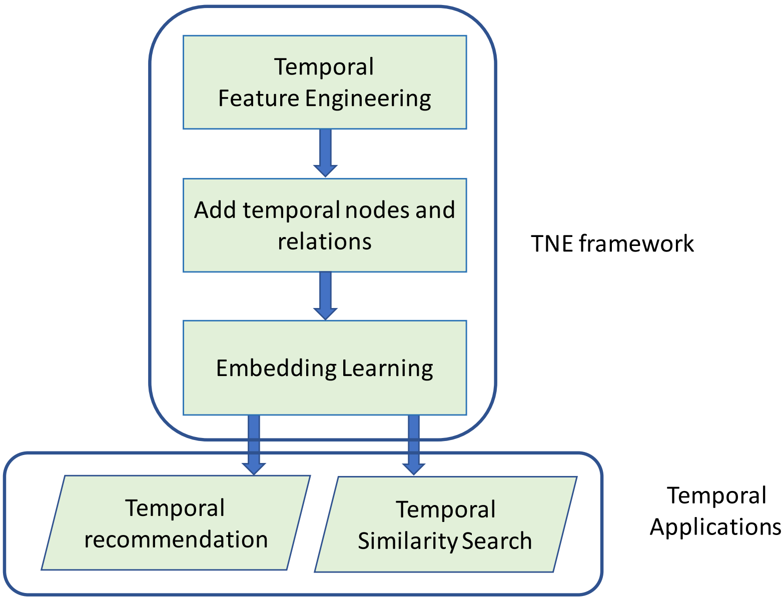


Figure 2: The Application framework of TNE

meta-paths. Let $sim(v, v'|C_t)$ be the similarity of two nodes v and v' in the context C_t. The top-k nodes of v is denoted as $\tilde{N}_v = \{v'|v' \in maxK_{v' \in V}\{sim(v, v'|C_t)\}\}, v' \in \mathcal{V}$, where $maxK\{\}$ returns the top-k most similar nodes to v. For heterogeneous entity type network, the similarity nodes is restricted to the same type of nodes with v.

4.2. Temporal Recommendation

This section discusses how to make Top-N recommendations. Let $\mathcal{L}_u(u_i, p_j)$ be the predicted score of how much the target user u_i would be interested in the item p_k, the problem of Top-N recommendation is defined as to generate a set of (i.e., N numbers of) ordered items p_l, ..., $p_m \in P - \mathcal{O}_i$ to the user u_i, where $\mathcal{O}_i$ is the set of observed items of u_i and $\mathcal{L}_u(u_i, p_l) \geq$ ... $\geq \mathcal{L}_u(u_i, p_m)$. For temporal recommendation, the task is to predict the item preferences of the target user u_i after a given time stamp t. The user behaviour data before the time stamp t is used as training while the data after the time stamp t is used for testing. With the proposed TAN framework, we can incorporate various kinds of temporal and semantic factors such as user active patterns and item life cycles, temporal decay of users and items in user and item embedding. The constructed TAN can be used for different kinds of recommendations approaches. The *neighborhood* based collaborative filtering approaches are popularly used recommendation approaches. Comparing with matrix factorisation and deep learning models, they are simple, explainable, and easy to implement. For this approach, each user or item is profiled with the learned embeddings. We calculate the similarity of the embeddings of each user with other users. The neighborhood of user u_i is denoted as $\tilde{N}_{u_i} = \{u_j|u_j \in maxK_{u_j \in U}\{sim(u_i, u_j)\}\}, u_j \in U$. For each target user u_i, the prediction score of how much u_i will be interested in an unobserved candidate item $p_j \in P - \mathcal{O}_i$ is calculated by considering the similarities between user u_i and those users who are neighbors of u_i and have rated item p_j [2]:

$$\mathcal{L}_u(u_i, p_j) = \sum_{u_k \in (\tilde{N}_{u_i} \cap L_j)} sim(u_i, u_k) \tag{12}$$

Where L_j denotes the user nodes that item node p_j has linked based on **hop**-1 relation $\mathcal{R}_{UP}{}^{-1}$ (i.e., those users that have rated item p_j). The Top N items with high prediction scores will be recommended to the target user u_i. This is a user-based approach. Let L_i denotes the item nodes that user node u_i has collected. The neighborhood of item p_j is denoted as $\tilde{N}_{p_j} = \{p_k|p_k \in maxK_{p_k \in P}\{sim(p_j, p_k)\}\}, p_k \in P$. The item based approach is given as below:

$$\mathcal{L}_p(u_i, p_j) = \sum_{p_k \in (\tilde{N}_{p_j} \cap L_i)} sim(p_j, p_k) \tag{13}$$

5. Experiments

5.1. Datasets

To evaluate the effectiveness of the proposed approaches, this work conducted experiments on the following two datasets:

- **HetRec2011-MovieLens Dataset**: This is an extension of MovieLens 10M dataset. It has enriched with various kinds of affiliated information about movies [32]. This includes the following data files or relations: the tagging assignment user-taggedmovies.dat, the movie-genres.dat, movie-directors.dat, movie-actors.dat, movie-countries.dat. The recommended items are movies.[1] The tagging behavior has timestamp information.

- **Amazon Dataset**: This dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014.[2] This dataset includes reviews (i.e., ratings, text, helpfulness votes, reviewTime), product metadata (i.e., descriptions, category information, price, brand, and image features), and links (i.e., also viewed/also bought graphs) [40]. We only selected the "Clothing, Shoes and Jewelry" category. We only retained 5-rated reviews in the dataset to ensure the users' satisfaction for learning their preferences and considered user rating as implicit feedback.

To reduce the sparseness in the dataset, this work filtered out those user nodes that have tagged less than 10 items and those item nodes that have been used by less than 10 users. We used local temporal type and global temporal type of items. For each item of HetRec2011-MovieLens Dataset, we aggregated all the tagging behaviour represented as quadruples *(User, Item, Tag, Timestamp)* related to this item. Then we convert the timestamps to Year-Month format. The global temporal type (denoted as S_g) of items are defined as Year-Month of items. If two items have the same Year-Month value, these two items are connect to the same temporal node. The local temporal type (denoted as S_l) of items are defined as Year-Month-User. If two items have been tagged at the same Year and Month by the same user, then these two items are connect to the same temporal node. The local temporal type is a personalized temporal factor. For each item of Amazon Dataset, we aggregated all the review behaviour represented as quadruples *(User, Item, Review, Timestamp)* related to this item. Then we follow the same process to get the local and global temporal type nodes and relations. The statistics of the two datasets after pre-processing and adding the temporal nodes and relations are shown in Table 2. On average, each node of HetRec2011-MovieLens has 22.82 edges, and each node of Amazon Dataset has 14.22 edges.

For all embedding methods, we use the same parameters including the vector dimension d=128 and the neighborhood size k=100. The other parameter settings include $\rho = 0.8$, $\beta = 1$. We selected some popularly used meta-paths. As symmetric meta-paths that both start from and end with item nodes can help find similar items, we used them for the temporal similarity search application. In recommendation tasks, those asymmetric meta-paths that start from a user node and end with an item node are popularly used [8, 32]. We use those popular asymmetric meta-paths for the temporal recommendation application. The selected semantic and temporally annotated meta-paths that used in the experiments are shown in Table 3. For any temporally annotated meta-paths, both S_g and S_l can be considered as the temporal types. For simplicity, we denote the temporally annotated meta-paths with S_g as $\mathcal{M}_{1,2}$, $\mathcal{M}_{2,2}$, ..., $\mathcal{M}_{8,2}$ and the temporally annotated meta-paths with S_l as $\mathcal{M}'_{1,2}$, $\mathcal{M}'_{2,2}$, ..., $\mathcal{M}'_{8,2}$. **Accuracy** and **Diversity** are used as evaluation metrics to measure the performance of the compared approaches.

5.2. *Temporal Similarity Search*

We conducted Top-N (N=[5,10]) temporal similarity search experiments. We compared the proposed embedding learning approach with the metapath2vec approach. We used the metapath2vec++ [13] approach in the experiments.[3] The embeddings learned by metapath2vec approach is denoted as **MPE** and those learned by the proposed approach denoted is denoted as **TNE**. We used $Coverage$ [26] to measure the diversity of similarity search. We randomly selected 20% of item nodes as query nodes. The results of averaged $Coverage$ value is used to measure the **Diversity** of the compared embeddings.

Table 4 shows a case study of Top-5 similar nodes of a given item node for HetRec2011-MovieLens Dataset of **MPE** and **TNE** with linear (i.e., d_l) and exponential (i.e., d_e) decay functions for both local (i.e., S_l) and global (i.e., S_g) temporal types of item nodes. We also listed the Genres of each item. We can see **MPE** mainly returns items that have the same Genres with the query item such as "Action", "Adventure", "Sci-Fi". As the proposed approach **TNE** considers both semantically and temporally associated items, the returned items is more diversified in terms of Genres such as "Comedy" and "Drama". Linear and Temporal decay functions for local and global temporal types returns different items. Table 5 shows a case study of Top-5 similar nodes of a given item node for Amazon Dataset. The tables shows that **MPE** returned items with the same categories as the query node, for example, "Team Sports" and "Athletic". The proposed approach **TNE** returned items with more diversified and associated categories such as "Pants" and "T-shirts". These items are temporally associated with the query item. The Genre/ Category $Coverage$ results of Top-10 items of the two compared approaches for different meta-paths are shown in Figure 3. We can see that the proposed approach **TNE** achieved better **Diversity** than **MPE**.

To evaluate the effectiveness of preserving both temporal and semantic information in embeddings, we conducted clustering experiments for items with semantic, temporal, hybrid labels. We selected Genre/ Category as the semantic label, Most-Recent-Year-Month as the temporal label, and Genre/ Category-Most-Recent-Year-Month as the hybrid label. For items that have multiple genres or categories, we randomly selected one genre or category to create the label. We used Normalized Mutual Information *NMI* [13] to measure the performance of clustering based on item node embeddings. The results of **TNE**

[1] `https://grouplens.org/datasets/hetrec-2011/`
[2] `http://jmcauley.ucsd.edu/data/amazon/`
[3] `https://ericdongyx.github.io/metapath2vec/m2v.html`

Table 2: The basic statistics of datasets

Nodes	Number	Relations	Number
HetRec2011-MovieLens			
Users U	152		
Movies P	301	$\mathcal{R}_{UP}$	3,870
Tags T	3,031	$\mathcal{R}_{PT}$	11,289
Movie Genres G	18	$\mathcal{R}_{PG}$	871
Directors D	193	$\mathcal{R}_{PD}$	300
Actors A	7,206	$\mathcal{R}_{PA}$	8,743
Item Global Temporal factors S_g	36	$\mathcal{R}_{PS_g}$	2,766
Item Local Temporal factors S_l	306	$\mathcal{R}_{PS_l}$	3,870
Nodes $\mathcal{V}$	11,243	Edges $\mathcal{E}$	31,709
Amazon			
Users U	12,491		
Item P	1,019	$\mathcal{R}_{UP}$	207,281
Category C	604	$\mathcal{R}_{PC}$	5,775
Brand B	77	$\mathcal{R}_{PB}$	221
Item Global Temporal factors S_g	121	$\mathcal{R}_{PS_g}$	18,725
Item Local Temporal factors S_l	16,578	$\mathcal{R}_{PS_l}$	207,281
Nodes $\mathcal{V}$	30,890	Edges $\mathcal{E}$	439,283

Table 3: The Selected Semantic and Temporally Annotated Meta-paths

Semantic Meta-path	Temporally Annotated Meta-path
HetRec2011-MovieLens	
$\mathcal{M}_{1,1} = (PGP)$	$\mathcal{M}_{1,2} = (P\{G \oplus S\}P)$
$\mathcal{M}_{2,1} = (PGPGP)$	$\mathcal{M}_{2,2} = (P\{G \oplus S\}P\{G \oplus S\}P)$
$\mathcal{M}_{3,1} = (UPUP)$	$\mathcal{M}_{3,2} = (UP\{U \oplus S\}P)$
$\mathcal{M}_{4,1} = (UPGP)$	$\mathcal{M}_{4,2} = (UP\{G \oplus S\}P)$
$\mathcal{M}_{5,1} = (UPAP)$	$\mathcal{M}_{5,2} = (UP\{A \oplus S\}P)$
$\mathcal{M}_{6,1} = (UPDP)$	$\mathcal{M}_{6,2} = (UP\{D \oplus S\}P)$
$\mathcal{M}_{7,1} = (UPTP)$	$\mathcal{M}_{7,2} = (UP\{T \oplus S\}P)$
$\mathcal{M}_{8,1} = (UPUPUP)$	$\mathcal{M}_{8,2} = (UP\{U \oplus S\}P\{U \oplus S\}P)$
$\mathcal{M}_{9,1} = (UPGPUP)$	$\mathcal{M}_{9,2} = (UP\{G \oplus S\}P\{U \oplus S\}P)$
$\mathcal{M}_{10,1} = (UPTPTP)$	$\mathcal{M}_{10,2} = (UP\{T \oplus S\}P\{T \oplus S\}P)$
Amazon	
$\mathcal{M}_{1,1} = (PCP)$	$\mathcal{M}_{1,2} = (P\{C \oplus S\}P)$
$\mathcal{M}_{2,1} = (PCPCP)$	$\mathcal{M}_{2,2} = (P\{C \oplus S\}P\{C \oplus S\}P)$
$\mathcal{M}_{3,1} = (UPUP)$	$\mathcal{M}_{3,2} = (UP\{U \oplus S\}P)$
$\mathcal{M}_{4,1} = (UPCP)$	$\mathcal{M}_{4,2} = (UP\{C \oplus S\}P)$
$\mathcal{M}_{5,1} = (UPBP)$	$\mathcal{M}_{5,2} = (UP\{B \oplus S\}P)$
$\mathcal{M}_{6,1} = (UPUPUP)$	$\mathcal{M}_{6,2} = (UP\{U \oplus S\}P\{U \oplus S\}P)$
$\mathcal{M}_{7,1} = (UPCPUP)$	$\mathcal{M}_{7,2} = (UP\{C \oplus S\}P\{U \oplus S\}P)$
$\mathcal{M}_{8,1} = (UPBPUP)$	$\mathcal{M}_{8,2} = (UP\{B \oplus S\}P\{U \oplus S\}P)$

and **MPE** are shown in Figure 4. We can see that **TNE** had higher *NMI* values than **MPE** for the selected temporal label Most-Recent-Year-Month and hybrid label Genre/ Category-Most-Recent-Year-Month, while **MPE** had better *NMI* values for semantic label Genre. This shows that **TNE** can preserve both semantic and temporal information in embeddings.

We also used t-SNE [55] to visualise the nodes based on their embeddings. The embeddings are coloured with the three kinds of selected labels different meta-paths. The results of HetRec2011-MovieLens are shown in Figure 5, while the results of Amazon Dataset are shown in Figure 6. We can see that those nodes with the same colour which represent the same semantic label Genres are close to each other for **MPE**, while the nodes with the same colours are more scattered for **TNE**. For **TNE**, those nodes with the same colour which represent the same temporal label Most-Recent-Year-Month and hybrid label Genre-Most-Recent-Year-Month are more close to each other than **MPE**. This also demonstrates the effectiveness of preserving temporal and hybrid information in embeddings for the proposed approach **TNE**.

5.3. *Temporal Recommendation*

Top-N recommendation task is popularly used for implicit or binary ratings [2]. As the task is to recommend items, we only keep User-Item $\mathcal{R}_{UP}$ relation in the Test Set. For a target user, we randomly selected 20% of User-Item relation as Test Set, the rest User-Item relation and the affiliated information forms the training graph set. The $Precision$ and $Recall$ are used to measure the **Accuracy** performance of the Top-N item recommendation task. The averaged values of all test users are used to measure the overall **Accuracy** performances of recommendation approaches. This work conducted Top-N (N = [1,5,10,50,100]) recommendation tasks on both datasets. The $Coverage$ of item Genres and $Novelty$ of Top-10 items is used to measure the **Diversity** of recommendations [26].

5.3.1. Recommendation Results

To evaluate the effectiveness of the proposed approach, we compared the performances of the following approaches.

- $\boldsymbol{MV}$: is the user-based Collaborative Filtering approach based on the **MPE** embeddings learned by metapath2vec approach [13]. This approach only considers semantic factors and did not consider temporal nodes and relations.
- $\boldsymbol{TM}$: is the proposed user based Collaborative Filtering approach based on the **TNE** embedding. This approach considers both semantic and temporal factors.
- $\boldsymbol{CF}$: is the typical user based collaborative filtering approach. It is based on the user-item relation.

The best parameter settings are selected for all the compared models. For HetRec2011-MovieLens Dataset, we selected $\mathcal{M}_{10,1}$ for MV model and $\mathcal{M}_{10,2}$ for TM model. Meanwhile, for Amazon Dataset, we selected $\mathcal{M}_{7,1}$ and $\mathcal{M}_{7,2}$ for MV and TM respectively. For both datasets, the exponential decay function (d_e) and the local temporal type (S_l) were selected. The $Precision$, $Recall$, $Coverage$, and $Novelty$ values are shown in Figure 7. We can see that for HetRec2011-MovieLens Dataset, TM performed the best in evaluation matrices $Precision$, $Coverage$, and $Novelty$. The $Recall$ values of all approaches are close, while MV had lower $Recall$ values than that of the other approaches. For Amazon Dataset, TM performed the best among these approaches from all four evaluation metrics. MV and TM performed better than CF approach from the aspect of $Coverage$ and $Novelty$ for both datasets. The results show that we can boost both **Accuracy** and **Diversity** performances, if we consider both semantic and temporal factors in recommendation approaches.

5.3.2. The effectiveness of TAN for Recommendations

Furthermore, we evaluated whether the added temporal nodes and relations are effective or not when the constructed temporally-annotated network TAN is applied to Knowledge Graph based recommender systems such as knowlege graph attention networks [57].

- $\boldsymbol{GA}$: the state-of-the-art recommendation approach based on knowledge graph attention network [57]. The input graph is an expanded graph that generated by the combination of various types of hop-n item-feature relations. In the experiments, $n = [1, 2, 3]$ [57]. The original input graph of this approach ignores temporal information.

We used the cold start datasets that filtered out those users that only had one item and those items that only being used by one user. We use GA to denote this model based on the original semantic network, GA-S_g denote the GA model with added global temporal type relations and nodes, GA-S_l denote the model with local temporal type and nodes. Note, for this set of experiments, we directly applied the constructed TANs without learning the embeddings. The temporal decay functions are not applicable for GA related models. The $Precision$, $Recall$, and $Novelty$ results of these models are shown in Figure 8. We can see that GA had slightly higher performance for $Novelty$ values than the other two approaches. GA-S_g and GA-S_l performed better than GA for both datasets from the aspects of $Precision$, $Recall$, and $Coverage$.

5.3.3. Detailed analysis of the proposed approach

We compared the performance of TM based on various kinds of meta-paths. The meta-path set for each dataset is shown in table 3. The results are shown in Figure 9. We can see that $\mathcal{M}_{10,2} = (UP\{T \oplus S\}P\{T \oplus S\}P)$ performed the best for HetRec2011-MovieLens Dataset. Among all the selected meta-paths, $\mathcal{M}_{7,2} = (UP\{C \oplus S\}P\{U \oplus S\}P)$ performed the best for Amazon dataset.

We compared the performance of TM with different settings for temporal factor weight b_p^t. b_p^t is set from 0 to 1. The semantic factor weight b_p^e can be calculated based on the formula $b_p^e = 1 - b_p^t$. $b_p^t = 0$ means that only semantic nodes will be selected in a temporally annotated meta-paths, while $b_p^t = 1$ means that only temporal nodes will be selected. The meta-path is $\mathcal{M}_{4,1}$ for both datasets. The results of different settings for b_p^t is shown in Figure 10. We can see that $b_p^t = 0$ performed the worst for both datasets. With a b_p^t value between 0.25 to 0.75, TM can achieve better results. This also demonstrated that considering both the semantic and temporal factors can promote the recommendation performances.

6. Conclusion

This work proposed TNE a temporally annotated information network embedding learning framework to preserve temporal information in embeddings. In this framework, temporal nodes and temporal relations were introduced to construct temporally annotated networks to incorporate various kinds of temporal factors in information networks. We also proposed temporal annotated meta-paths guided random walk to form hybrid context that simultaneously considers both semantic and temporal factors to find neighbour nodes. A skip-gram based self-learning approach has been used to learn the embeddings.

We conducted experiments in open accessed datasets including HetRec2011-MovieLens Dataset and Amazon Dataset. We compared the **Diversity** and **Accuracy** of the proposed approach with other static network embedding learning approaches in the tasks of temporal similarity search and recommendation. Extensive experiments show that the proposed framework is effective.

Table 4: The Case Study of Temporal Similarity Search for HetRec2011-MovieLens Dataset

Query:	**Star Wars: Episode I-The Phantom Menace**	*Genres (Action, Adventure, Sci-Fi)*	
Method	**MPE**	**TNE + linear decay function** d_l	
		(global temporal factor S_g)	(local temporal factor S_l)
Meta-path	$\mathcal{M}_{1,1} =(PGP)$	$\mathcal{M}_{1,2} =(P\{G \oplus S_g\}P)$	$\mathcal{M}'_{1,2} =(P\{G \oplus S_l\}P)$
Top N			
1	Indiana Jones and the Last Crusade (Action, Adventure)	Akira (Action, Adventure, Animation, Sci-Fi)	Mr. & Mrs. Smith (Action, Adventure, Comedy, Romance, Thriller)
2	Raiders of the Lost Ark (Action, Adventure)	The Life Aquatic with Steve Zissou (Adventure, Comedy, Fantasy)	Young Frankenstein (Action, Comedy, Horror)
3	Total Recall (Action, Adventure, Sci-Fi, Thriller)	2001: A Space Odyssey (Adventure, Sci-Fi)	Terminator 3: Rise of the Machines (Action, Adventure, Sci-Fi)
4	Indiana Jones and the Temple of Doom (Action, Adventure)	Hotaru no haka (Animation, Drama, War)	The Bourne Identity (Action, Thriller)
5	Star Wars: Episode II - Attack of the Clones (Action, Adventure, Sci-Fi)	La cité des enfants perdus (Adventure, Drama, Fantasy, Mystery, Sci-Fi)	Indiana Jones and the Last Crusade (Action, Adventure)
Method		**TNE + exponential decay function** d_e	
		(global temporal factor S_g)	(local temporal factor S_l)
Meta-path		$\mathcal{M}_{1,2} =(P\{G \oplus S_g\}P)$	$\mathcal{M}'_{1,2} =(P\{G \oplus S_l\}P)$
Top N			
1		Waiting for Guffman (Comedy)	To Kill a Mockingbird (G-Drama)
2		King Kong (Action, Adventure, Drama, Fantasy, Thriller)	Indiana Jones and the Temple of Doom (Action, Adventure)
3		I Robot (Action, Adventure, Sci-Fi, Thriller)	La cité des enfants perdus (Adventure, Drama, Fantasy, Mystery, Sci-Fi)
4		Good Night and Good Luck. (Crime, Drama, Film-Noir)	Ben-Hur (Action, Adventure, Drama, Romance)
5		Smultronstället (Drama)	Sideways (Adventure, Comedy, Drama, Romance)
Method	**MPE**	**TNE + linear decay function** d_l	
		(global temporal factor S_g)	(local temporal factor S_l)
Meta-path	$\mathcal{M}_{2,1} = (PGPGP)$	$\mathcal{M}_{2,2} = (P\{G \oplus S_g\}P\{G \oplus S_g\}P)$	$\mathcal{M}'_{2,2} = (P\{G \oplus S_l\}P\{G \oplus S_l\}P)$
Top N			
1	Dark City (Adventure, Fantasy, Film-Noir, Sci-Fi, Thriller)	Close Encounters of the Third Kind (Adventure, Drama, Sci-Fi)	The Shawshank Redemption (Drama)
2	Superman (Action, Adventure, Sci-Fi)	Raiders of the Lost Ark (Action, Adventure)	Star Wars: Episode II - Attack of the Clones (Action, Adventure, Sci-Fi)
3	Terminator 2: Judgment Day (Action, Sci-Fi)	Edward Scissorhands (Comedy, Drama, Fantasy, Romance)	Léon (Crime, Drama, Romance, Thriller)
4	Total Recall (Action, Adventure, Sci-Fi, Thriller)	Superman (Action, Adventure, Sci-Fi)	Psycho (Horror, Thriller)
5	Raiders of the Lost Ark (Action, Adventure)	The Truman Show (Drama, Fantasy)	Indiana Jones and the Temple of Doom (Action, Adventure)
Method		**TNE + exponential decay function** d_e	
		(global temporal factor S_g)	(local temporal factor S_l)
Meta-path		$\mathcal{M}_{2,2} = (P\{G \oplus S_g\}P\{G \oplus S_g\}P)$	$\mathcal{M}'_{2,2} = (P\{G \oplus S_l\}P\{G \oplus S_l\}P)$
Top N			
1		The Matrix Reloaded (Action, Sci-Fi, Thriller)	Terminator 3: Rise of the Machines (Action, Adventure, Sci-Fi)
2		Ferris Bueller's Day Off (Comedy)	Interview with the Vampire: The Vampire Chronicles (Drama, Horror)
3		This Is Spinal Tap (Comedy, Musical)	Being John Malkovich (Comedy, Drama, Fantasy)
4		The Usual Suspects (Crime, Mystery, Thriller)	Dr. Strangelove (Comedy, War)
5		Mononoke-hime (Action, Adventure, Animation, Drama, Fantasy)	Napoleon Dynamite (Comedy)

Table 5: The Case Study of Temporal Similarity Search for Amazon Dataset

Query:		*Categories (Team_Sports, Basketball, Men, Shoes, Athletic)*			
Method	**MPE**	**TNE + linear decay function d_l** (global temporal factor S_g)	**TNE + linear decay function d_l** (local temporal factor S_l)	**TNE + exponential decay function d_e** (global temporal factor S_g)	**TNE + exponential decay function d_e** (local temporal factor S_l)
Meta-path	$\mathcal{M}_{1,1} = (PCP)$	$\mathcal{M}_{1,2} = (P\{C \oplus S_g\}P)$	$\mathcal{M}'_{1,2} = (P\{C \oplus S_l\}P)$	$\mathcal{M}_{1,2} = (P\{C \oplus S_g\}P)$	$\mathcal{M}'_{1,2} = (P\{C \oplus S_l\}P)$
Top N					
1	Team_Sports, Basketball, Men, Shoes, Athletic	Uniforms_Work_Safety, Pants, Men, Clothing	Fashion_Sneakers, Men, Shoes	Fashion_Sneakers, Men, Shoes	Fashion_Sneakers, Women, Shoes
2	Team_Sports, Basketball, Men, Shoes, Athletic	No_Show_Liner_Socks, Women, Clothing	Sneakers, Girls, Shoes	Sneakers, Girls, Shoes	Women, Shoes, Athletic
3	Running, Men, Shoes, Athletic	Clothing, Uniforms_Work_Safety	Fashion_Sneakers, Women, Shoes	Shoes, Boots, Men, Chukka	Women, Imported, Shoes
4	lace_up, Running, canvas, Women, Shoes, Athletic	Jeans, Men	Fashion_Sneakers, Women, Imported, Shoes	Men, Shoes	Fashion_Sneakers, Women, Imported, Shoes
5	canvas, Women, Shoes, Athletic	canvas, Women, Shoes, Athletic	Fashion_Sneakers, Men, Shoes	Socks, Men, Clothing, Casual_Socks	Fashion_Sneakers, Women, Shoes
Meta-path	$(\mathcal{M}_{2,1} = (PCPCP)$	$\mathcal{M}_{2,2} = (P\{C \oplus S_g\}P\{C \oplus S_g\}P)$	$\mathcal{M}'_{2,2} = (P\{C \oplus S_l\}P\{C \oplus S_l\}P)$	$\mathcal{M}_{2,2} = (P\{C \oplus S_g\}P\{C \oplus S_g\}P)$	$\mathcal{M}'_{2,2} = (P\{C \oplus S_l\}P\{C \oplus S_l\}P)$
Top N					
1	Team_Sports, Basketball, Men, Shoes, Athletic	T_Shirts, Men, Shirts, Clothing	Fashion_Sneakers, Men, Shoes	Women, Clothing, Tops_Tees_Blouses	Fashion_Sneakers, Men, Shoes
2	Team_Sports, Basketball, Men, Shoes, Athletic	Fashion_Sneakers, Men, Shoes	Sneakers, Girls, Shoes	Surf_Skate_Street, Fashion_Sneakers, Men, Shoes	Fashion_Sneakers, Women, Imported, Shoes
3	Team_Sports, Basketball, Men, Shoes, Athletic	Team_Sports, Basketball, Men, Shoes, Athletic	Team_Sports, Basketball, Men, Shoes, Athletic	Sneakers, Girls, Shoes	Fashion_Sneakers, Women, Shoes
4	Team_Sports, Basketball, Men, Shoes, Athletic	Military, Uniforms_Work_Safety, Tops, Men, Clothing	Fashion_Sneakers, Men, Shoes	Surf_Skate_Street, Fashion_Sneakers, Men, Shoes	Sneakers, Girls, Shoes
5	Team_Sports, Basketball, Men, Shoes, Athletic	Accessories, Men, Belts	Fashion_Sneakers, Women, Shoes	Uniforms_Work_Safety, Pants, Men, Clothing	Fashion_Sneakers, Men, Shoes

This paper focuses on the static embedding learning approaches and did not consider the time-evolving of nodes (e.g., adding, deletion, updating of nodes). The proposed framework can be expanded to support dynamic embedding learning approaches. For example, we can record the sequence or order of nodes and edges in TNE. The future work will investigate novel dynamic time-aware embedding learning approaches based on TNE. The future work will also extend the proposed framework to include the content information of node and explore novel self-learning approaches for the proposed framework.

References

[1] Abdulla-Al-Maruf, H. Huang, and K. Kawagoe. 2012. Time Series Classification Method Based on Longest Common Subsequence and Textual Approximation. In *Seventh International Conference on Digital Information Management (ICDIM 2012)*. 130–137.

[2] Gediminas Adomavicius and Alexander Tuzhilin. [n.d.]. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. ([n. d.]).

[3] James F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Commun. ACM* 26, 11 (Nov. 1983), 832–843. https://doi.org/10.1145/182.358434

[4] Ana Paula Appel, Renato Luiz de Freitas Cunha, Charu C. Aggarwal, and Marcela Megumi Terakado. 2018. Temporally Evolving Community Detection and Prediction in Content-Centric Networks. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part II (Lecture Notes in Computer Science)*, Michele Berlingerio, Francesco Bonchi, Thomas Gärtner, Neil Hurley, and Georgiana Ifrim (Eds.), Vol. 11052. Springer, 3–18. https://doi.org/10.1007/978-3-030-10928-8_1

[5] Luca Bonomi and Xiaoqian Jiang. 2018. Patient ranking with temporally annotated data. *Journal of Biomedical Informatics* 78 (2018), 43 – 53. https://doi.org/10.1016/j.jbi.2017.12.007

[6] George Edward Pelham Box and Gwilym Jenkins. 1990. *Time Series Analysis, Forecasting and Control*. Holden-Day, Inc., USA.

[7] Hongyun Cai and Fuzhi Zhang. 2019. Detecting shilling attacks in recommender systems based on analysis of user rating behavior. *Knowledge-Based Systems* 177 (2019), 22 – 43. https://doi.org/10.1016/j.knosys.2019.04.001

[8] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C. Aggarwal, and Thomas S. Huang. [n.d.]. Heterogeneous Network Embedding via Deep Architectures. In *KDD '15*. ACM, 119–128.

[9] James Clark and Ottar Bjornstad. 2004. Population time series: Process variability, observation errors, missing values, lags, and hidden states. *Ecology* 85 (11 2004), 3140–3150. https://doi.org/10.1890/03-0520

[10] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and Diversity in Information Retrieval Evaluation *(SIGIR '08)*. Association for Computing Machinery, New York, NY, USA, 659–666. https://doi.org/10.1145/1390334.1390446

[11] Ali Daud, Juanzi Li, Lizhu Zhou, and Faqir Muhammad. 2010. Temporal Expert Finding through Generalized Time Topic Modeling. *Know.-Based Syst.* 23, 6 (Aug. 2010), 615–625. https://doi.org/10.1016/j.knosys.2010.04.008

[12] Alan Dix, Janet Finlay, and Russell Beale. 1993. Analysis of User Behaviour as Time Series *(HCI'92)*. Cambridge University Press, USA, 429–444.

[13] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *KDD'17*. ACM, 135–144.

[14] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. *Proceedings of The Web Conference 2020* (Apr 2020). https://doi.org/10.1145/3366423.3380297

[15] Hongchang Gao and Heng Huang. 2018. Deep Attributed Network Embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 3364–3370. https://doi.org/10.24963/ijcai.2018/467

[16] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. 2020. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems* 187 (2020), 104816. https://doi.org/10.1016/j.knosys.2019.06.024

[17] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. 2019. Scaling and Benchmarking Self-Supervised Visual Representation Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[18] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 855–864. https://doi.org/10.1145/2939672.2939754

[19] Jiazhen He, James Bailey, and Rui Zhang. 2014. Exploiting Transitive Similarity and Temporal Dynamics for Similarity Search in Heterogeneous Information Networks. In *Database Systems for Advanced Applications*, Sourav S. Bhowmick, Curtis E. Dyreson, Christian S. Jensen, Mong Li Lee, Agus Muliantara, and Bernhard Thalheim (Eds.). Springer International Publishing, Cham, 141–155.

[20] Neil Hurley and Mi Zhang. 2011. Novelty and Diversity in Top-N Recommendation – Analysis and Evaluation. *ACM Trans. Internet Technol.* 10, 4, Article 14 (March 2011), 30 pages. https://doi.org/10.1145/1944339.1944341

[21] Di Jin, Xiaobao Wang, Dongxiao He, Jianwu Dang, and Weixiong Zhang. 2021. Robust Detection of Link Communities With Summary Description in Social Networks. *IEEE Transactions on Knowledge and Data Engineering* 33, 6 (2021), 2737–2749. https://doi.org/10.1109/TKDE.2019.2958806

[22] Rohit J. Kate. 2016. Using Dynamic Time Warping Distances as Features for Improved Time Series Classification. *Data Min. Knowl. Discov.* 30, 2 (March 2016), 283–312. https://doi.org/10.1007/s10618-015-0418-x

[23] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, I. Kobyzev, Akshay Sethi, Peter Forsyth, P. Poupart, and K. Borgwardt. 2020. Representation Learning for Dynamic Graphs: A Survey. *J. Mach. Learn. Res.* 21 (2020), 70:1–70:73.

[24] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR* abs/1609.02907 (2016). arXiv:1609.02907 http://arxiv.org/abs/1609.02907

[25] Yehuda Koren. 2009. Collaborative Filtering with Temporal Dynamics *(KDD'09)*. Association for Computing Machinery, New York, NY, USA, 447–456. https://doi.org/10.1145/1557019.1557072

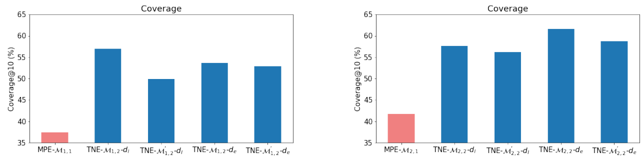
[26] Matev Kunaver and Toma Porl. 2017. Diversity in Recommender Systems A Survey. *Know.-Based Syst.* 123, C (May 2017), 154–162. https://doi.org/10.1016/j.knosys.2017.02.009

[27] Raid Lafta, Ji Zhang, Xiaohui Tao, Jerry Chun-Wei Lin, Fulong Chen, Yonglong Luo, and Xiaoyao Zheng. 2018. A Recommender System with Advanced Time Series Medical Data Analysis for Diabetes Patients in a Telehealth Environment. In *Database and Expert Systems Applications*, Sven Hartmann, Hui Ma, Abdelkader Hameurlain, Günther Pernul, and Roland R. Wagner (Eds.). Springer International Publishing, Cham, 185–192.

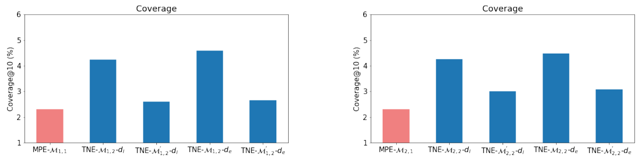
[28] Janette Lehmann, Mounia Lalmas, Elad Yom-Tov, and Georges Dupret. 2012. Models of User Engagement. In *User Modeling, Adaptation, and Personalization*, Judith Masthoff, Bamshad Mobasher, Michel C. Desmarais, and Roger Nkambou (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 164–175.

[29] Chaozhuo Li, Senzhang Wang, Dejian Yang, Zhoujun Li, Yang Yang, Xiaoming Zhang, and Jianshe Zhou. 2017. PPNE: Property Preserving Network Embedding. In *Database Systems for Advanced Applications*, Selçuk Candan, Lei Chen, Torben Bach Pedersen, Lijun Chang, and Wen Hua (Eds.). Springer International Publishing, Cham, 163–179.

[30] Dong Li, Shengping Zhang, Xin Sun, Huiyu Zhou, Sheng Li, and Xuelong Li. 2017. Modeling Information Diffusion over Social Networks for Temporal Dynamic Prediction. *IEEE Transactions on Knowledge and Data Engineering* 29, 9 (1 Sept. 2017), 1985–1997. https://doi.org/10.1109/TKDE.2017.2702162

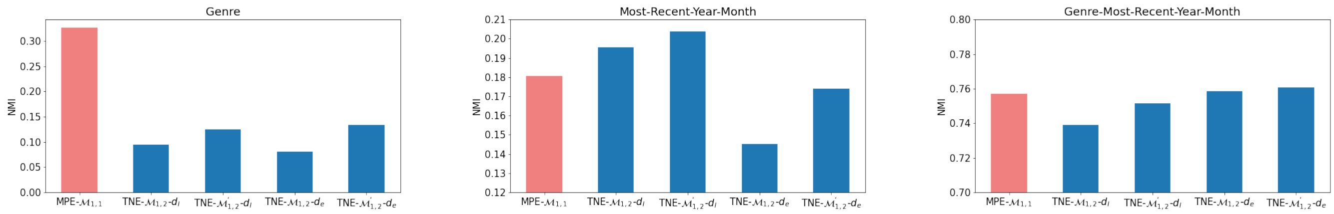


(a) HetRec2011-MovieLens Dataset

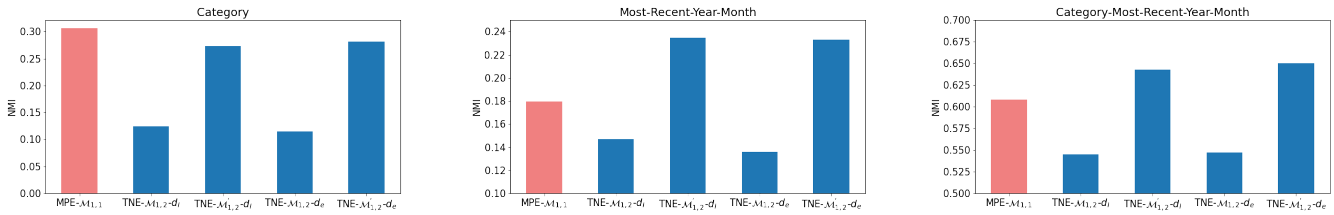


(b) Amazon Dataset

Figure 3: Top-10 Genre/ Category $Coverage$ of MPE and TNE with different meta-paths



(a) HetRec2011-MovieLens Dataset



(b) Amazon Dataset

Figure 4: Results of *NMI* of MPE and TNE with different labels

[31] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 322–330. https://doi.org/10.1145/3336191.3371786

[32] H. Liang. 2020. DRprofiling: Deep Reinforcement User Profiling for Recommendations in Heterogenous Information Networks. *IEEE Transactions on Knowledge and Data Engineering* (2020), 1–1.

[33] Huizhi Liang, Umarani Ganeshbabu, and Thomas Thorne. [n.d.]. A dynamic Bayesian network approach for analysing topic-sentiment evolution. 8 ([n. d.]), 54164–54174. https://doi.org/10.1109/ACCESS.2020.2979012

[34] Huizhi Liang, Yue Xu, Dian Tjondronegoro, and Peter Christen. 2012. Time-aware Topic Recommendation Based on Micro-blogs. In *CIKM '12*. 1657–1661.

[35] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI'15*. AAAI Press, 2181–2187.

[36] Y. Ling, W. Zhao, W. Yang, and F. Cai. 2018. TMP: Meta-path based Recommendation on Time-Weighted Heterogeneous Information Networks. In *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*. 679–683.

[37] Yuanfu Lu, Xiao Wang, Chuan Shi, Philip S. Yu, and Yanfang Ye. 2019. Temporal Network Embedding with Micro- and Macro-Dynamics *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 469–478. https://doi.org/10.1145/3357384.3357943

[38] Cheng Luo, Xiongcai Cai, and Nipa Chowdhury. 2014. Self-training Temporal Dynamic Collaborative Filtering. In *Advances in Knowledge Discovery and Data Mining*, Vincent S. Tseng, Tu Bao Ho, Zhi-Hua Zhou, Arbee L. P. Chen, and Hung-Yu Kao (Eds.). Springer International Publishing, Cham, 461–472.

[39] Yao Ma, Ziyi Guo, Zhaocun Ren, Jiliang Tang, and Dawei Yin. 2020. *Streaming Graph Neural Networks*. Association for Computing Machinery, New York, NY, USA, 719–728.

[40] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 43–52. https://doi.org/10.1145/2766462.2767755

[41] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *CoRR* abs/1310.4546 (2013). arXiv:1310.4546 http://arxiv.org/abs/1310.4546

[42] Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen K. Ahmed, Eunyee Koh, and Sungchul Kim. 2018. Continuous-Time Dynamic Network Embeddings *(WWW'18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 969–976. https://doi.org/10.1145/3184558.3191526

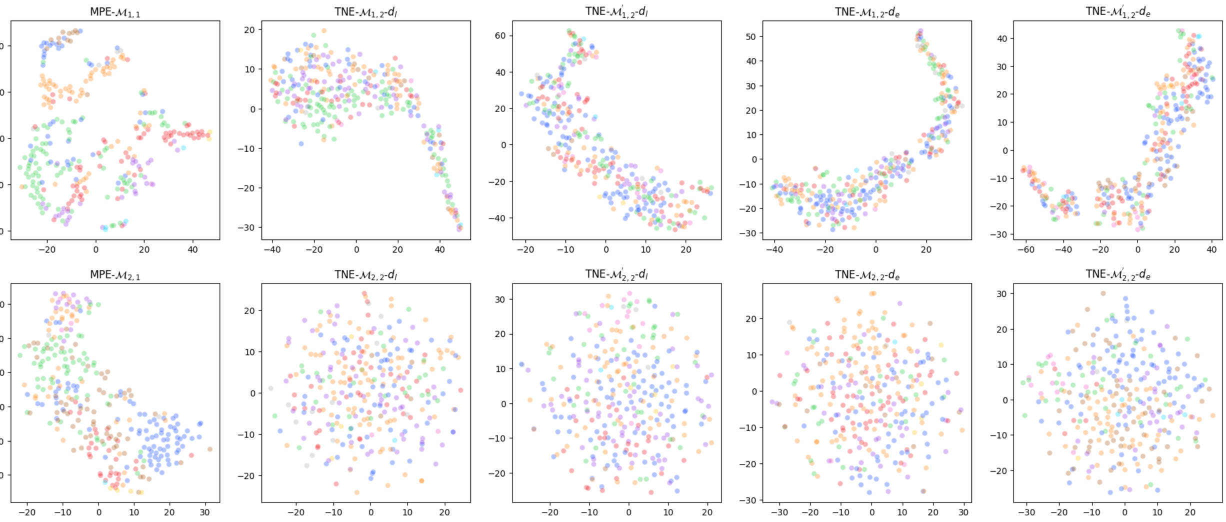
[43] Girish Palshikar. 2009. Simple Algorithms for Peak Detection in Time-Series. (01 2009).

[44] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. 2020. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

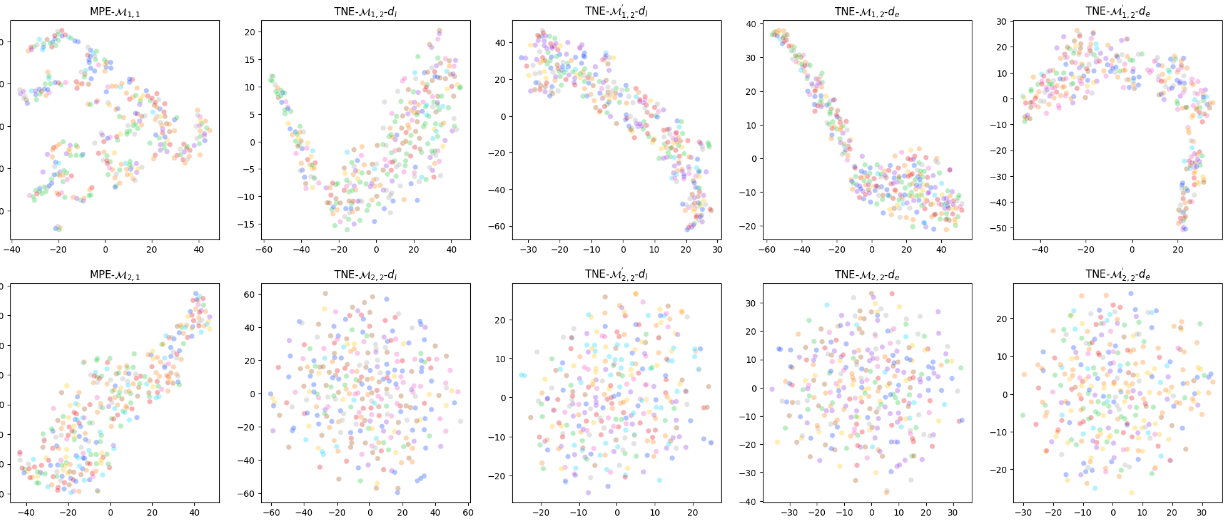
[45] Tiago P. Peixoto and Martin Rosvall. 2017. Modelling sequences and temporal networks with dynamic community structures. *Nature Communications* 8, 1 (2017), 582. https://doi.org/10.1038/s41467-017-00148-9

[46] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations *(KDD '14)*. Association for Computing Machinery, New York, NY, USA, 701–710. https://doi.org/10.1145/2623330.2623732

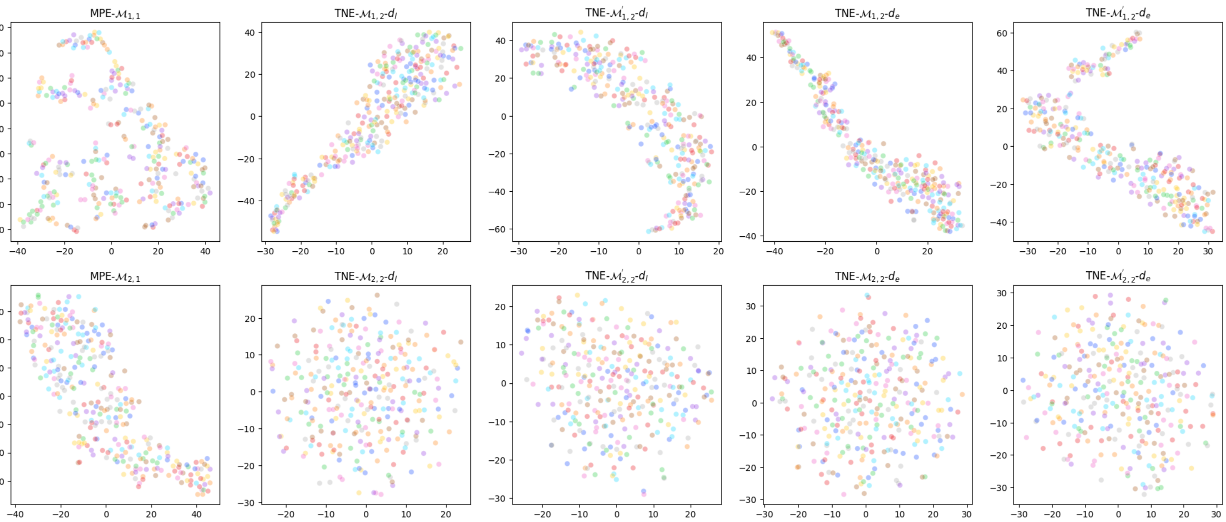
[47] M. Prasad, N. Snyderman, J. Verbeke, and R. Wurtz. 2013. Time Interval



(a) Genre Labels

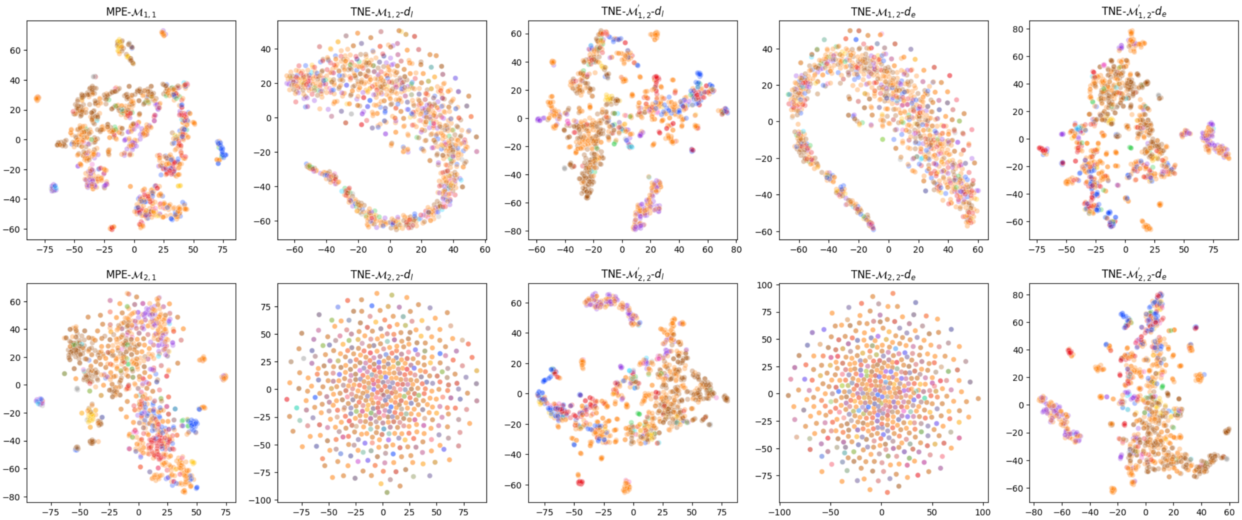


(b) Most-Recent-Year-Month Labels

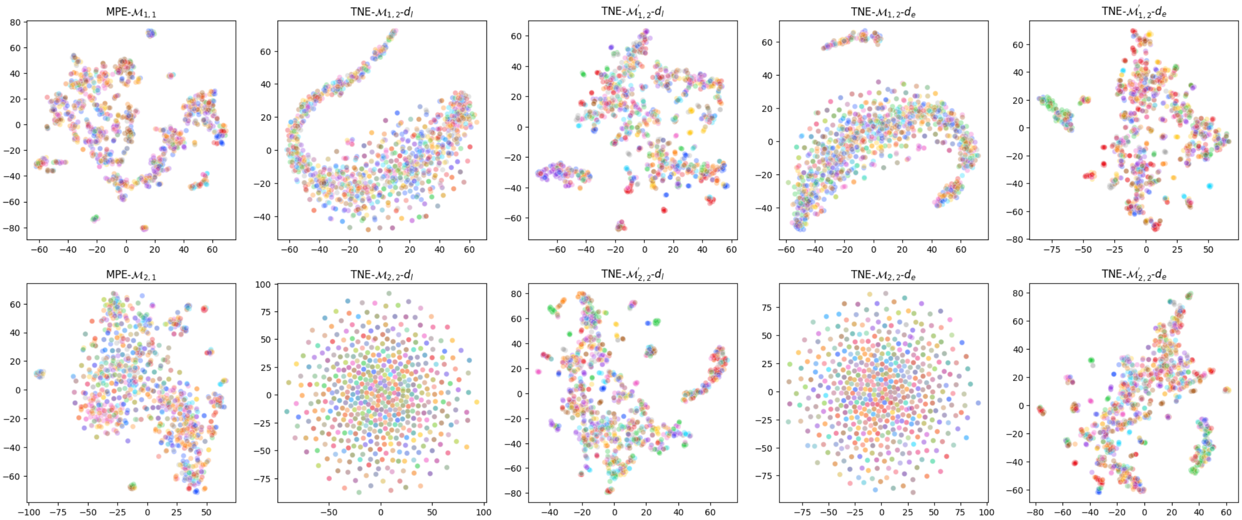


(c) Genre-Most-Recent-Year-Month Labels

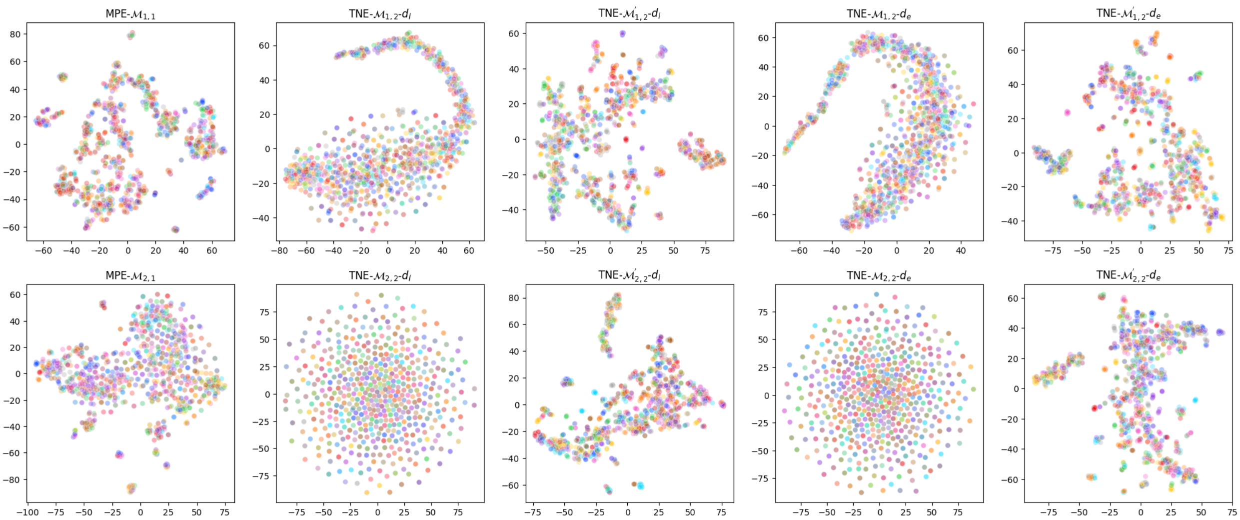
Figure 5: TSNE visualisation of item embeddings with different labels for HetRec2011-MovieLens Dataset.Coloured with Genres, Recent-Year-Month, and Genre-Most-Recent-Year-Month Labels respectively



(a) Category Labels

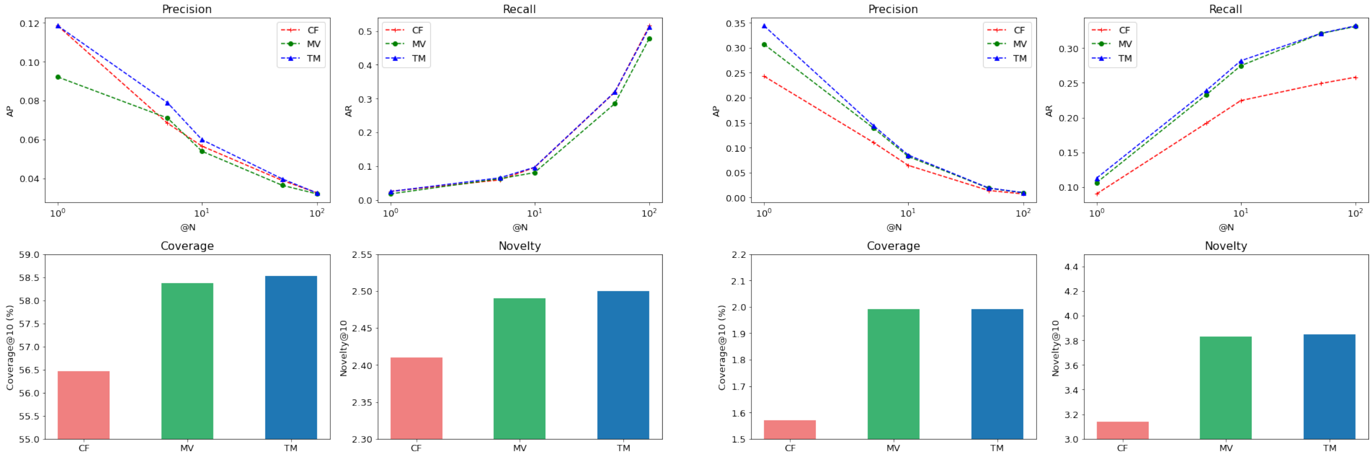


(b) Most-Recent-Year-Month Labels



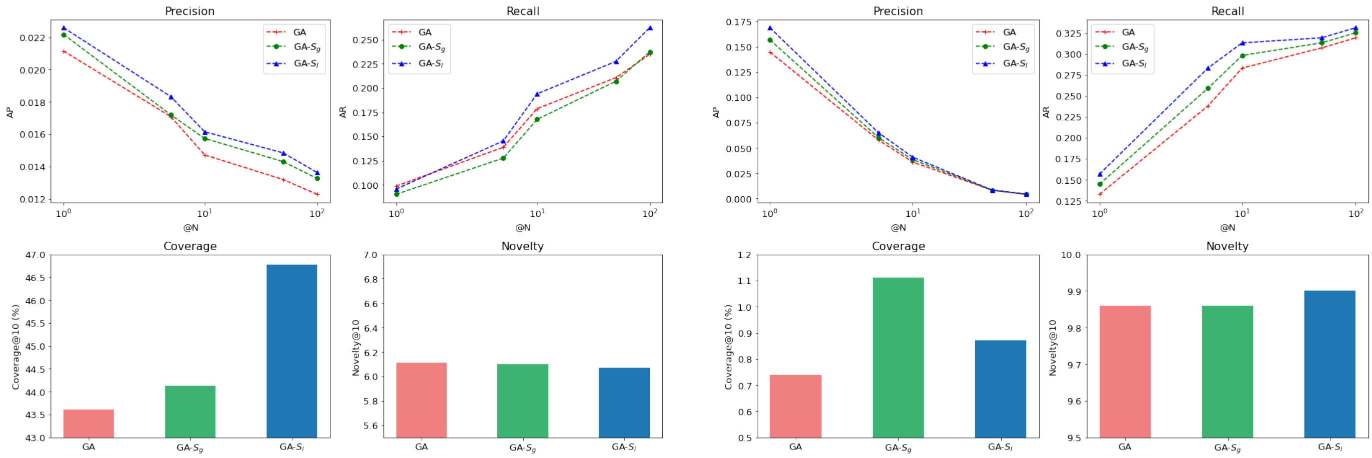
(c) Category-Most-Recent-Year-Month Labels

Figure 6: TSNE visualisation of item embeddings with different labels for Amazon Dataset. Coloured with Genres, Most-Recent-Year-Month, and Genre-Most-Recent-Year-Month Labels respectively



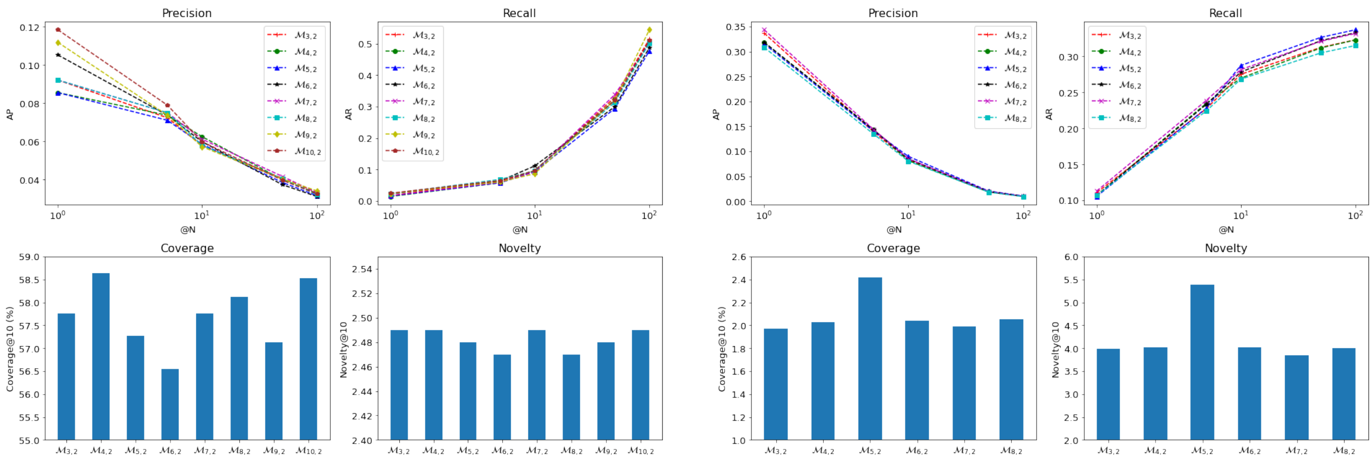
(a) HetRec2011-MovieLens Dataset (b) Amazon Dataset

Figure 7: Results of Comparison with baseline models



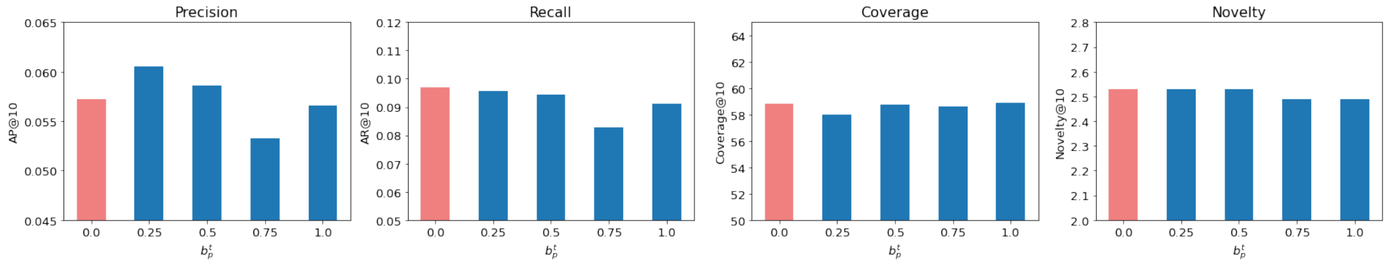
(a) HetRec2011-MovieLens Dataset (b) Amazon Dataset

Figure 8: Results of GA model based on different TANs for cold start datasets

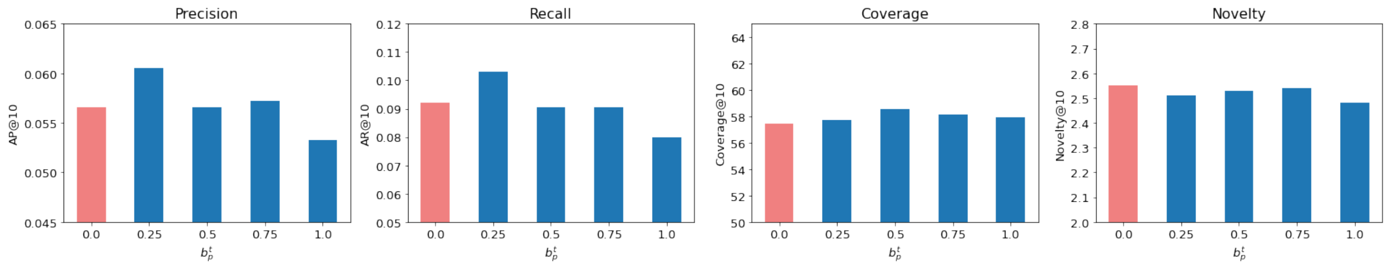


(a) HetRec2011-MovieLens Dataset (b) Amazon Dataset

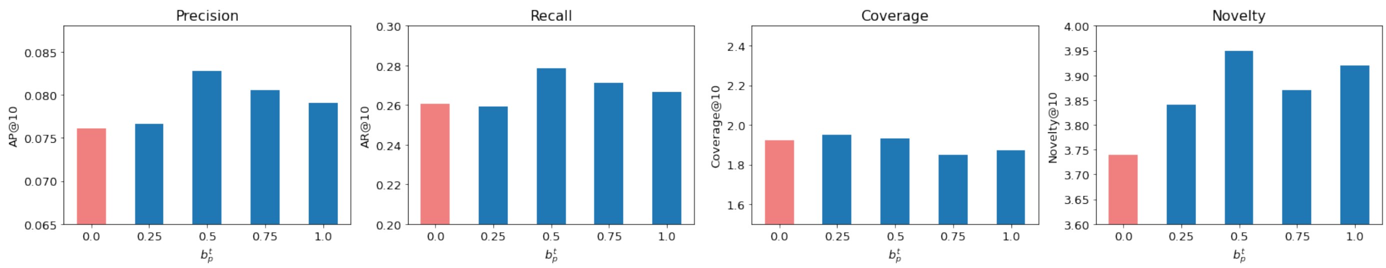
Figure 9: Results of TM model based on different meta-paths



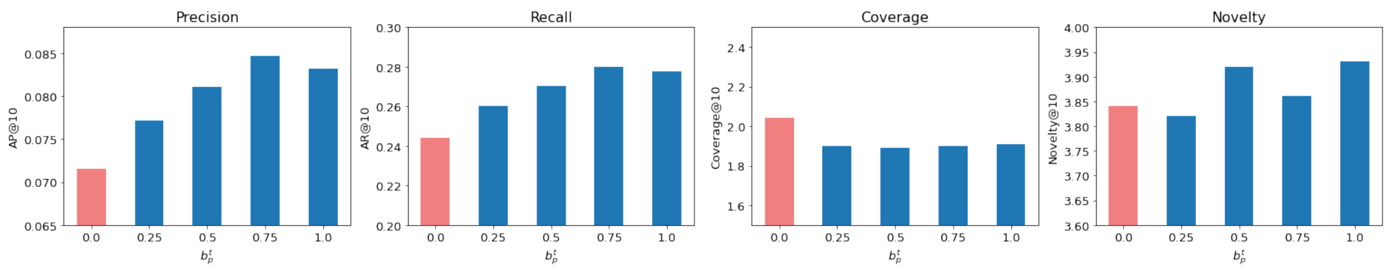
(a) Linear Decay d_g with different b_p^t weights for HetRec2011-MovieLens Dataset



(b) Exponential Decay d_e with different b_p^t weights for HetRec2011-MovieLens Dataset



(c) Linear Decay d_g with different b_p^t weights for Amazon Dataset



(d) Exponential Decay d_e with different b_p^t weights for Amazon Dataset

Figure 10: Results of different settings for temporal weight parameter b_p^t of TM model based on meta-path $\mathcal{M}_{4,1}$ with different temporal decay functions

Distributions and the Rossi Correlation Function. *Nuclear Science and Engineering* 174, 1 (2013), 1–29. https://doi.org/10.13182/NSE11-87 arXiv:https://doi.org/10.13182/NSE11-87

[48] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2020. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 519–527. https://doi.org/10.1145/3336191.3371845

[49] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. 2017. Heterogeneous Information Network Embedding for Recommendation. *CoRR* abs/1711.10730 (2017).

[50] Uriel Singer, Ido Guy, and Kira Radinsky. 2019. Node Embedding over Temporal Graphs. *CoRR* abs/1903.08889 (2019). arXiv:1903.08889 http://arxiv.org/abs/1903.08889

[51] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *PVLDB* 4, 11 (2011), 992–1003.

[52] Maddalena Torricelli, Márton Karsai, and Laetitia Gauvin. 2020. weg2vec: Event embedding for temporal networks. *Scientific Reports* 10, 1 (2020), 7164. https://doi.org/10.1038/s41598-020-63221-2

[53] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-Evolve: Deep Reasoning in Temporal Knowledge Graphs. *CoRR* abs/1705.05742 (2017). arXiv:1705.05742 http://arxiv.org/abs/1705.05742

[54] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2018. Representation Learning over Dynamic Graphs. *CoRR* abs/1803.04051 (2018). arXiv:1803.04051 http://arxiv.org/abs/1803.04051

[55] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.

html

[56] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations* (2018). https://openreview.net/forum?id=rJXMpikCZ

[57] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *KDD*.

[58] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI '14*. AAAI Press, 1112–1119.

[59] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. 2010. Temporal Recommendation on Graphs via Long- and Short-Term Preference Fusion *(KDD '10)*. Association for Computing Machinery, New York, NY, USA, 723–732. https://doi.org/10.1145/1835804.1835896

[60] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff G. Schneider, and Jaime G. Carbonell. 2010. Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization.. In *SDM*. SIAM, 211–222. http://dblp.uni-trier.de/db/conf/sdm/sdm2010.html#XiongCHSC10

[61] yu yf, Yuelong Zhu, Shijin Li, and Dingsheng Wan. 2014. Time Series Outlier Detection Based on Sliding Window Prediction. *Mathematical Problems in Engineering* 2014 (10 2014). https://doi.org/10.1155/2014/879736

[62] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. 2016. Temporal Regularized Matrix Factorization for High-dimensional Time Series Prediction. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 847–855.

[63] Zheng Yu, Xuhui Fan, Marcin Pietrasik, and Marek Z. Reformat. 2020. Fragmentation Coagulation Based Mixed Membership Stochastic Blockmodel. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 04 (Apr. 2020), 6704–6711. https://doi.org/10.1609/aaai.v34i04.6148

[64] Yongfeng Zhang, Min Zhang, Yi Zhang, Guokun Lai, Yiqun Liu, Honghui Zhang, and Shaoping Ma. 2015. Daily-Aware Personalized Recommendation Based on Feature-Level Time Series Analysis *(WWW '15)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1373–1383. https://doi.org/10.1145/2736277.2741087

[65] Y. Zhou, W. Liu, Y. Pei, L. Wang, D. Zha, and T. Fu. 2019. Dynamic Network Embedding by Semantic Evolution. In *2019 International Joint Conference on Neural Networks (IJCNN)*. 1–8.

[66] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. 2018. Embedding Temporal Network via Neighborhood Formation *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 2857–2866. https://doi.org/10.1145/3219819.3220054

[67] A. Özcan and Ş. G. Öğüdücü. 2015. Multivariate temporal Link Prediction in evolving social networks. In *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*. 185–190.