HAL open science

DyANE: Dynamics-aware node embedding for temporal networks

Koya Sato, Mizuki Oka, Alain Barrat, Ciro Cattuto

To cite this version:

Koya Sato, Mizuki Oka, Alain Barrat, Ciro Cattuto. DyANE: Dynamics-aware node embedding for temporal networks. 2019. hal-02289376

HAL Id: hal-02289376

https://hal.science/hal-02289376

Preprint submitted on 16 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

arXiv:1909.05976v1 [physics.soc-ph] 12 Sep 2019

DyANE: Dynamics-aware node embedding for temporal networks

Koya Sato
University of Tsukuba,
Tsukuba, Japan
koya@osss.cs.tsukuba.ac.jp

Mizuki Oka
University of Tsukuba,
Tsukuba, Japan
mizuki@cs.tsukuba.ac.jp

Alain Barrat
Aix Marseille Univ,
Université de Toulon,
CNRS, CPT, Marseille, France
ISI Foundation, Turin, Italy
Alain.Barrat@cpt.univ-mrs.fr

Ciro Cattuto
ISI Foundation, Turin, Italy
ciro.cattuto@isi.it

Abstract

Low-dimensional vector representations of network nodes have proven successful to feed graph data to machine learning algorithms and to improve performance across diverse tasks. Most of the embedding techniques, however, have been developed with the goal of achieving dense, low-dimensional encoding of network structure and patterns. Here, we present a node embedding technique aimed at providing low-dimensional feature vectors that are informative of dynamical processes occurring over temporal networks – rather than of the network structure itself - with the goal of enabling prediction tasks related to the evolution and outcome of these processes. We achieve this by using a modified supra-adjacency representation of temporal networks and building on standard embedding techniques for static graphs based on random-walks. We show that the resulting embedding vectors are useful for prediction tasks related to paradigmatic dynamical processes, namely epidemic spreading over empirical temporal networks. In particular, we illustrate the performance of our approach for the prediction of nodes' epidemic states in a single instance of the spreading process. We show how framing this task as a supervised multi-label classification task on the embedding vectors allows us to estimate the temporal evolution of the entire system from a partial sampling of nodes at random times, with potential impact for nowcasting infectious disease dynamics.

Introduction

The ubiquity of network representations of widely different systems has led to a flourishing of methods aimed at the analysis of their structure. Among those, network node embedding methods has recently gained a lot of popularity (Cai, Zheng, and Chang 2018; Goyal, Chhetri, and Canedo 2019). Node embedding maps each node of a network into a low-dimensional vector, such that the vectors representing different nodes are close if the network nodes share some similarity or are close in the network. Node embedding thus aims at exposing in the low-dimensional space structural features and relevant patterns of the network that are not necessarily evident in the network representation. Most importantly, the embedding vectors can be used as feature vectors in machine learning applications, and have been shown to yield improved performance for tasks such as node classification, link prediction, clustering, or visualization.

While node embeddings have proven successful in achieving low-dimensional encoding of network structures, networks are also the support of important dynamical processes, such as epidemic or rumor spreading, cascading failures, consensus formation, etc. (Barrat, Barthélemy, and Vespignani 2008). Here we introduce and experiment with node embedding methods tailored to the study of *dynamical processes on temporal networks*, and in particular to the task of predicting the evolution and outcome of one instance of the dynamics (e.g., an epidemic spread) from partial information and without detailed knowledge of the dynamical process itself. A useful embedding should thus yield low-dimensional vectors that encode information relevant to the *dynamics* of the process occurring over a temporal network – rather than information about the network *structure* itself. Since dynamical processes unfold over time-respecting paths determined by the underlying network and by its evolution over time, we argue that the sought embeddings should be informative of these paths – the paths along which information can propagate. Driven by this idea, we propose to map the temporal network to a static graph representation, a so-called supra-adjacency representation, whose nodes are `(node, time)` pairs of the original temporal network (Valdano et al. 2015). We modify the original supra-adjacency representation method to only consider nodes at those times when they interact, and we map the original temporal edges to edges between the corresponding `(node,time)` pairs, so that the static graph representation preserves the temporal paths of the original temporal network (i.e., the paths supporting and constraining the dynamical process at hand). An example of the supra-adjacency representation we use here is shown in Fig. 1. Since the resulting representation is a static graph, we can then apply standard embedding techniques: we focus on embeddings based on random walks as they provide an efficient way to sample the relevant paths.

We study the usefulness of the proposed embeddings in the context of a paradigmatic dynamical process – epidemic spread over temporal networks – in which network nodes exist in few discrete states and the dynamics consists of transitions between such states (e.g., a "susceptible" node becoming "infectious"). We focus on the task of predicting the nodes' states over time for a single realization of the epidemic process. Specifically, we set up a multi-label super-

vised classification problem with a training set obtained by sampling the node states at random times, with no information about the mechanics of state transitions nor on the parameters of the epidemic process. Our contributions are as follows:

- We propose a new method for node embedding tailored to the study of dynamical process on temporal networks, using a modified supra-adjacency representation for temporal networks and building on standard random-walk based embeddings for static graphs.
- We show that in the important case of epidemic spreading, a satisfactory prediction performance of nodes' states can be achieved in a supervised multi-label classification setting informed by the proposed embeddings.
- We show that our method achieves good performance in estimating the temporal evolution of the entire system from sparse observations, consistently across several data sets and across a broad range of parameters of the epidemic model. Our approach requires no fine-tuning of the embedding hyper-parameters and yields consistently superior performance than other embedding methods.

Problem Formulation

Temporal Network

We consider a temporal network g in discrete time on the set of timestamps $T = (1, 2, \cdots, |T|)$: g is defined as the set E of the undirected temporal edges (v_i, v_j, t), meaning that nodes v_i and v_j are linked at $t \in T$, possibly with a weight w (for simplicity we consider only positive weights). At each timestamp t, E_t denotes the set of temporal edges at t, and V_t is the set of nodes which have at least one temporal edge at t: $V_t = \{v_i | \exists v_j, (v_i, v_j, t) \in E_t\}$. We define the snapshot network at t as the undirected weighted network $G_t = (V_t, E_t)$, and the temporal network g can be seen as the succession of snapshot networks $(G_1, \cdots, G_{|T|})$. The overall set of nodes is $V = \cup_{t \in T} V_t$.

For each node $v_i \in V$, we define its set of active times T_{v_i} as the set of timestamps t in which it has at least one temporal edge, i.e., such that $v_i \in V_t$: $T_{v_i} = \{t | \exists v_j, (v_i, v_j, t) \in E_t\}$. We denote the a-th active time of v_i by $t_{v_i,a} \in T_{v_i}$, with $t_{v_i,a} < t_{v_i,a+1}$. We then define the set of active copies of each node v_i, that we call "active nodes", as $\mathcal{V}_{v_i} = \{(v_i, t) | t \in T_{v_i}\}$. The overall set of active nodes is $\mathcal{V} = \cup_{v_i \in V} \mathcal{V}_{v_i}$.

Dynamical process

We consider a dynamical process taking place on the temporal network, such that the nodes $v_i \in V$ can be in one of a finite set of discrete states S. Nodes can change state either spontaneously or through interaction along temporal edges. Our definition is thus very general and encompasses in particular models of epidemic propagation, rumor propagation, opinion formation or cascading processes (Barrat, Barthélemy, and Vespignani 2008; Castellano, Fortunato, and Loreto 2009; Pastor-Satorras et al. 2015).

The mapping $f : (v_i, t) \in \mathcal{V} \rightarrow s \in S$ specifies the state of each node at each of its active times. We assume that a sample of these states is known: we define the set of the corresponding observed active nodes as $D \subset \mathcal{V}$. Here, for simplicity, we will assume that D results from a uniform random sampling of $\mathcal{V}$. We also assume that the state of a node can be only be observed when it is active, i.e., in contact with at least another node.

For clarity, here we will focus on a paradigmatic dynamical process taking place on the temporal network, the Susceptible-Infectious-Recovered (SIR) model for epidemic spreading, which has been widely used to model contagious infections such as flu-like diseases (Keeling and Rohani 2008). In this model, each node can be at each time in one of three possible states: susceptible (S), infectious (I), and recovered (R). At the start of the process, all nodes are in state S, except for the epidemic seeds, which are in state I. A contact between an S and an I nodes leads to a contagion event in which the S node becomes infectious with probability β per unit time (recall we work in discrete time). Let us denote by I_t the set of infectious nodes at t, and consider a susceptible node v_i. We denote its set of neighbours at t as $N_t(v_i) = \{v_j | (v_i, v_j, t) \in E_t\}$, and $N_t(v_i) \cap I_t$ is the set of its infectious neighbours at t. The probability that none of these infectious neighbours transmits the disease to v_i at timestep t is $(1-\beta)^{|N_t(v_i) \cap I_t|}$, and thus the probability that v_i becomes infectious at time t, due to its interactions, is $1-(1-\beta)^{|N_t(v_i) \cap I_t|}$. Recovery from state I to state R occurs also stochastically: each infectious node becomes recovered (R) at each timestamp with probability μ. Recovered nodes do not change state any more. The parameters of the model are thus the infection and recovery rates β and μ (Keeling and Rohani 2008).

Problem statement

Given a known temporal network and a partial observation of the dynamical states of the nodes, the problem consists in predicting the dynamical state of all nodes at all their active times. In other words, knowing the state of the subset D of observed nodes at some active times, we want to predict the state of all the active nodes at all times. Crucially, we seek to achieve this prediction without any detailed information on the dynamical process at hand, except for the set of possible states of each node. In particular, we do not make assumptions on the allowed state transitions, the parameters governing the dynamical process, nor even the reversibility or irreversibility of the process. We also remark that the above problem statement implies that we will be working on *single realizations* of the dynamical process, with the goal of predicting the state of a given node at a given time, rather than predicting statistical properties averaged over a sample of simulated or observed dynamics.

Notice that the SIR model – in addition to its relevance to many real-world phenomena – is particularly interesting to study in this context: it features both state transitions occurring upon interaction (hence, along the edges of the temporal network) as well as spontaneous state transitions that can occur at any time, and in particular between successive active times of a node (the infectious-recovered transition).

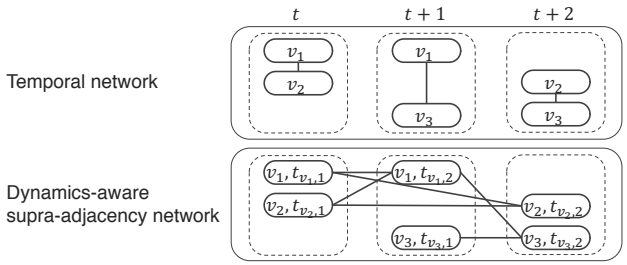


Figure 1: Proposed supra-adjacency representation (*dyn-supra*). The original temporal network (top), whose state is shown at three different times, is mapped to a static representation (bottom) where nodes are `(node,time)` pairs of the original network.

Our Approach: DyANE

Our approach consists of three steps. First, we map the temporal network to a static network between active nodes through a modified supra-adjacency representation. Second, we apply standard embedding techniques for static graphs to this supra-adjacency network. We will consider embeddings based on random walks as they explore the temporal paths on which transmission between nodes can occur. Finally, we train a classifier to predict the dynamical state of all active nodes based on the vector representation of active nodes and the partially observed states.

Supra-adjacency representation

We first map the temporal network to a supra-adjacency representation, i.e., to a new static network whose nodes are the active nodes of the temporal network. We thus define the supra-adjacency network as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ are (weighted, undirected) edges joining active nodes. The mapping from the temporal network to the supra-adjacency network consists of the following two procedures (Fig. 1):

- For each node v_i, we connect its successive active times: for each active time $t_{v_i,a}$ of v_i, we draw a "self-coupling" edge between $(v_i, t_{v_i,a})$ and $(v_i, t_{v_i,a+1})$ (recall that active times are ordered in increasing temporal order).
- For each temporal edge (v_i, v_j, t), the time t corresponds by definition to an active time for both v_i and v_j, that we denote respectively by $t_{v_i,a}$ and $t_{v_j,b}$. We then map $(v_i, v_j, t) \in E$ to two undirected edges $\in \mathcal{E}$, namely $\big((v_i, t_{v_i,a}), (v_j, t_{v_j,b+1})\big)$ and $\big((v_j, t_{v_j,b}), (v_i, t_{v_i,a+1})\big)$. In other words, the active copy of v_i at t, (v_i, t), is linked to the next active copy of v_j, and vice-versa.

The first procedure makes each active node adjacent to its adjacent past and future versions (active times), which ensures that a node carrying an information at a certain time can propagate it to its future self along the self-coupling edges, and is useful in an embedding perspective to favor temporal continuity. The second procedure encodes the temporal interactions, and yields the crucial property that any time-respecting path existing on the original temporal network, on which a dynamical process can occur, is also represented in the supra-adjacency representation. Indeed, if an interaction between two nodes v_i and v_j occurs at time t and potentially modifies their states, e.g., by contagion or opinion exchange or modification, this can be observed and will have consequences only at their next respective active times: for instance, if v_i transmits a disease to v_j at t, v_j can propagate it further to other neighbours only at its next active time, and not immediately at t. This is reflected in the supra-adjacency representation we propose.

The edges in $\mathcal{E}$ are thus of two types, joining two active nodes corresponding either to the same original node, or to distinct ones. For each type, we can consider various ways of assigning weights to the edge. We first consider for simplicity that all self-coupling edges carry the same weight ω, which becomes thus a hyperparameter of the procedure. Moreover, we simply report the weight of each original temporal edge (v_i, v_j, t) on the two supra-adjacency edges $\big((v_i, t_{v_i,a}), (v_j, t_{v_j,b+1})\big)$ and $\big((v_j, t_{v_j,b}), (v_i, t_{v_i,a+1})\big)$ (with $t = t_{v_i,a} = t_{v_j,b}$).

In the following, we will refer to the above supra-adjacency representation as *dyn-supra*. We will moreover consider two variations of this representation. First, we can encode the direction of time of the original temporal network in the supra-adjacency representation by making all links of $\mathcal{E}$ directed: an edge $((v_i, t), (v_j, t')) \in \mathcal{E}$ is then oriented according to the direction of increasing time, i.e., pointing from the active node with the earlier time $\min(t, t')$ to the one with the later time $\max(t, t')$. We will refer to this representation as *dyn-supra-directed*.

Another possible variation consists in encoding the time delay between active nodes into edge weights, with decreasing weights for increasing temporal differences. This decay of edge weights is consistent with the idea that successive active nodes that are temporally far apart are less likely to influence one another (which is the case for many important dynamical processes). In our case, we will consider that the original weight of the edge $\big((v_i, t_{v_i,a}), (v_j, t_{v_j,b})\big)$ in the *dyn-supra* representation (i.e., ω if $v_i = v_j$, or the original weight w of the temporal edge if $v_i \neq v_j$) is multiplied by the reciprocal of the time difference between the active nodes if $t_{v_i,a} \neq t_{v_j,b}$, i.e., $\big|1/(t_{v_i,a} - t_{v_j,b})\big|$. We will refer to this representation as *dyn-supra-decay*.

Node embedding

The central idea of the node embedding method we propose for temporal networks, which we call *DyANE* (Dynamics-Aware Node Embeddings), is to apply to the supra-adjacency network $\mathcal{G}$ any of the node embedding methods that have been developed for static networks. In particular, here we will use DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), as it is a simple and paradigmatic algorithm, and it is known to yield high performance in node classification tasks (Goyal and Ferrara 2018).

DeepWalk is based on an exploration of the neighborhood of a node by truncated random-walks rooted at that node, which makes it particularly appropriate in our framework. Indeed, in the supra-adjacency representation, these random-walks will explore for each active node both the self-coupling edges leading to past and future versions of the

Table 1: Empirical temporal networks. Columns, from leftmost to rightmost: data set name, number of active nodes, number of nodes, number of timestamps, number of temporal edges, average weight of temporal edges, average fraction of timestamps in which a node is active.

Name	$\lvert\mathcal{V}\rvert$	$\lvert V\rvert$	$\lvert T\rvert$	$\lvert E\rvert$	$\frac{1}{\lvert E\rvert}\sum_{e\in E} w(e)$	$\frac{1}{\lvert V\rvert\lvert T\rvert}\sum_{v\in V}\lvert\mathcal{V}_v\rvert$
InVS15	22451	217	699	37582	4.164	0.148
LH10	4880	76	342	14870	4.448	0.188
SFHH	10815	403	127	34446	4.079	0.211
Thiers13	32546	327	246	71724	5.256	0.405
LyonSchool	17174	242	104	89640	2.806	0.682

same original node, and the edges representing the interactions between nodes. As written above, these edges encode the paths along which dynamical processes occur, meaning that the final embedding will preserve structural similarities relevant to these dynamical processes. Note that DeepWalk does not consider weighted edges, but it can easily be generalized so that the random walks take into account edge weights (Grover and Leskovec 2016).

Prediction of dynamical states

Once we have obtained an embedding for the supra-adjacency representation of the temporal network, we can turn to the task of predicting the dynamical states of active nodes. Since we assume that the set of possible states is known, this is naturally cast as a (supervised) classification task, in which each active node should be classified into one of the possible states. In our specific case, the three possible node states are S, I, and R. Note that the classification task is not informed by the actual dynamical process (except knowing the set of possible node states). In particular, no information is available about the possible transitions nor about the parameters of the actual process.

We will use here a one-vs-rest logistic regression classifier, which is customarily used in multi-label node classification tasks based on embedding vectors. Naturally, we could use any other suitable classifier.

We remark that we seek to predict node states for individual realizations of the dynamics. This is relevant to several applications: for example, in the context of epidemic spreading, and given a temporal interaction network, one might use such a predictive capability to infer the state of all nodes from the observed states of few active nodes ("sentinel" nodes).

Experiments

We study the effectiveness of the DyANE, in particular with the *dyn-supra*+DeepWalk combination, in the context of node classification tasks. For our experiments we use temporal networks built from empirical datasets that describe close-range proximity interactions of persons in a variety of real world environments. We simulate the SIR (Susceptible-Infected-Recovered) dynamical process described above over these temporal networks, generating state labels for all active nodes.

Based on the above temporal networks and node labels, we run DyANE with different combinations of supra-adjacency representations and of embedding methods for the static network, and use the resulting embedding vectors as inputs to a supervised multi-label classifications tasks. We test the sensitivity of our approach with respect to the choice of parameters and to the number of sampled active nodes $\mathcal{D}$. Finally, we also compare classifiers based on DyANE embeddings with methods that directly embed the nodes of a temporal network without relying on a supra-adjacency representation.

Data sets and dynamical process

We used publicly available data sets describing the face-to-face proximity of individuals with a temporal resolution of 20 seconds (Cattuto et al. 2010). These datasets were collected by the SocioPatterns collaboration[1] and we specifically use data sets collected in a variety of contexts, namely in offices ("InVS15"), a hospital ("LH10"), a highschool ("Thiers13"), a conference ("SFHH") and a school ("LyonSchool") (Génois and Barrat 2018). We built a temporal network from each data set by aggregating the data on 600 seconds time windows. Whenever multiple proximity events were registered between two individuals within a time window, we used the number of such events as the edge weight. Table 1 shows some basic statistics for each data set.

We simulated the SIR model on each data set (with the original temporal resolution of 20 seconds), using the following five combinations of epidemic parameters: $(\beta,\mu) = \{(0.25, 0.002), (0.125, 0.004), (0.125, 0.002), (0.125, 0.001), (0.0625, 0.002)\}$. In each case, we consider as initial state a single randomly selected node as seed, setting its state as infectious, with all others susceptible. Given the stochastic nature of the model, in some cases the infectious state barely spreads, with a large majority of the nodes remaining susceptible. The prediction task would then be trivial, and we restrict our study to non-trivial cases in which there is still at least one infectious node when more than half of the total data set time span has elapsed (i.e., $|I_{|T|/2}|\geq 1$). We thus run the SIR model up to 100 times for each data set until we obtain a simulation in which this condition is met. If the condition is not met in any of the 100 simulations, we discard the corresponding case (see Table 2). For each selected simulation, we assign as ground truth label to each active node (v_i, t_a) the state of node v_i at time t_a. Table 2 shows the proportion of each label among active nodes for each case.

We select uniformly at random $|D|= \rho|V|$ active nodes, and build our training data using those nodes and the corresponding node states. Unless otherwise noted, $\rho = 1$. We evaluate the prediction performance on a test data consisting of the remaining active nodes and their states. We report the prediction performance averaged over five realizations of the random choice of training data, for each data set and parameter values.

Evaluation

We quantified the prediction performance of each method by the Macro-F1 and Micro-F1 indices, which are widely used

[1] http://www.sociopatterns.org/

Table 2: Proportion of the three possible states S, I and R among the active nodes, for each data set and each parameter set of the SIR model. "-" indicates the cases for which the epidemic did not spread sufficiently (i.e., $|I_{|T|/2}|= 0$).

	$\beta = 0.25$, $\mu = 0.002$			$\beta = 0.125$, $\mu = 0.004$			$\beta = 0.125$, $\mu = 0.002$			$\beta = 0.125$, $\mu = 0.001$			$\beta = 0.0625$, $\mu = 0.002$		
Data set	P(S)	P(I)	P(R)	P(S)	P(I)	P(R)	P(S)	P(I)	P(R)	P(S)	P(I)	P(R)	P(S)	P(I)	P(R)
InVS15	0.171	0.0500	0.778	0.103	0.0534	0.843	0.154	0.0457	0.799	-	-	-	0.123	0.0769	0.799
LH10	0.0801	0.152	0.767	0.292	0.0975	0.609	0.197	0.120	0.681	-	-	-	0.0891	0.168	0.742
SFHH	0.0488	0.259	0.692	0.337	0.195	0.467	0.653	0.155	0.191	0.0737	0.151	0.774	0.163	0.333	0.502
Thiers13	0.0340	0.0908	0.875	0.139	0.0959	0.764	0.183	0.0923	0.723	0.240	0.051	0.707	0.111	0.164	0.723
LyonSchool	0.0955	0.155	0.748	0.104	0.191	0.703	0.128	0.156	0.715	0.102	0.0908	0.806	0.107	0.311	0.580

for evaluating multi-label node classification tasks (Goyal and Ferrara 2018). These indices range between 0 and 1, with higher values indicating better performance. Macro-F1 is an unweighted average of the F1 scores of each label, while Micro-F1 gives more importance to the labels which are more represented. When classes are imbalanced, Macro-F1 and Micro-F1 give a measure of the effectiveness of the method on respectively the smaller and the larger classes. As the number of active nodes in each state might indeed be imbalanced here, we measure both Macro- and Micro-F1 to evaluate models by the prediction performance against both the smaller and the larger classes.

Baseline methods

We considered variations of our method both at the level of the supra-adjacency representation and of the static embedding method. First, as a variation of our proposed supra-adjacency representation (*dyn-supra*), we consider a "baseline" supra-adjacency representation, which we denote by *mlayer-supra*, in which we simply map each temporal edge (v_i, v_j, t) to a single edge between active nodes, namely $((v_i, t), (v_j, t))$, similarly to the original supra-adjacency representation developed for multilayer networks (Kivelä et al. 2014). Self-coupling edges are drawn as in *dyn-supra*. We also considered two alternative embedding methods for the static networks obtained by each supra-adjacency representation: (i) LINE (Tang et al. 2015), which embeds nodes in a way to preserve both first and second-order proximity; (ii) Graph Factorization (GF) (Ahmed et al. 2013), which considers first-order proximity.

In addition, we considered two methods for temporal network embedding, which thus do not use the intermediate supra-adjacency representation, namely: (i) DynamicTriad (DTriad) (Zhou et al. 2018), which embeds the temporal network by modeling triadic closure events; (ii) DynGEM (Goyal et al. 2018), which is based on a deep learning model. It outputs an embedding for the network of each timestamp, initializing the model at timestamp $t + 1$ with the parameters found at time t, thus transferring knowledge from t to $t + 1$ and learning about the changes from G_t to G_{t+1}. Overall, we obtain eight methods – six variations of DyANE and two methods that directly embed temporal networks – which we denote respectively *dyn-supra*+DeepWalk, *dyn-supra*+LINE, *dyn-supra*+GF, *mlayer-supra*+DeepWalk, *mlayer-supra*+LINE, *mlayer-supra*+GF, DTriad, and DynGEM.

We used publicly available implementations of all embedding methods, namely the implementation of LINE[2], DynamicTriad[3], and DynGEM[4] by the original authors, and the implementation of GF by GEM[5]. As for DeepWalk, we used an implementation of node2vec[6] with $p = q = 1$. Unless otherwise noted, we conducted experiments with embedding dimension $d = 128$ and self-coupling edge weight $\omega = 1$. We used the default values of each implementation of the embedding methods, except for the number of iterations of LINE and GF, which we took equal to the number of samples of DeepWalk. We used Scikit-Learn (Pedregosa et al. 2011) to implement one-vs-rest logistic regression, with the default hyper-parameter setting.

Prediction performance

Figure 2 shows the prediction performance of the eight methods considered, for all data sets and SIR parameters considered. The *dyn-supra* representation combined with DeepWalk yields almost always the highest value both for Macro-F1 and Micro-F1. While *dyn-supra*+DeepWalk is consistently the best method for both Macro-F1 and Micro-F1 across all considered data sets and SIR parameters, DynGEM yields a higher Micro-F1 value in a few cases. However, DynGEM's Macro-F1 score is rather low in such cases.

We moreover observe that: (i) for a given static embedding method, the *dyn-supra* supra-adjacency representation gives better results than the baseline (*mlayer-supra*) one; and (ii) for a given supra-adjacency representation, DeepWalk performs better than LINE, which in turn outperforms GF.

Finally, we show in the supplementary material that the *dyn-supra*+DeepWalk prediction recovers the overall temporal evolution of the spreading process.

Sensitivity analysis

We first investigate here the effect of the hyper-parameters, of the supra-adjacency representation (the weight ω of self-coupling edges) and of the embedding (the embedding dimension d). We show in Fig. 3 the results obtained for the Macro-F1, for the InVS15 data set and $(\beta, \mu) = (0.025, 0.002)$, but we have confirmed the same tendency for the other data sets, parameters and also for Micro-F1 values (the omitted results are included in the Supplementary Materials). The results show that the performance of

[2]https://github.com/tangjianpku/LINE

[3]https://github.com/luckiezhou/DynamicTriad

[4]http://www-scf.usc.edu/~nkamra/

[5]https://github.com/palash1992/GEM

[6]https://snap.stanford.edu/node2vec/

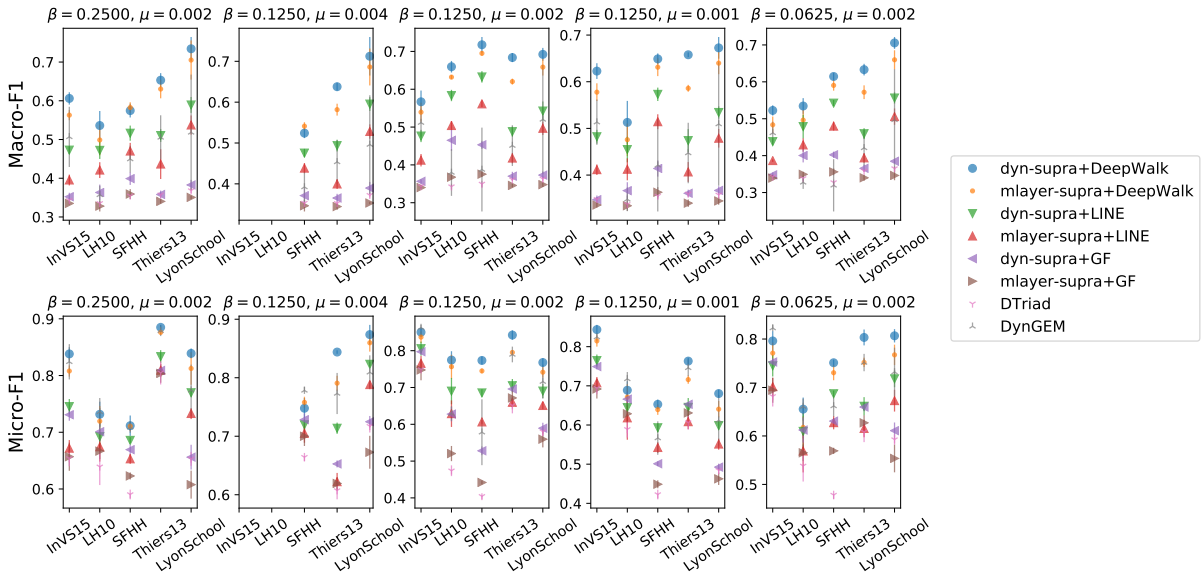


Figure 2: Prediction performance of each method, for each data set and spreading parameter set. The top and bottom row indicate results of Macro-F1 and Micro-F1, respectively.

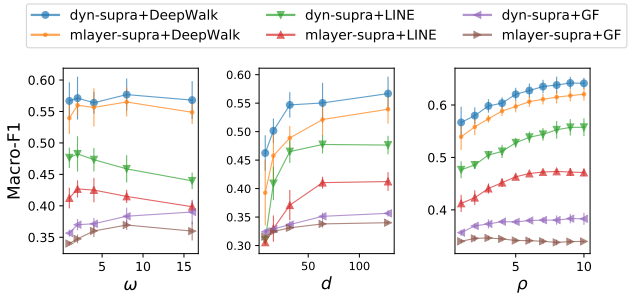


Figure 3: Results of the hyper-parameter sensitivity on the InVS15 data set. Bars indicate the standard deviation.

dyn-supra+DeepWalk is very stable with respect to changes in ω, while other methods depend weakly on ω. The performance of all methods is stable on a wide range of embedding dimensions, and decrease when it becomes smaller than 32. Overall, *dyn-supra*+DeepWalk remains the most effective method without the need for fine-tuning ω or d.

Figure 3 also shows the effect of increasing the parameter ρ, i.e., of being able to observe a larger number of active nodes. The performance increases with ρ for most methods, and in particular for *dyn-supra*+DeepWalk, which consistently yields the best result at all values of ρ.

As mentioned above, we moreover considered two variations if the *dyn-supra* representation: (i) we regard edges as directed towards increasing timestamps (*dyn-supra-directed*); (ii) we let the weight of an edge decay with increasing temporal lag between the active nodes it links, e.g., we modulate the edge weight according to the reciprocal of the lag (*dyn-supra-decay*). We also consider these variations for *mlayer-supra* representation, yielding *mlayer-supra-directed* and *mlayer-supra-decay*, respectively. Notice that, in the *mlayer-supra*, the supra-adjacency edges representing temporal edges are not affected by both variations. We report in Fig. 4 the results for $(\beta, \mu) = (0.1250, 0.002)$ and for the DeepWalk embedding, as DeepWalk overall yielded the best results. We checked that the results of Fig. 4 hold similarly for the LINE and GF embeddings.

Figure 4 indicates that using directed edges worsens the performance of both *dyn-supra* and *mlayer-supra* methods. Introducing weights depending on the time difference between active edges also worsens the performance for *mlayer-supra*, with little effect on *dyn-supra*. Overall, the original *dyn-supra* method with undirected edges and using only the weights of the original temporal edges yields the highest prediction performance.

Related Work

Network node embedding

Numerous embedding techniques have been proposed for static networks, and we refer to the recent reviews (Cai, Zheng, and Chang 2018; Goyal and Ferrara 2018) for detailed descriptions. Most techniques encode as proximity or similarity between nodes either a first-order proximity (two nodes are more similar if they are connected by an edge with larger weight) or a second-order proximity (nodes are more similar if their neighborhoods are similar). A popular way of exploring the (structural) similarity of nodes consists in using random walks rooted at the nodes, which thus explore their neighborhoods. Two of the most well-known embedding techniques, DeepWalk and node2vec (Grover and Leskovec 2016) are indeed based on such random walks.

As temporal network data has become more widely accessible in a range of contexts, the issue of embedding dy-

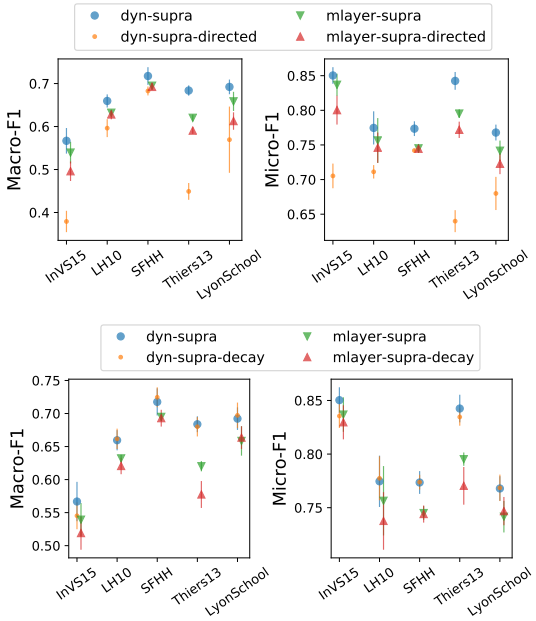


Figure 4: Effect of the variations in the supra-adjacency representation. Top: effect of using directed edges. Bottom: effect of introducing weights depending on time difference.

namically evolving networks has emerged and some methods have been put forward. As nodes' relationships evolve with time, the structure of their neighborhoods and the similarity or proximity between nodes indeed evolve as well. The typical approach consists then (in discrete time, although see (Nguyen et al. 2018) for a case of continuous time embedding) in defining a distinct embedding for each temporal snapshot and to ensure some continuity between the embeddings of successive snapshots (Zhou et al. 2018; Goyal et al. 2018; Goyal, Chhetri, and Canedo 2019). For a comprehensive review of recent embedding methods for dynamically evolving networks see (Kazemi et al. 2019).

None of these methods however consider, as we do here, supra-adjacency representations of the temporal network as a whole in order to then take advantage of static network techniques for the embedding of temporal networks.

Supra-adjacency representation

The supra-adjacency matrix representation has been developed for multilayer networks (Gómez et al. 2013; Kivelä et al. 2014), in which nodes interact on different layers (for instance different communication channels in a social network). Using the pairs (node, layer) as elementary entities, this representation builds a graph between these pairs, consisting in (i) the links within each layer, such as $(i,\alpha)-(j,\alpha)$ if nodes i and j are linked in layer α, and (ii) the links between different copies of the same node in different layers, such as $(i,\alpha)-(i,\beta)$ between layers α and β. The adjacency matrix of this new graph is the supra-adjacency matrix. This representation has proven very convenient in the study of processes on multilayer networks (Kivelä et al. 2014) as it makes it possible to use the methods and theoretical results developed using adjacency matrix of simple graphs.

It has been generalized to temporal networks, seen as special multilayer networks in which every timestamp is a layer: in the supra-adjacency representation, each node is identified by the pair of indices (i,t), corresponding to the node label i and the time frame t, respectively (Valdano et al. 2015). Most importantly, (i) each node (i,t) is linked by a directed edge to its successive copy $(i,t+1)$, but not to other future timestamps and (ii) intra-layer edges are absent: instead, if i and j are connected at time t, in the supra-adjacency representation this is replaced by two directed edges respectively from (i,t) to $(j,t+1)$ and from (j,t) to $(i,t+1)$. This representation was put forward in the context of spreading processes on temporal networks as it indeed preserves the information relevant for the spreading process (Valdano et al. 2015): if a contagion event occurs from i to j at time t, j will be contaminated (and could propagate the disease further) at $t+1$, but not yet at t.

Note that the nodes (i,t) are in this representation present for all nodes i and timestamps t, even if i is isolated at t, and that the links $(i,t)-(j,t+1)$ and $(j,t)-(i,t+1)$ also exist even if i or j have no links at t or $t+1$. The modified supra-adjacency representation we propose considers instead only the pairs (i,t) such that i is not isolated at t, in order to avoid potentially long chains $(i,t)-(i,t+1)-\cdots-$ if i remains isolated for some time, and draws links only between (i,t) and (j,t') such that i (resp. j) is not isolated at t (resp. t').

Recovering dynamics on a network

Given a dynamical process occurring on a network, such that nodes change state over time, only partial knowledge of this evolution can in general be realistically envisioned, as for instance in diffusion processes such as the spread of contagious diseases or rumors. The task of recovering the complete knowledge of a diffusion process from partial observations has thus been addressed in a variety of settings.

Some works have put forward methods to recover the state of all nodes and the seeds of a spread from a partial observation of nodes at a given time (Sundareisan, Vreeken, and Prakash 2015; Xiao, Aslay, and Gionis 2018), without attempting to recover the process whole temporal evolution. Methods to recover the state evolution of all nodes have also been proposed, using as input snapshots of the whole system, i.e., the knowledge of the state of all the nodes at a certain time (Sefer and Kingsford 2016; Feizi et al. 2018; Chen, Tong, and Ying 2019). Bayesian inference methods from partially observed snapshots have also been proposed (Altarelli et al. 2014). These works make use of strong assumptions on the nature of the underlying diffusion model.

Some works have also proposed to recover the time evolution of node states without detailed knowledge of the diffusion model (Rozenshtein et al. 2016; Xiao et al. 2018; Xiao, Aslay, and Gionis 2018). In particular, Rozenshtein et al. map the temporal network to a supra-adjacency representation almost identical to the *mlayer-supra* representation described above, in order to preserve the temporal paths on the network, and solve the recovery of the node states

as a Steiner tree problem. This relies on the fact that each node changes state only once, hence on the irreversibility of the process. Finally, we mention a method based on tensor decomposition to recover coexisting information diffusion processes from partial knowledge of the node states, without detailed knowledge of the processes, by exploiting the fact that these processes occur in synergy (Sun et al. 2017). The fraction of unknown states is however limited, while we consider a very small fraction of active node states as known.

Conclusion

Here we introduced a method for embedding nodes of temporal networks suited for the task of recovering the dynamical evolution of a single instance of a process occurring on the network, from partial observations and without information on the nature of the process itself except from the set of possible states of the nodes. Our method first maps the temporal network to a modified supra-adjacency representation, which preserves the paths on which the process unfolds. As this representation yields a static graph among the active nodes, which are pairs of the form (node of the temporal network, time of interaction), it enables the use of embedding techniques for static networks. We choose to use DeepWalk, as it is a simple and paradigmatic algorithm based on random walks and thus particularly suited to explore the neighborhood of the nodes of the supra-adjacency representation in a way relevant to the dynamical process on the network. We finally frame the inference of the dynamical state of all active nodes from a set of observations as a classification task.

We have shown the performance of our method on the concrete case of an epidemic-like model on empirical temporal networks and compared it with other state of the art methods. Our method consistently yields very good classification performance in a robust way across data sets and process parameters, without fine-tuning hyper-parameters.

Our results show that it is possible, without any knowledge of the precise nature of the process nor of its parameters, to recover crucial information on its outcome. Note in particular that our method assumes no knowledge of which transitions between states actually occur in the real dynamics: this means that the predicted sequence of states of each individual node might yield "forbidden" transitions (e.g., in the SIR example, transitions from I to S or from R to I). Nevertheless, we have shown that the outcome of the classification task gives a good estimation of the actual dynamics.

Our method has however the clear limitation that we assume the whole temporal network to be known. Although a full observation of the contact patterns of individuals could be envisioned in some specific controlled settings such as hospitals, this is not generally the case. Further work will address this limitation by considering the effect of noise and errors in the temporal network data, and by considering the case in which only a (more or less detailed) set of statistics of the temporal network is known. Noise could also impact the quality of the sampling (e.g., observational errors), and we will check its impact on our method's performance. Further work will also address different sampling strategies such as a sampling concentrated at early times, or focused on few specific "sentinel" nodes followed at all times, or of a whole snapshot of the system but only at a specific time. This could yield crucial insights on how to optimize surveillance strategies in concrete settings.

Finally, since our method is largely agnostic with respect to the specific dynamical process, we will consider other processes such as other models of disease propagation, complex contagion phenomena or opinion formation.

References

[Ahmed et al. 2013] Ahmed, A.; Shervashidze, N.; Narayanamurthy, S.; Josifovski, V.; and Smola, A. J. 2013. Distributed large-scale natural graph factorization. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, 37–48. New York, NY, USA: ACM.

[Altarelli et al. 2014] Altarelli, F.; Braunstein, A.; Dall'Asta, L.; Ingrosso, A.; and Zecchina, R. 2014. The patient-zero problem with noisy observations. *Journal of Statistical Mechanics: Theory and Experiment* 2014(10):P10016.

[Barrat, Barthélemy, and Vespignani 2008] Barrat, A.; Barthélemy, M.; and Vespignani, A. 2008. *Dynamical processes on complex networks*. Cambridge: Cambridge University Press.

[Cai, Zheng, and Chang 2018] Cai, H.; Zheng, V. W.; and Chang, K. C. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering* 30(9):1616–1637.

[Castellano, Fortunato, and Loreto 2009] Castellano, C.; Fortunato, S.; and Loreto, V. 2009. Statistical physics of social dynamics. *Rev. Mod. Phys.* 81:591–646.

[Cattuto et al. 2010] Cattuto, C.; Van den Broeck, W.; Barrat, A.; Colizza, V.; Pinton, J.-F.; and Vespignani, A. 2010. Dynamics of person-to-person interactions from distributed rfid sensor networks. *PLoS ONE* 5(7):e11596.

[Chen, Tong, and Ying 2019] Chen, Z.; Tong, H.; and Ying, L. 2019. Inferring full diffusion history from partial timestamps. *IEEE Transactions on Knowledge and Data Engineering* 1–1.

[Feizi et al. 2018] Feizi, S.; Medard, M.; Quon, G.; Kellis, M.; and Duffy, K. 2018. Network infusion to infer information sources in networks. *IEEE Transactions on Network Science and Engineering* 1–1.

[Génois and Barrat 2018] Génois, M., and Barrat, A. 2018. Can co-location be used as a proxy for face-to-face contacts. *EPJ Data Science* 7(1):11.

[Gómez et al. 2013] Gómez, S.; Díaz-Guilera, A.; Gómez-Gardeñes, J.; Pérez-Vicente, C. J.; Moreno, Y.; and Arenas, A. 2013. Diffusion dynamics on multiplex networks. *Phys. Rev. Lett.* 110:028701.

[Goyal and Ferrara 2018] Goyal, P., and Ferrara, E. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* 151:78–94.

[Goyal et al. 2018] Goyal, P.; Kamra, N.; He, X.; and Liu,

Y. 2018. Dyngem: Deep embedding method for dynamic graphs. *arXiv preprint arXiv:1805.11273*.

[Goyal, Chhetri, and Canedo 2019] Goyal, P.; Chhetri, S. R.; and Canedo, A. 2019. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems*.

[Grover and Leskovec 2016] Grover, A., and Leskovec, J. 2016. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 855–864. New York, NY, USA: ACM.

[Kazemi et al. 2019] Kazemi, S. M.; Goel, R.; Jain, K.; Kobyzev, I.; Sethi, A.; Forsyth, P.; and Poupart, P. 2019. Relational representation learning for dynamic (knowledge) graphs: A survey. *arXiv preprint arXiv:1905.11485*.

[Keeling and Rohani 2008] Keeling, M. J., and Rohani, P. 2008. *Modeling Infectious Diseases in Humans and Animals*. Princeton, N.J.: Princeton University Press.

[Kivelä et al. 2014] Kivelä, M.; Arenas, A.; Barthelemy, M.; Gleeson, J. P.; Moreno, Y.; and Porter, M. A. 2014. Multilayer networks. *Journal of Complex Networks* 2(3):203–271.

[Nguyen et al. 2018] Nguyen, G. H.; Lee, J. B.; Rossi, R. A.; Ahmed, N. K.; Koh, E.; and Kim, S. 2018. Continuous-time dynamic network embeddings. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, 969–976. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

[Pastor-Satorras et al. 2015] Pastor-Satorras, R.; Castellano, C.; Mieghem, P. V.; and Vespignani, A. 2015. Epidemic processes in complex networks. *Rev. Mod. Phys.* 87(3):925.

[Pedregosa et al. 2011] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

[Perozzi, Al-Rfou, and Skiena 2014] Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 701–710. New York, NY, USA: ACM.

[Rozenshtein et al. 2016] Rozenshtein, P.; Gionis, A.; Prakash, B. A.; and Vreeken, J. 2016. Reconstructing an epidemic over time. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1835–1844. ACM.

[Sefer and Kingsford 2016] Sefer, E., and Kingsford, C. 2016. Diffusion archeology for diffusion progression history reconstruction. *Knowledge and Information Systems* 49(2):403–427.

[Sun et al. 2017] Sun, Y.; Qian, C.; Yang, N.; and Yu, P. S. 2017. Collaborative inference of coexisting information diffusions. In *2017 IEEE International Conference on Data Mining (ICDM)*, 1093–1098.

[Sundareisan, Vreeken, and Prakash 2015] Sundareisan, S.; Vreeken, J.; and Prakash, B. A. 2015. Hidden hazards: Finding missing nodes in large graph epidemics. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, 415–423. SIAM.

[Tang et al. 2015] Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, 1067–1077. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

[Valdano et al. 2015] Valdano, E.; Ferreri, L.; Poletto, C.; and Colizza, V. 2015. Analytical computation of the epidemic threshold on temporal networks. *Phys. Rev. X* 5:021005.

[Xiao, Aslay, and Gionis 2018] Xiao, H.; Aslay, C.; and Gionis, A. 2018. Robust cascade reconstruction by steiner tree sampling. In *2018 IEEE International Conference on Data Mining (ICDM)*, 637–646.

[Xiao et al. 2018] Xiao, H.; Rozenshtein, P.; Tatti, N.; and Gionis, A. 2018. Reconstructing a cascade from temporal observations. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, 666–674. SIAM.

[Zhou et al. 2018] Zhou, L.; Yang, Y.; Ren, X.; Wu, F.; and Zhuang, Y. 2018. Dynamic network embedding by modeling triadic closure process. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

arXiv:1909.05976v1 [physics.soc-ph] 12 Sep 2019

DyANE: Dynamics-aware node embedding for temporal networks
Supplemental Material

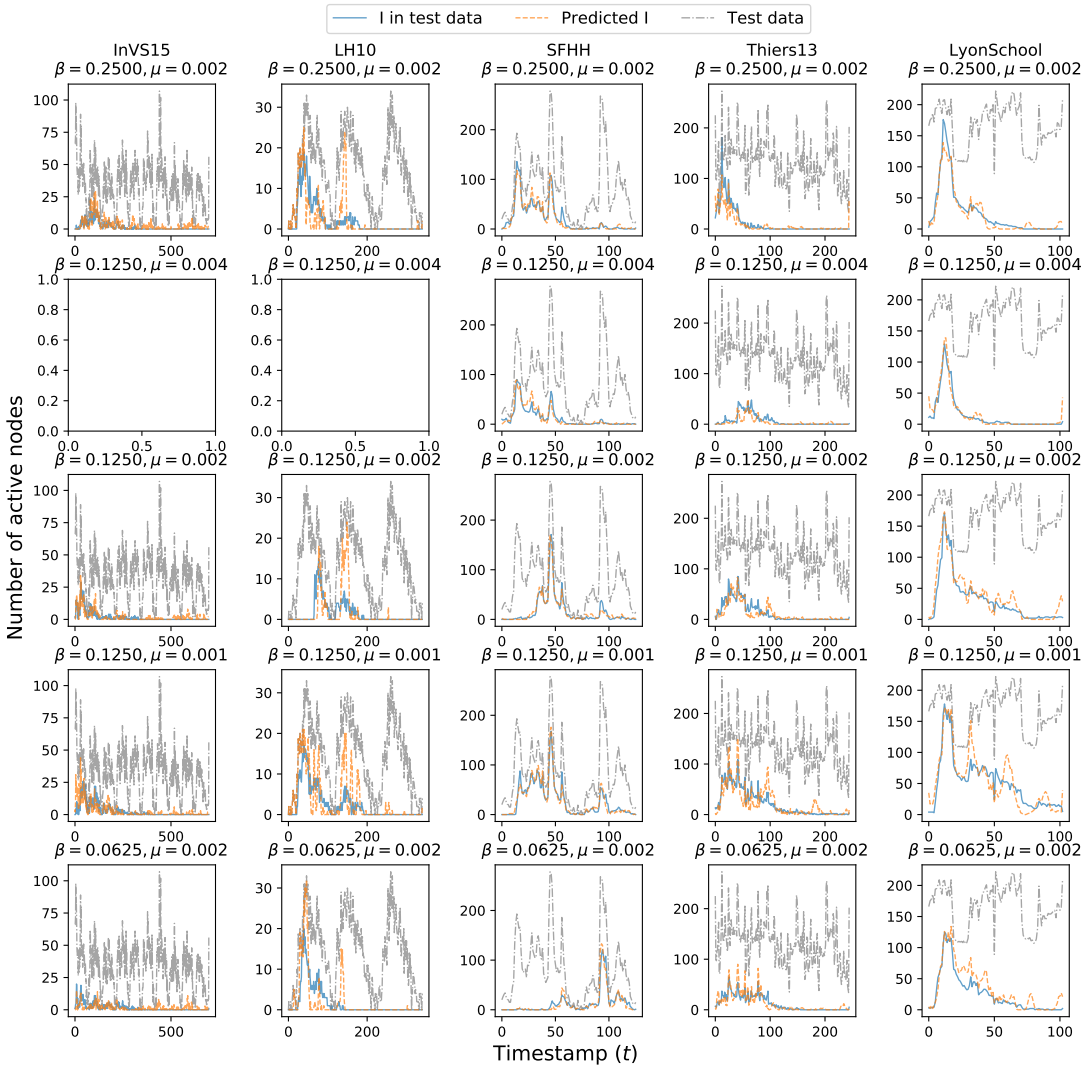


Figure 1: One example of predicted timeline of the number of active nodes in the infectious state, for *dyn-supra*+DeepWalk, for each SIR parameter set and each data set. The blue, orange and gray lines are respectively the number of actual active nodes in state I in the test data, the number of predicted active nodes in state I and the number of active nodes in the test data at time t. Note that the number of active nodes in the test data is almost the same as the total number of active nodes, as the training data is of small size ($\rho = 1$, i.e., $|D| = |V|$).

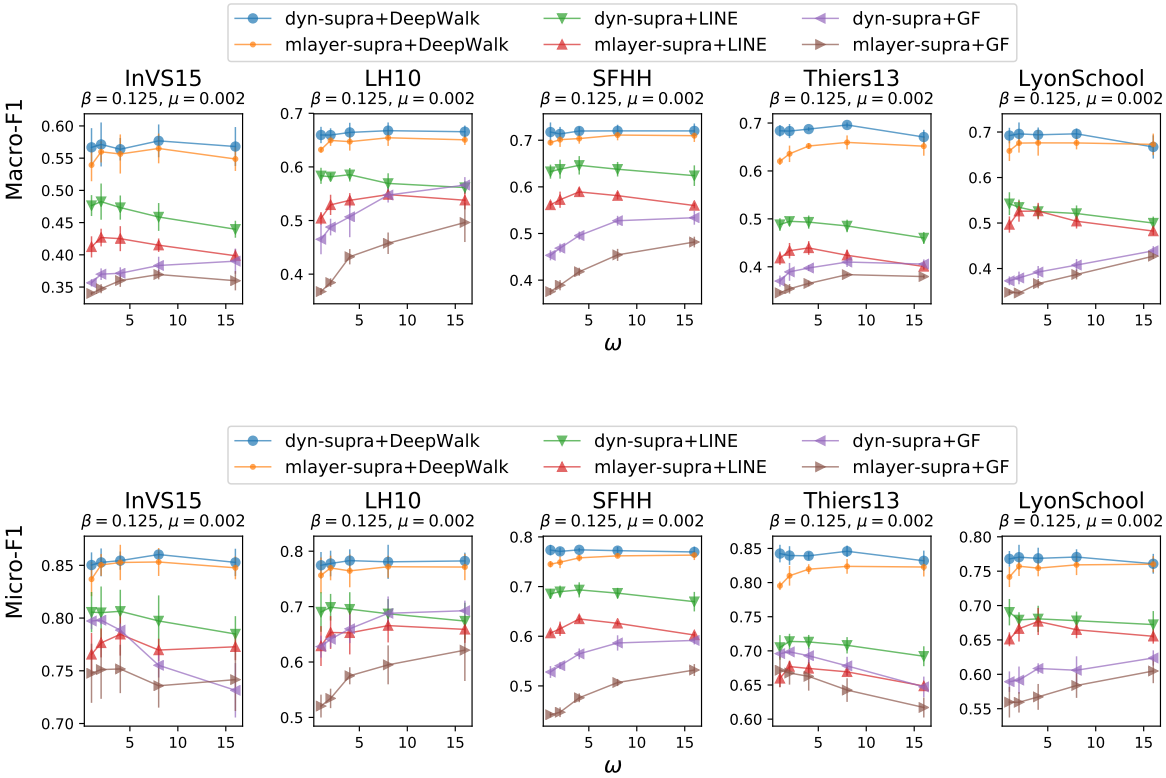


Figure 2: Effect of the self-coupling edges weight ω. The top and bottom row indicate results of Macro-F1 and Micro-F1, respectively.

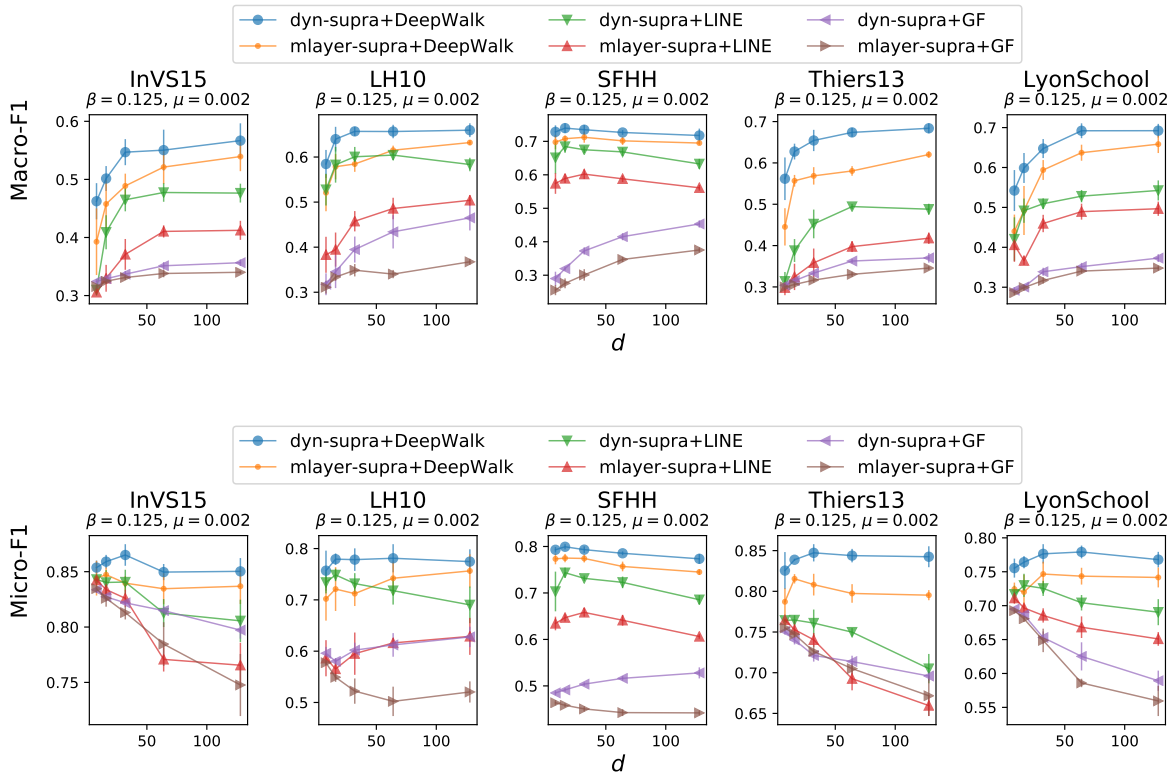


Figure 3: Effect of the embedding dimension d. The top and bottom row indicate results of Macro-F1 and Micro-F1, respectively.

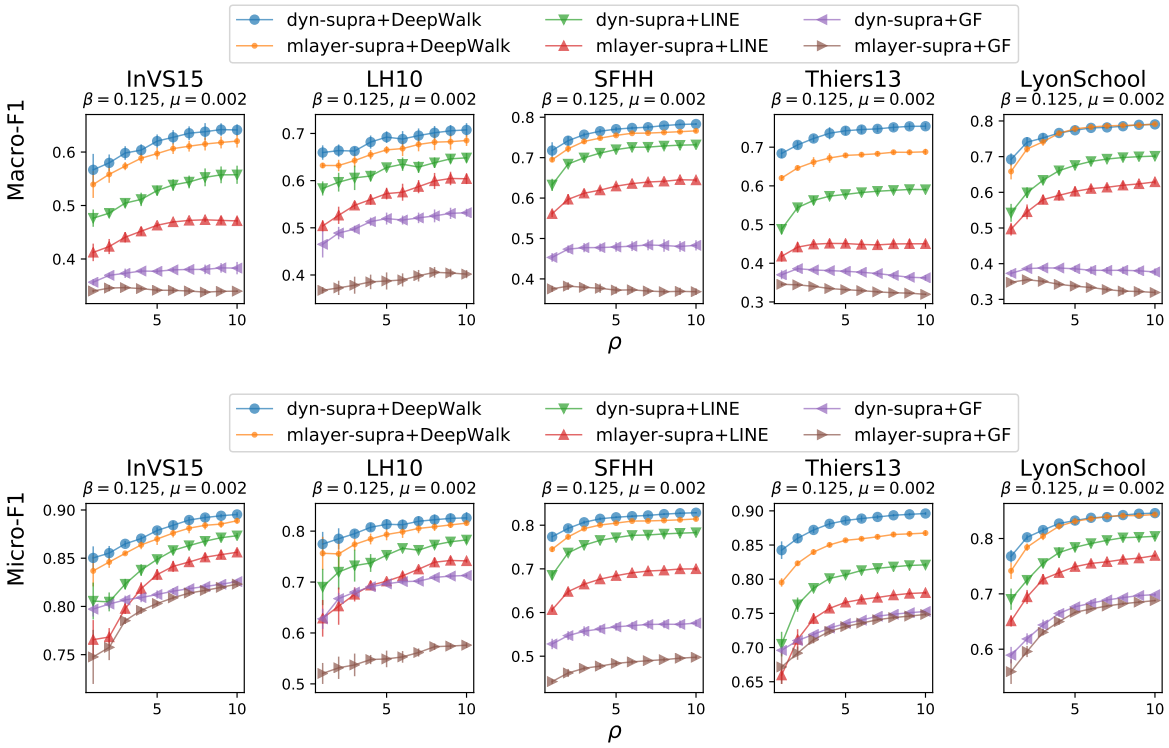


Figure 4: Effect of the sampling parameter ρ. The top and bottom row indicate results of Macro-F1 and Micro-F1, respectively.

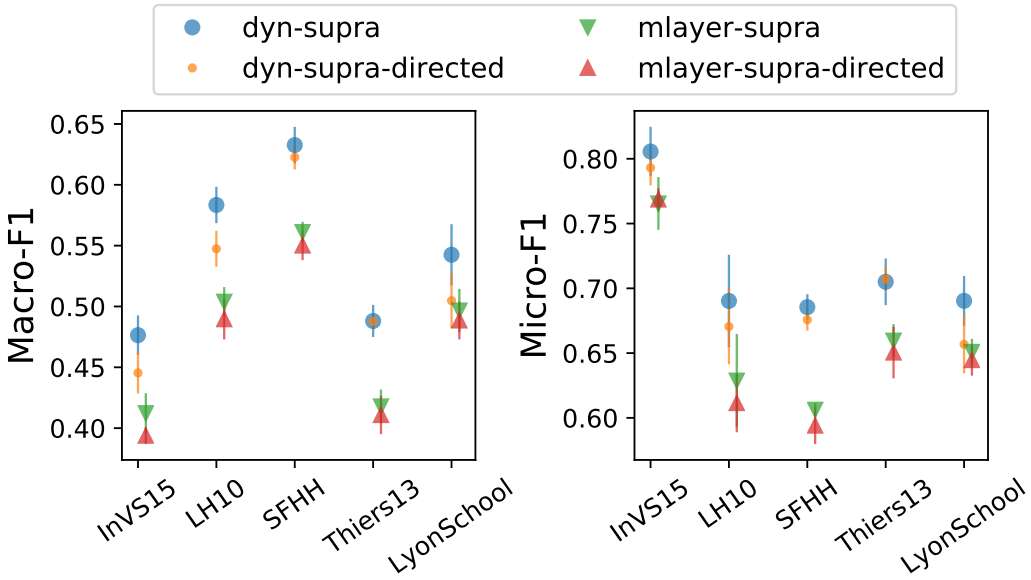
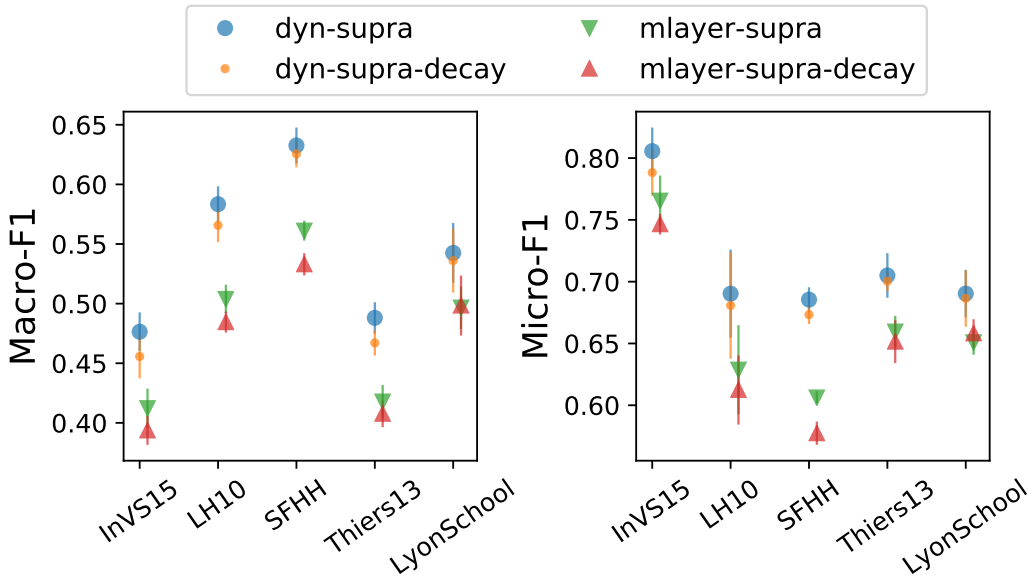
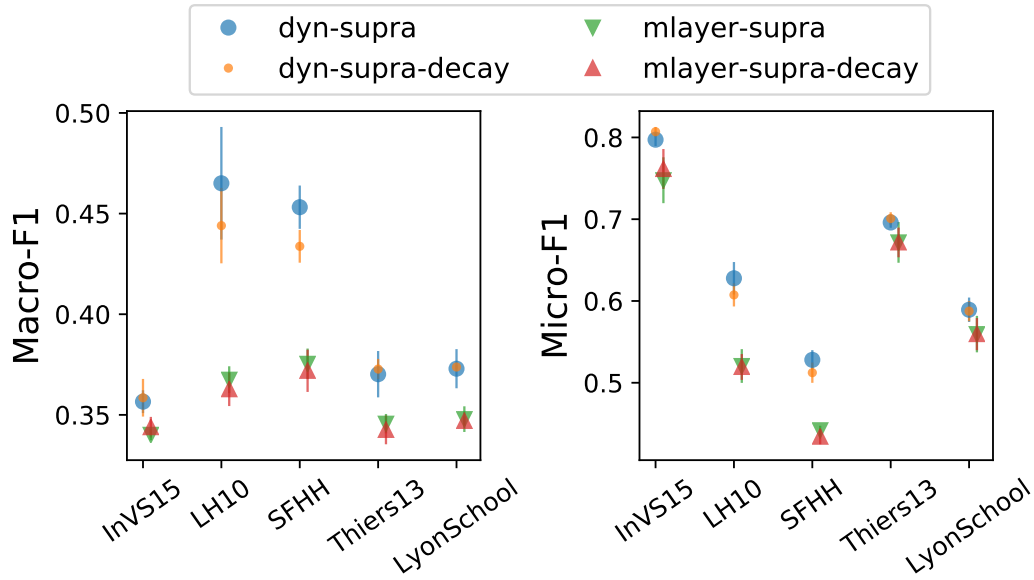


Figure 5: Effect on the reconstruction performance of using directed edges and the LINE embedding (we show only the results for LINE as GF is not applicable to directed networks). Here $(\beta, \mu) = (0.1250, 0.002)$.



(a) Results obtained with the LINE embedding.



(b) Results obtained with the GF embedding.

Figure 6: Effect on the reconstruction performance of introducing weights depending on time difference. These results are for $(\beta, \mu) = (0.1250, 0.002)$.