Link Prediction Based on Stochastic Information Diffusion

Didier A. Vega-Oliveros, Liang Zhao, *Senior Member, IEEE*, Anderson Rocha, *Senior Member, IEEE*, and Lilian Berton

***Abstract*— Link prediction (LP) in networks aims at determining future interactions among elements; it is a critical machine-learning tool in different domains, ranging from genomics to social networks to marketing, especially in e-commerce recommender systems. Although many LP techniques have been developed in the prior art, most of them consider only static structures of the underlying networks, rarely incorporating the network's information flow. Exploiting the impact of dynamic streams, such as information diffusion, is still an open research topic for LP. Information diffusion allows nodes to receive information beyond their social circles, which, in turn, can influence the creation of new links. In this work, we analyze the LP effects through two diffusion approaches, susceptible-infected-recovered and independent cascade. As a result, we propose the progressive-diffusion (PD) method for LP based on nodes' propagation dynamics. The proposed model leverages a stochastic discrete-time rumor model centered on each node's propagation dynamics. It presents low-memory and low-processing footprints and is amenable to parallel and distributed processing implementation. Finally, we also introduce an evaluation metric for LP methods considering both the information diffusion capacity and the LP accuracy. Experimental results on a series of benchmarks attest to the proposed method's effectiveness compared with the prior art in both criteria.**

***Index Terms*— Diffusion process, edge additions, graph based, information spreading, link prediction (LP), network evolution.**

Manuscript received 31 August 2019; revised 1 April 2020 and 13 October 2020; accepted 16 January 2021 . Date of publication 4 February 2021; date of current version 4 August 2022. This work was supported in part by the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded through Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) under Grant 2013/07375-0; in part by the FAPESP under Grant 18/01722-3 and Grant 2017/12646-3; in part by the C4AI of FAPESP/IBM/USP under Grant 2019/07665-4; in part by the FAPESP under Grant 19/26283-5, Grant 18/24260-5, and Grant 16/23698-1; and in part by the FAPESP under Grant 15/50122-0 and DFG-GRTK under Grant 1740/2. *(Corresponding author: Didier A. Vega-Oliveros.)*

Didier A. Vega-Oliveros is with the RECOD Laboratory, Institute of Computing, University of Campinas, Campinas/SP CEP 13083-852, Brazil, and also with the Center for Complex Networks and Systems Research, Luddy School of Informatics, Computing and Engineering, Indiana University, Bloomington, IN 47408 USA (e-mail: davegaol@iu.edu).

Liang Zhao is with the Department of Computing and Mathematics, Faculty of Philosophy, Science, and Letters at Ribeirão Preto (FFCLRP), University of São Paulo, Ribeirão Preto/SP CEP 14040-901, Brazil (e-mail: zhao@usp.br).

Anderson Rocha is with the RECOD Laboratory, Institute of Computing, University of Campinas, Campinas/SP CEP 13083-852, Brazil (e-mail: anderson.rocha@ic.unicamp.br).

Lilian Berton is with the Institute of Science and Technology, Federal University of São Paulo, São José dos Campos/SP CEP 12247-014, Brazil (e-mail: lberton@unifesp.br).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2021.3053263.

Digital Object Identifier 10.1109/TNNLS.2021.3053263

I. Introduction

NETWORKS are powerful modeling tools for an extensive range of real-world systems, especially for uncovering interaction patterns among the elements or groups of elements that the systems comprise [1]. A network (or a graph) consists of a set of nodes and a set of links between each pair of nodes. In this structure, a node corresponds to an element (e.g., a person in a social network), and edges represent an association between two elements. How to take advantage of connections between the elements is a fundamental task in network understanding to improve decision-making in different areas, such as marketing, politics, and business. This is even more important with the growth of social network platforms, such as Twitter, Facebook, WhatsApp, and We-chat. However, the nodes of many networks are not i.i.d. (independent and identically distributed), rather presenting interdependency and complex relations among them. Such complexity behind relational data sets brings challenges to traditional machine-learning algorithms. Thus, different network data techniques have been developed for different tasks, such as pattern classification [2], natural language processing [3], forecasting, and time-series analysis [4], among others.

Networks are also dynamically evolving structures in which connections may appear or disappear from time to time. In this context, link prediction (LP) aims at anticipating future associations [5], [6] and various applications directly benefit from such predictions, such as friendship analysis in social networks [7], associations and monitoring of suspects in terrorist networks [8], protein associations [9], recommendation systems in e-commerce [10], and relevant future collaborations in cooperation networks [11].

Given a network, we have access to both its local and global structures, such as the neighborhood of a node, the distance among nodes, and the nodes' community structure. LP methods aim at discovering potential links with structural influence based on a local or global view, which can lead to different prediction results. Typically, the local metrics are generic and straightforward, whereas global methods need substantial processing time and often cannot handle large and dense networks. Due to the complex structure and noise nature of networks, it is difficult to predict future links, and it is unlikely to develop a method that can outperform all others [5], [6].

Modeling information diffusion in online social networks is a challenging problem, and various researchers contributed to this end [12]–[14]. In the LP research area, some strategies

consider a random walk in a network, such as DeepWalk [15], which has been proposed to search likely edges using uniform random walks, and Node2vec [16], which explores network neighborhoods through unbalanced random walks. In the diffusion processes, Ally *et al.* [17] proposed two rewiring models and compared the effects on information spreading in scale-free and small-world networks. However, the authors did not consider adding new edges or the effects on the networks' structure. In [18], the authors have addressed Sina Weibo and detected an important feature from the information diffusion process, which promoted LP performance. Wu *et al.* [19] proposed a framework called influential nodes identification LP (INILP) to quantify the importance of a node in a network by assigning each node a ranking score. The influence of a node represents its ability to spread information to other nodes. However, the proposed metrics are structural node rankings of centrality measures, not a proper influential spreading model, such as epidemics, rumors, or information propagation. Recently, Wang *et al.* [20] proposed a neighborhood adversarial LP method, but they disregarded the local influence that the diffusion dynamics have on the link formation.

In this article, we propose a progressive-diffusion (PD) method for LP that improves information propagation. The method's rationale is that the node's local dynamics in the early stage of an information diffusion process, i.e., the microscopic interactions of "who influenced whom," positively predicts reliable new links in the network. The method leverages a stochastic discrete-time rumor model centered on each node's propagation dynamics, in which a small number of iterations are required. The model works on arbitrary network structures and presents a low-processing and low-memory footprint. The differences with respect to the prior art include incorporating local diffusion information into the LP task and using intermediate information of the network, being amenable to parallel and distributed processing implementation.

We also present some contributions in terms of evaluating LP when considering local and global properties of a network. Existing LP methods mostly focus on classification tasks (prediction accuracy of new links) [5], [6], [20]–[22]. However, it is somewhat intuitive that new links are also directly related to the information diffusion in the network [7], [18], [23]. For example, users in online communication platforms not only passively receive information but generate and disseminate news pieces, messages, and memes. Weng *et al.* [7] reported evidence of the role of information diffusion in evolving social networks. Vega-Oliveros *et al.* [23] further explored this aspect and evaluated the impact of edge additions, information diffusion, and structural properties on the evolved networks. In this latter study, although the authors assessed rules that are more consistent with longitudinal structural changes observed in a data set, they did not propose any LP strategy neither diffusion models in their formulation. In [21], the authors proposed a measure based on the geometric mean of the area under the receiver operator's characteristic curve (AUC) for evaluating the accuracy of LP methods. Nevertheless, the authors did not consider either the diffusion process, network evolution, or structural characterization in their formulation.

The spreading capacity is also a relevant metric to be considered in LP research, which is currently ignored. However, improving the spreading capacity and link classification task simultaneously is not a simple task. For instance, the random inclusion of edges improves the spreading capacity, but it has a small effect in improving classification performance, as expected. On the other hand, some of the LP methods' inclusion of new edges may not improve the spreading capacity, which is contrary to our expectation. In practical situations, a suitable LP method should have good performance in both aspects: the spreading capacity and the LP classification tasks.

In summary, the main contributions of this work are threefold.

1) We introduce an LP method that leverages the network's local and progressive diffusion of information over time. The method considers the structure of the underlying network (static information) and also introduces a dynamic process into LP, incorporating both micro and macro information in the prediction.
2) The method presents low-memory and low-processing footprint and outperforms random-walk- and stochastic-based strategies. Only a small number of iterations are required in the proposed method's prediction task. The method also leverages intermediate information of the network, such as quasi-local/global information, parallel, and distributive processing.
3) We present a performance assessment methodology that considers the spreading capacity and the classification accuracy of LP tasks.

We performed extensive experiments with eight real-world data sets and one synthetic benchmark and made comparisons with different methods: common neighbors (CN), Jaccard Similarity (JC), Adamic Adar (AA), Rooted PageRank (RP), SimRank (SR), GraphDistance (GD), DeepWalk (DW), Node2vec (NV), and the variational graph autoencoder (VGAE).

We organized this article as follows. Section II presents the proposed LP method and the progressive-diffusion (PD) process. Section III discusses the adopted benchmarks and methods in prior art used for comparisons. Section IV presents the experimental results and the main research findings. Finally, Section V concludes the work and entails some ideas for future work.

II. Link Prediction Method Based on Progressive-Diffusion Process

We introduce a link prediction method that relies upon a progressive discrete rumor diffusion model centered on each node's dynamics. The proposed rumor model is a stochastic approach that considers the microscopic dynamics of the diffusion and is applicable to arbitrary network structures, different from the broadly reported macroscopic homogeneous mean-field models [12] in the prior art. In this way, the proposed model's spreading process takes place via the contact interaction between neighbor nodes.

As a rumor diffusion model underpins our methodology, it is worth differentiating the spreading behavior in such models from that in epidemic ones. In epidemic dynamics [12], each active node uniformly selects a node at each time, and the

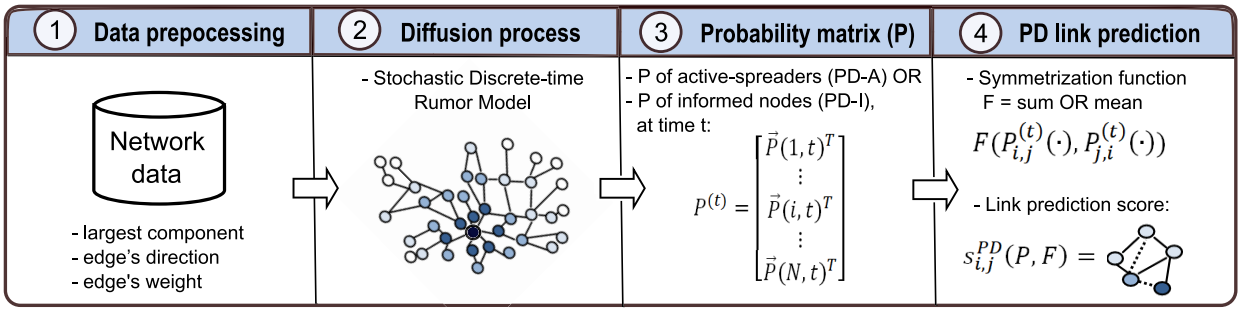


Fig. 1. Graphical model. Given a network data set, some information piece is propagated to neighbors from an informed node by a discrete-time stochastic rumor model. A probability matrix P is generated, indicating how likely node j will be informed (PD-I) or an active spreader (PD-A) at time t when the rumor started on node i. However, matrix P is not symmetric, and a symmetrization function F should be applied. Finally, the nodes with higher similarity in the diffusion process will be ranked to receive a new connection.

deactivation is spontaneous, whereas, in rumor's behavior, the activation/deactivation of nodes occurs by the interaction with informed pairs [12], [23].

Fig. 1 shows the pipeline and idea of the proposed PD method. We first preprocess the data set into a network representation, where we can choose to consider the largest component of the network and the edge's direction, among other features. When the data are not a graph or a relational representation, we can transform the data set into a network by using an appropriated graph construction method [2], [4]. With the network at hand, we evaluate each node's rumor diffusion dynamics as an initial spreader by employing the proposed stochastic discrete-time rumor model (Section II-A) for this purpose.

As we are interested in obtaining the nodes' influence in the early stage of the spreading process, we do not run the diffusion until the end of the dynamic (i.e., until reaching the absorbing state). After the spreading, we obtain a diffusion probability matrix P [see Fig. 1 (3)], which indicates the probability of influence among the nodes at time t. This probability matrix is not symmetric, which means that the influence over the network that node i exerts on node j is different from the influence that j has over i. For this reason, we need to apply a symmetrization function F on the matrix P. The PD method consists of a scoring function that predicts the likelihood of potential new edges based on nodes with highly similar influence interaction or diffusion embedding on the network [see Fig. 1 (4)]. Section II-B presents the details of the probability matrix and the PD method.

The key advantages of the proposed discrete-time rumor model include its low-computational cost and the suitability for parallel and distributed processing; each node as initial spreader can be treated as an independent diffusion dynamic, as we show in Sections II-C and II-D.

A. Stochastic Discrete-Time Rumor Model

The discrete model we exploit in this work describes, for each node, its probability of being in one of two states: spreader or stifler, therefore enabling the system to incorporate evolution aspects over time. Formally, given a network $G = (V, E)$ (see Table I), it can be represented as an adjacency matrix A, where if node i is connected with j, we have $A_{ij} = 1$ or 0 otherwise. In (1), a node i has the probability to be a spreader at time quantum $(t + 1)$ if either it was already a spreader in the last step $(I_i(t))$ and it did not become stifler during that time $(\bar{r}_i(t))$ or if i was an ignorant $(S_i(t))$ but was informed by any of its neighbors $([\,1 - \bar{i}_i(t)\,])$. The probability of a spreader i to become inactive (2) depends on the probability of node i being a spreader and becoming stifler by being in contact with any of its informed neighbors $(I_i(t)[\,1 - \bar{r}_i(t)\,])$ or if node i was already a stifler in the last time step $(R_i(t))$, that is,

$$I_i(t+1) = I_i(t)\bar{r}_i(t) + S_i(t)\Big[1 - \bar{i}_i(t)\Big] \quad (1)$$

$$R_i(t+1) = R_i(t) + I_i(t)\Big[1 - \bar{r}_i(t)\Big] \quad (2)$$

where $\bar{i}_i(t)$ and $\bar{r}_i(t)$ are the transition probabilities of node i to not become a spreader nor a stifler through contact with any of its neighbors at time t, respectively. Node i does not become a spreader if it was not convinced, with probability γ, by any spreader neighbor j, formally $1-(A_{ji}/k_j)\,I_j(t)\,\gamma$. The contact happens according to the number of neighbors

TABLE I
SYMBOLS

Notation	Description
$G(V,E)$	Graph or network with the set of $\|V\| = N$ nodes and $\|E\| = M$ edges
A_{ij}	The adjacency matrix representing the graph G
k_i	The degree of node i
$\langle k \rangle$	The average degree of the network
$\langle k^2 \rangle$	The second moment of the degree distribution
$\mathbf{C}$	The network complexity measure
CC	The average clustering coefficient of the network
$\max(\ell)$	The network diameter
β, λ, γ	The transmission probability of the SIR, IC simulations, and PD diffusion model, respectively
S	The susceptible or ignorant state
I	The active spreaders state
R	The recovery, stifler or inactive node state
$\vec{i}(t)$	Inverse probability transition to turn into active spreader at time t
$\vec{r}(t)$	Inverse probability transition to turn into inactive spreader at time t
$\varphi(t)$	Probability to be an informed node at time t
$P^{(t)}(\varphi)$	Probability matrix to be an informed node
$P^{(t)}(I)$	Probability matrix to be an active spreader

(degree k_j) of node j. Because each neighbor can independently make contact with node i, $\bar{i}_i(t)$ is the product of this probability among all neighbors of i. A susceptible or ignorant node will become "infected" given the contact interaction with its spreader neighbors. In turn, if the node i is a spreader, it will become inactive depending on the contact interaction with its informed neighbors (active or inactive spreaders), i.e., $\mu\,(A_{ij}/k_i)(I_j(t) + R_j(t))$, where μ is the contact probability of the spreader to transit to the inactive state (Recovery). Therefore, $\bar{r}_i(t)$ is the complement of the sum over all neighbors of i, normalized by its degree k_i. Without loss of generality, here, we consider $\mu = 1$ in the model. However, we show in Section IV-C a parameter sensitivity evaluation in the link classification task.

Given the $N \times 1$ nodes states $\vec{S}(t)$, $\vec{I}(t)$, and $\vec{R}(t)$, the ith position on each vector represents the probability of node i to be part of the corresponding state. We formalize our model in an algebra notation as follows:

$$\vec{I}(t+1) = \vec{I}(t) \odot \vec{r}(t) + \vec{S}(t) \odot \left[1^{N\times 1} - \vec{i}(t)\right] \quad (3)$$

$$\vec{R}(t+1) = \vec{R}(t) + \vec{I}(t) \odot \left[1^{N\times 1} - \vec{r}(t)\right] \quad (4)$$

recalling that $\vec{S}(t) = 1^{N\times 1} - [\vec{R}(t) + \vec{I}(t)]$. The operator $\odot$ denotes the componentwise multiplication between two vectors, e.g., for any vectors $U^{N\times 1}$ and $V^{N\times 1}$, we have $U^{N\times 1} \odot V^{N\times 1} = [u_1 v_1, \ldots, u_N v_N]^{N\times 1}$. We define $\vec{\varphi}(t) = \vec{R}(t) + \vec{I}(t)$ as the overall probability that nodes are informed, i.e., how likely each node knows the information at time quantum t.

The vector expression for the negative transition probabilities is given by

$$\vec{i}(t) = \coprod\left(1^{N\times N} - \left[A^{(T)} \cdot \mathbf{Dig}\left(\gamma \cdot \frac{1}{\vec{K}} \odot \vec{I}(t)\right)\right]\right) \quad (5)$$

$$\vec{r}(t) = 1^{N\times 1} - \left[\left(\mathbf{Dig}\left(\frac{1}{\vec{K}} \odot \vec{\varphi}(t)\right) \cdot A\right) \cdot 1^{N\times 1}\right] \quad (6)$$

where $\vec{K}$ is the vector of centrality measures of all the nodes and $1/\vec{K}$ its reciprocal. The function $\mathbf{Dig}(U^{N\times 1})$ transforms vector U into a diagonal matrix $N \times N$. The operator $\coprod(M^{N\times N})$ denotes the vector of columns product of the given matrix, i.e.,

$$\coprod\left(M^{N\times N}\right) = \left[\prod_{j=1}^{N} M_{1,j}, \ldots, \prod_{j=1}^{N} M_{l,j}, \ldots, \prod_{j=1}^{N} M_{N,j}\right]^{N\times 1}. \quad (7)$$

B. Progressive-Diffusion Link Prediction

With the proposed diffusion framework, we can analyze the progression of each node's diffusion and influence in a specific time, given the probability matrix $P^{(t)}$. This matrix represents the probability of one node influencing (or affecting in some way) another on the network, directly or not, by the diffusion process after t time steps. For example, the probability $P^{(t)}_{i,j}$ indicates how likely node j will be informed when the rumor starts on node i. We define $\vec{I}(i,t)$ as the probability for each node to be an active spreader at time step t when the rumor started on node i. Similarly, the vector $\vec{\varphi}(i,t)$ denotes the probabilities that nodes have been informed (by active or inactive spreaders) at time t when the rumor started on node i. The active spreaders and informed probability matrices $P^{(t)}(I)$ and $P^{(t)}(\varphi)$ are defined by

$$P^{(t)}(I) = \begin{pmatrix} \vec{I}(1,t)^T \\ \vdots \\ \vec{I}(i,t)^T \\ \vdots \\ \vec{I}(N,t)^T \end{pmatrix}, \quad P^{(t)}(\varphi) = \begin{pmatrix} \vec{\varphi}(1,t)^T \\ \vdots \\ \vec{\varphi}(i,t)^T \\ \vdots \\ \vec{\varphi}(N,t)^T \end{pmatrix} \quad (8)$$

in which $\vec{I}(i,t)$ and $\vec{R}(i,t)$ are initially column vectors ($N \times 1$), but we transpose them to row vectors, just for clarity. The matrix values are related to the probability of arrival, in connection with a random walk, or to be infected by a specific node. The matrices $P^{(t)}(\cdot)$ are not symmetric [i.e., $P^{(t)}_{i,j}(\cdot) \neq P^{(t)}_{j,i}(\cdot)$] as the paths and behavior of each node are particular for infecting others. For this reason, we consider that nodes with similar particularities in the diffusion process share an equivalent perspective or embedding on the network.

We consider that nodes have higher interaction with similar pairs in the diffusion process, which shapes the structure of the network, like the communities [24]. This point has been previously explored [7], [18], [23]. For this reason, we consider that two nodes are similar if they share higher contagion characteristics on the network. Therefore, the generalized PD LP score is

$$s^{PD}_{i,j}\left(P, F\right) = F\left(P^{(t)}_{i,j}(\cdot), P^{(t)}_{j,i}(\cdot)\right) \quad (9)$$

where $P^{(t)}(\cdot)$ denotes the probability matrix from (8), which leads to the proposal of two LP approaches: considering the active spreaders **PD-A** and informed **PD-I** probability matrices. $F(\cdot)$ is the symmetrization function between a pair of nodes (i, j). Traditionally, authors consider F as the sum of the values $P^{(t)}_{i,j}(\cdot)$ and $P^{(t)}_{j,i}(\cdot)$, as in random walk LP methods. Here, we explore making the sum (s) and the mean (m) of the probabilities as a symmetrization function in both the proposed LP approaches.

C. Progression-Diffusion Algorithm

In Algorithm 1,[1] we present the proposed generalized PD algorithm and note its parallelism and distribute processing nature, useful for larger networks. There are two main functions for calculating the LP score matrix. The global parameters of the algorithm are: γ, the propagation probability; NSTEPS, the maximum number of time steps for the propagation; and F, the function for symmetrizing the probability matrix P (Algorithm 1, line 1).

The GET-PD-SCORE is the main function that yields the score matrix. The array **rK** has two columns, which are the reciprocal of the degree distribution vector with the first column also multiplied by γ (Algorithm 1, line 5). Lines 5 and 6 can execute in a single command with just one inline *for*. In Lines 7–10, we calculate each row of the initial P matrix,

[1]The source code of the algorithm presented in this manuscript is available at https://github.com/didiervega/progressive-diffusion-link-prediction

Algorithm 1 Proposed PD Algorithm

```
1: γ ← 0.2,  μ ← 1.0,  nSteps ← max(ℓ), F                          ▷ global parameters
2: function GET-PROGRESSIVE-DIFFUSION-SCORE(G)
3:     N ← LEN(G[:])                                               ▷ the size or number of nodes
4:     P ← ZEROS((N,N))                                            ▷ initialize the diffusion probability matrix N × N
5:     rK[:,0] ← ARRAY([ γ / LEN(G[x]) for x ∈ G[:]])              ▷ G[x] is the id-list of neighbors of node x
6:     rK[:,1] ← ARRAY([ μ / LEN(G[x]) for x ∈ G[:]])              ▷ rK is a N × 2 array of reciprocal degree vector
7:     for all i ∈ G[:] do ⋆                                       ▷ G[:] is the list of all node ids
8:         state_i ← GET-PROGRESSIVE-DIFFUSION-ROW(G, i, rK, N)
9:         P[i,:] ← 1.0 - state_i['S']
10:    end for
11:    A ← G.ADJMATRIX( )
12:    P ← P - A - 2 * IDENTITY(N)                                 ▷ removing self-loops and previous edges
13:    return s^PD ← F(P)                                          ▷ the F function for symmetrization
14: end function

15: function GET-PROGRESSIVE-DIFFUSION-ROW(G, i, rK, N)
16:    t_1 ← 0
17:    state ← INIT-ROW-STATES(N, i)
18:    for all step ∈ RANGE(nSteps) do ⋆
19:        t_0 ← t_1                                               ▷ t_0 current and t_1 future state
20:        t_1 ← t_0 ⩒ 1                                           ▷ xor operation
21:        P_A ← state[t_0]['I']                                   ▷ probability of Active spreaders
22:        P_I ← P_A + state[t_0]['R']                             ▷ probability of Informed nodes
23:        negT[:,0] ← ARRAY([ ∏(1.0 - rK[G[x],0] * P_A[G[x]]) for x ∈ G[:]])   ▷ column 0 is the i⃗(t) vector
24:        negT[:,1] ← ARRAY([1.0 - ∑(rK[x,1] * P_I[G[x]]) for x ∈ G[:]])       ▷ column 1 is the r⃗(t) vector
25:        state[t_1]['I'] ← state[t_0]['I'] * negT[:,1] + state[t_0]['S'] * (1.0 - negT[:,0])
26:        state[t_1]['R'] ← state[t_0]['R'] + state[t_0]['I'] * (1.0 - negT[:,1])
27:        state[t_1]['S'] ← 1.0 - state[t_1]['I'] - state[t_1]['R']   ▷ calculating the states of the next step t_1
28:    end for
29:    return state[t_1]                                           ▷ probability states at the last step
30: end function
```

as described in (8). The main function finishes removing scores of self-loops and previous edges and symmetrizing P, at Lines 11–13. F, at line 13, can be either the sum ($P + P^T$) or the mean ($P/2.0 + P^T/2.0$) symmetrization function. At Line 9, we take the probabilities of nodes being informed by node i after NSTEPS ($\vec{\varphi}(i,t)$). However, the probabilities of the active spreader ($\vec{I}(i,t)$) could also be considered. The star symbol at Line 7 indicates a point in which the parallelization can be achieved, with a direct parallel_for or a pool of workers. In this way, each worker (i.e., threads or computers) can calculate the graph's fractions, distributing the processing load and memory footprint.

The GET-PD-ROW function denotes the proposed diffusion framework to calculate the rows of P (8). The algorithm's parameters i, the initial seed, and the global NSTEPS, γ, and $\mu = 1.0$ are specifically set for the LP algorithm. However, this function can serve general purposes, like adopting a set of initial seeds, employing heterogeneous values of γ and μ, and running until the end of the diffusion process.

At Line 17, we initialize the system's state with node i as a single initial spreader. Lines 21 and 22 show the probabilities of nodes to be active spreaders (P_A) or informed (P_I) at the current time quantum. The vector $\vec{negT}$ denotes the two columns array of the negative transition probabilities presented in (5) and (6) positioning in columns 0 and 1, respectively. Lines 23 and 24 can execute in a single command with just one inline *for*. At Lines 25–27, we update the system to the next state [see (3) and (4)], iterating until the maximum number of NSTEPS. At Line 18, we have another point that could be parallelized, depending on the computational requirements, the function's adaptation for a more general-purpose one, i.e., running until the end of the diffusion process.

D. Complexity Analysis

In terms of computational cost, our method considers the quasi-local/global information for each node. It is because the method does not need to iterate until the end of the dynamic but only needs to run a small number of steps. As a default value, we adopt NSTEPS equal to the network's diameter, which is a minimal value for most of the social networks. Given that T is equal or close to the network diameter (we can assume a constant value of $T = 15$), and the $\langle k \rangle$ is much smaller than N, due to the power-law degree distribution or ultrasmall world property in the case of larger real-world networks, the computational order of the proposed framework is $\approxeq \mathcal{O}(N^2)$. For more details about the complexity calculation, please consult the Supplementary Material-I accompanying this work. As a result, our method

TABLE II
TOPOLOGICAL PROPERTIES OF THE NETWORKS

Network	N	$\langle k \rangle$	**C**	$\langle k^2 \rangle$	CC	$\max(\ell)$
BA	10000	11.9	48.4	581.5	0.015	5
Email	1133	9.6	18.6	179.8	0.22	8
Hamsterster	2000	16.1	43.7	704.7	0.54	10
Facebook	4039	43.7	106.6	4656.1	0.60	8
Advogato	5054	15.6	82.8	1290.4	0.25	9
PGP	10680	4.55	18.8	85.9	0.266	24
Astrophysics	14845	16.1	45.4	732.3	0.66	14
GooglePlus	23613	3.3	377.2	1251.6	0.17	8
Caida	26475	4.03	280.2	1130.1	0.208	17

does not need the entire network for the score calculation; it has a low computational cost and can be easily parallelized to distribute the processing.

III. EXPERIMENTAL SETUP

The LP methods approach the problem by ranking the likelihood of each node pair $(i, j) \notin E$ to appear at a later time. We consider the network as the last observed state at some discrete time and include a proportion of new predicted edges, as a future network state, for a particular LP method. Therefore, the evolved network contains an increment of edges concerning its original state. We evaluate the network evolution considering fixed fractions of new recommended edges by the LP methods.

A. Data Sets

We analyze the spreading capacity and the prediction of new links in the evolution of the networks in nine data sets. We adopt the Barabási–Albert (BA) [25] as a baseline synthetic network model. This model is representative of network characteristics such as scale-free degree distribution, but it lacks important properties present in real-world networks, such as assortativity, clustering, and the community structure [1].

We adopt eight real-world publically available network data sets [26] (see Table II): Email [27], Hamsterster [26], Facebook [26], Advogato [28], PGP [29], Astrophysics [30], GooglePlus [31], and Caida [32]. In all cases, we considered the undirected and unweighted main component for the simulations. Table II summarizes the topological characteristics of these networks. We consider the measures: number of nodes (N), average degree ($\langle k \rangle$), network complexity (**C**), second moment of degree distribution ($\langle k^2 \rangle$), average clustering coefficient (CC), and the diameter ($\max(\ell)$).

B. Link Prediction Methods

LP methods seek to learn a scoring function $s_{i,j} : E \mapsto \Re$, which is a link similarity score matrix $s^{N \times N}$. Each s_{ij} indicates the predicted likelihood of an edge between node i and j. Then, they recommend potential new links based on the higher scores. If the LP method produces scores based on the network topology, it is known as a structural similarity information approach [5]. In addition, the methods can employ local or global structural information [5], [6].

In this work, we adopt nine methods of LP as representative and classical approaches recommended in the prior art [5], [33]. From local measures, we select Common Neighbors (CN), Jaccard Coefficient (JC), and Adamic Adar (AA)From global measures, we select the Rooted PageRank (RP), SimRank (SR), Graph Distance (GD), DeepWalk (DW) [15],[2] Node2vec (NV) [16],[3] and the Variational Graph Autoencoder (VGAE) method [34],[4] which is a convolutional graph neural network approach.

The above techniques represent the most well-known LP methods in the area. Moreover, these methods also represent the main strategies used in many other competing art techniques, including triangle, paths, embeddings, deep convolutional, or neighborhood optimization. In this way, we can analyze the influence of different strategies to increase the networks' links and the relation between the prediction of links and the diffusion capacity underlying network evolution. For the classical LP methods, we used the Python implementation from Networkx.[5]

C. Information Propagation Models

Spreading is a pervasive process in society, and several models have been developed to understand the propagation of ideas or information through social networks [12], [24]. The susceptible-infected-recovered (SIR) [12], [23], [35] is the typical approach employed in epidemic spreading, where a pathogen spreads from infected to susceptible users with a probability β, and the recovered individuals are those infected that spontaneously obtained immunity to the pathogen with probability μ [12], [35]. On the other hand, for information spreading [12], the independent cascade (IC) spreading approach assumes the diffusion process as a cascade of activation [24], [35], i.e., the subsequent activation of informed nodes.

We employ the SIR and IC numerical simulations for evaluating the increase, or decrease, of the information capacity in the LP evolved networks. We calculate the final fraction of informed/infected nodes at the end of the simulation for each node as the only seed and averaging over all the vertices. For more details on the numerical simulation setups, please consult the Supplementary Material II accompanying this work.

IV. RESULTS AND DISCUSSION

This section presents the experiments and results for different LP methods regarding the performance on spreading capacity improvement in the evolved networks and the performance in the link classification tasks. This evaluation is fundamental, given that LP methods can serve as tools for enhancing and predicting the growth of complex networks, e.g., in social networks by satisfying user's connectivity preference and improving the information diffusion as the network evolves [23]. Thus, it is relevant to understand the implications of how the evolution of adding new edges affects the network diffusion capacity and how suitable the LP methods are to model network evolution.

[2]https://pypi.org/project/deepwalk/
[3]https://github.com/aditya-grover/node2vec
[4]https://github.com/lucashu1/link-prediction
[5]https://github.com/networkx

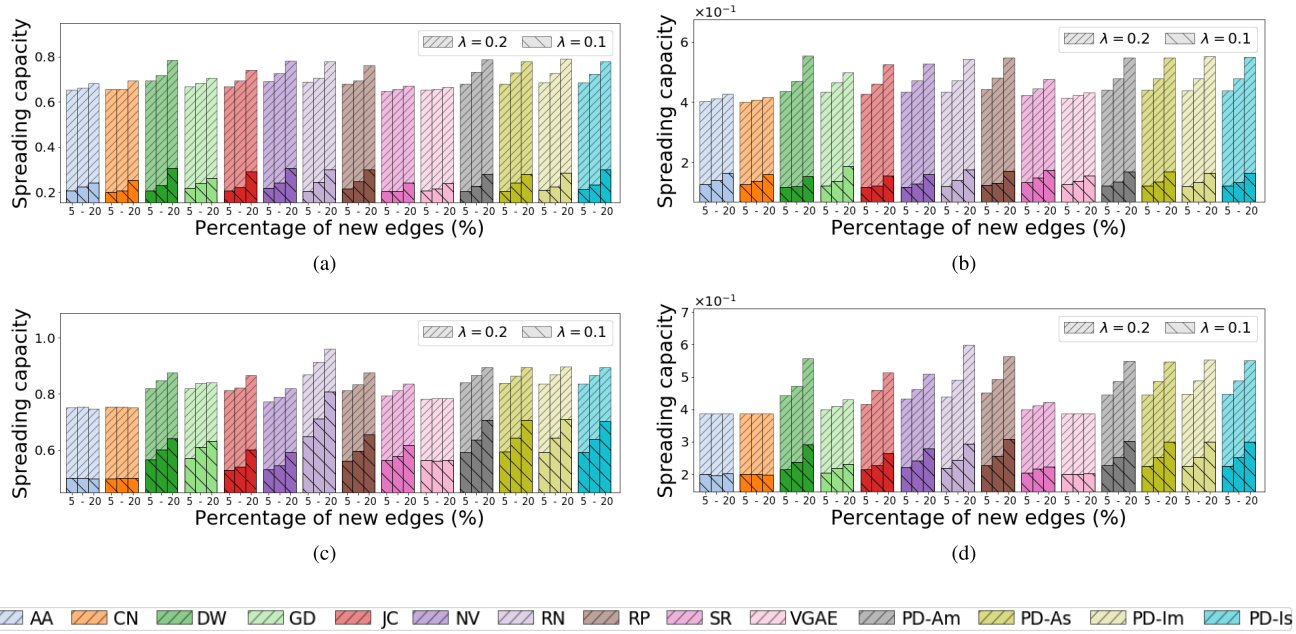


Fig. 2. Effects on the network spreading capacity when adding new edges to the networks (5%, 10%, and 20% concerning the original number of edges). The spreading capacities are calculated based on the numerical simulations in the IC diffusion approach. The LP methods are Adamic Adar (AA), Common Neighbors (CN), DeepWalk (DW), Graph Distance (GD), Jaccard Coefficient (JC), Node2Vec (NV), Random Selection (RN), Rooted Pagerank (RP), SimRank (SR), and Variational GAE (VGAE). There are four options for the PD: using the sum (PD-Is) or the mean (PD-Im) symmetrization in the matrix of Informed nodes or the sum (PD-As) or the mean (PD-Am) of the matrix of Active spreaders. (a)–(d) Artificial Barabási-Albert network, Email, Facebook, and Advogato, data sets, respectively. The results for the SIR diffusion approach are in the Supplementary Material III. (a) BA-IC. (b) Email-IC. (c) Facebook-IC. (d) Advogato-IC.

A. Setup of Spreading Analysis

Initially, we generate evolved versions of the networks by adding a certain percentage of new predicted edges. The adopted percentages of new edges concerning the original network are 5%, 10%, and 20%. The evolved versions are produced by the following nine LP methods: Adamic Adar (AA), Common Neighbors (CN), DeepWalk (DW), Jaccard Coefficient (JC), Rooted PageRank (RP), SimRank (SR), Graph Distance (GD), Node2vec (NV), Variational GAE (VGAE), and the random addition of edges (RN). We also explore four versions of the proposed PD method by considering: PD-Is and PD-Ia, which are the symmetrization by the sum ($s_{i,j}^{PD}(P^{(t)}(\varphi), sum)$) and mean ($s_{i,j}^{PD}(P^{(t)}(\varphi), \text{mean})$) of the informed probability matrix, respectively; PD-As and PD-Am, which are the respectively symmetrization cases ($s_{i,j}^{PD}(P^{(t)}(I), sum)$ and $s_{i,j}^{PD}(P^{(t)}(I), \text{mean})$) for the active-spreaders matrix. For the PD methods, the parameter NSTEPS is set to the network's diameter and $\gamma = 0.1$ when the epidemic threshold of the networks is lower or equal to 0.1, except for the PGP data set, whose epidemic threshold is equal to 0.21.

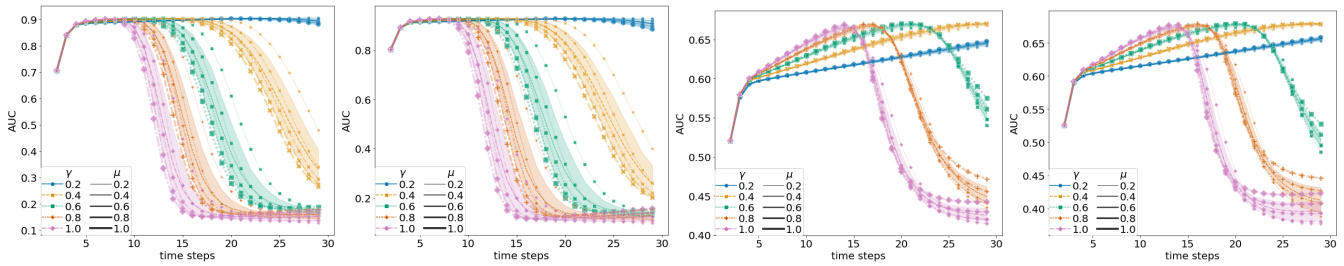
The spreading capacity of the network is the mean of the final fraction of informed individuals over all the nodes (please see the Supplementary Material II for more details). It is calculated on top of the SIR and IC MC simulations. Without loss of generality, for the SIR numerical simulations, we adopt the β/μ values as [0.2/0.8] and [0.4/0.8]; for the IC simulations, we consider a global spreading probability between the nodes, with $\beta_{ij} = \lambda = [0.1, 0.2]$. We notice that the results of the spreading capacity of the methods are consistent given different parameter combinations. Therefore, we calculate the spreading capacity in the original and evolved LP versions of the network according to the fraction of new edges, diffusion models, and parameters. After that, we analyze the features of each LP method considering the spreading capacities among the new networks and the different data sets. Notice that this is an exhaustive analysis, simulating over three evolved versions for each network from Table II times 14 LP methods, concerning the two diffusion models and the two combinations of diffusion parameters.

B. Results of the Spreading Analysis

Fig. 2 shows the diffusion models' results for the BA, Email, Facebook, Advogato, and PGP data sets. The methods' behavior is similar in the remaining networks and the SIR diffusion approach, which are reported in the Supplementary Material III accompanying this article.

For the BA network, the spreading capacities have a growing tendency as the percentage of new edges increases, which is expected. However, such behavior is less clear for the CN and AA methods. The increasing behavior is evident for the lowest λ values in both diffusion models. This result indicates that the diffusion dynamics with higher λ values reach the saturation of spreaders, producing a similar spreading outbreak.

For the real-world data sets, AA and CN methods present smaller improvements in the networks' spreading capacity when adding new edges, contrary to what we expected. This pattern has little changes with different diffusion models, and JC is also less prone to improve the spreading in the IC model (see Fig. 2 and the Supplementary Material III). The



(a) (b) (c) (d)

Fig. 3. AUC classification results of the parameter evaluation considering the peak of infected (active nodes) from the PD-As matrix. The evaluation results for the matrix of informed individuals (infected + recovered nodes) are in Supplementary Material IV. (a) Email—40% test set. (b) Email—20% test set. (c) BA—40% test set. (d) BA—20% test set.

other LP methods present a similar increasing pattern in both diffusion models although the reached values are different. We note that the random method, which ignores connection patterns between the nodes, always improves in both models the network's spreading capacity in the presence of new edges, which defines the expected behavior. Similarly, the proposed PD strategies lead to improvements in the spreading capacity, sometimes the highest, when the network connections grow.

C. Classification Analysis

We also evaluate the LP methods by the link classification task. We vary the test set in 20% and 40% of the total number of edges. The percentage of links is hidden in each case, and the LP methods need to predict them. Initially, we analyze the proposed approaches in terms of average AUC and precision (APR), varying the parameters γ, μ, and the number of time steps (NSTEPS). We have the parameter sensitivity given the proposed PD-As approach for the Email and BA networks in Fig. 3. The AUC results are consistent with the selection of γ and μ. Changes in μ do not show significant differences in the AUC when fixing some γ values initially. For this reason, and without loss of generality, we adopt $\mu = 1.0$.

We can observe that the peak of maximum AUC is affected by choice of γ; the smaller γ, the more steps to reach the peak. Thus, with a short exploration until 15-time steps and a $\gamma \geq 0.6$, it is possible to find the best parameter setup. We also notice that these results are independent of the fraction of the test set. The results for the PD-I approach are equivalent to PD-A and reported in Supplementary Material IV.

Table III presents the AUC and APR for the nine classic and recent LP methods (AA, RP, CN, SR, GD, JC, NV, VGAE, and DW) and random walk baseline (RN). The VGAE was trained with 90% of the training data and validated with 10% of the training data. We adopted the default hyperparameters in the codes of the methods.

We compare the prior art methods with the proposed PD-Is and PD-As approaches to verify the possible difference between adopting the probability matrix of informed (PD-Is) or the matrix of active spreaders (PD-As). We carried out this evaluation in five data sets (BA, Email, Facebook, Advogato, and PGP), which have different topological characteristics (see Table II for more details). Our approaches outperformed all methods regardless of the adopted data set (bold numbers in the table). This result highlights our methods' potential and suggests that the information spreading on the network is a critical factor in the LP task.

D. Statistical Analysis

We carried out a nonparametric statistical test on the spreading capacity and LP classification task of the LP methods to check the ranking and possible significant differences among them. We considered the Friedman–Nemenyi tests [36], grouping the LP methods by diffusion models, edge increasing, and AUC performance. In all tests, we considered the statistics at 95 percentile. According to the Friedman test on the spreading capacities in the IC, SIR simulations, and the LP classification results, the null hypothesis that all methods behave similarly should be rejected.

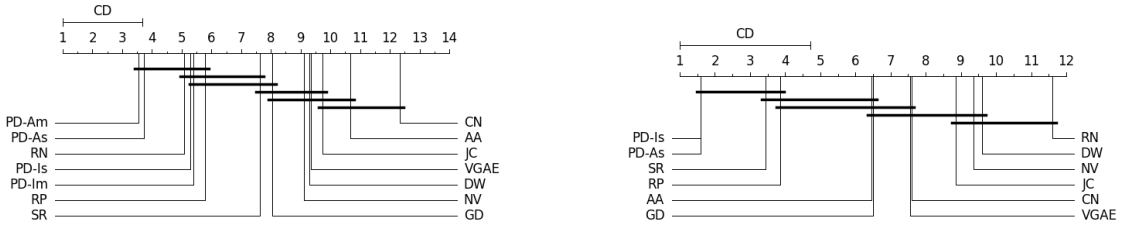
To visualize the difference among the methods, we execute the Nemenyi post hoc test plotting a diagram in which the critical difference (CD) is at the top. Besides, the LP methods' average ranks are plotted, where the lowest (best) positions are on the left side. If a set of methods have no significant difference, i.e., mean-ranking differences are below the CD value, a black line connects them.

For the Nemenyi post hoc test of the IC simulations, the CD between average ranks of two different LP methods is 2.70 [see Fig. 4(a)]. For the classification results, the CD between average ranks at the same percentile is 3.73 [see Fig. 4(b)]. More details about the Friedman and Nemenyi test and the post hoc test for the SIR simulations are present in Supplementary Material V.

The proposed PD methods are the best ranked in both diffusion models (see Fig. 4 and Supplementary Material V). In addition, PD-A strategies have significant ranking differences with the SR and RP method in the SIR, which are another type of diffusion/random walk strategy but with a higher computational cost. Traditional LP methods have lower ranking positions in the spreading capacity than the random selection of nodes. In general, CN and AA are the worst ranked methods with significant differences. RP is the best ranked LP within the traditional methods, even better than DW and NV, which also employ random walks when generating the embeddings, and VGAE, a graph convolutional network encoder. This result contrasts with what is expected, given that DW, NV, and VGAE are the approaches considered state

TABLE III
RESULTS OF AVERAGE AUC AND PRECISION (APR) FOR THE EVALUATED LP METHODS

Method	BA		Email		Facebook		Advogato		PGP	
	AUC	APR	AUC	APR	AUC	APR	AUC	APR	AUC	APR
	Results for 20% of sample test									
AA	0.5260	0.5260	0.8052	0.8065	0.9937	0.9922	0.8815	0.8863	0.9181	0.9182
RP	0.6624	0.6700	0.9203	0.9235	0.9922	0.9873	0.9528	0.9523	0.9944	0.9950
CN	0.5259	0.5221	0.8037	0.7978	0.9927	0.9896	0.8778	0.8738	0.9178	0.9176
SR	0.6635	0.6883	0.9153	0.9212	0.9924	0.9900	0.9548	0.9571	0.9948	0.9957
NV	0.4779	0.4799	0.7454	0.7625	0.9658	0.9457	0.6658	0.7062	0.9288	0.9364
GD	0.5801	0.5528	0.8860	0.8509	0.9196	0.8615	0.8986	0.8509	0.9917	0.9905
RN	0.5033	0.5033	0.4857	0.4785	0.4971	0.5001	0.4924	0.4957	0.4979	0.4964
VGAE	0.5858	0.6051	0.7539	0.7660	0.9489	0.9480	0.8368	0.8651	0.9484	0.9617
JC	0.5246	0.5039	0.8016	0.7942	0.9905	0.9868	0.8595	0.8307	0.9174	0.9158
DW	0.4828	0.4809	0.7614	0.7772	0.9635	0.9434	0.6516	0.6677	0.8867	0.8935
PD-Is	0.6776	0.7053	**0.9293**	**0.9349**	**0.9951**	0.9933	**0.9614**	**0.9644**	0.9949	**0.9958**
PD-As	**0.6798**	**0.7057**	0.9282	0.9313	**0.9951**	**0.9934**	0.9606	0.9632	**0.9952**	0.9934
	Results for 40% of sample test									
AA	0.5207	0.5196	0.7073	0.7056	0.9897	0.9884	0.8087	0.8123	0.7791	0.7790
RP	0.6522	0.6597	0.8993	0.8949	0.9913	0.9868	0.9398	0.9354	0.9865	0.9879
CN	0.5207	0.5169	0.7069	0.6998	0.9881	0.9835	0.8057	0.7998	0.7790	0.7788
SR	0.6553	0.6810	0.8947	0.8965	0.9904	0.9879	0.9458	0.9469	0.9818	0.9854
NV	0.4757	0.4790	0.7051	0.7340	0.9628	0.9419	0.6272	0.6637	0.9382	0.9499
GD	0.5765	0.5519	0.8661	0.8302	0.9219	0.8651	0.8866	0.8404	0.9820	0.9825
RN	0.4996	0.4991	0.5042	0.5087	0.4993	0.4989	0.5001	0.4994	0.4997	0.4980
VGAE	0.6093	0.6382	0.7286	0.7235	0.9354	0.9321	0.7792	0.8187	0.8758	0.9034
JC	0.5199	0.5033	0.7058	0.6958	0.9800	0.9695	0.7947	0.7600	0.7789	0.7784
DW	0.4887	0.4843	0.7248	0.7422	0.9605	0.9391	0.6188	0.6379	0.8930	0.9095
PD-Is	0.6690	0.6956	**0.9093**	**0.9102**	**0.9968**	**0.9965**	**0.9710**	**0.9747**	0.9965	0.9955
PD-As	**0.6709**	**0.6981**	0.9069	0.9060	0.9967	0.9964	0.9692	0.9728	**0.9979**	**0.9965**



(a) (b)

Fig. 4. Friedman–Nemenyi statistical tests of the spreading capacity and performance on predicting unseen links for the LP methods. (a) Ranking diagram for the IC diffusion model. (b) Ranking diagram for the average AUC results. Methods with no significant difference are connected. The lower the ranking, the more to the left in the diagram, and the better the method. The statistical test for the SIR numerical simulations is in Supplementary Material V.

of the art in LP. It is worth noting that the spreading capacity improvement measure is introduced as an evaluation metric of LP methods.

E. Computational Performance Analysis

We also evaluate the proposed PD method's memory and processing footprint concerning the network's number of nodes (N). For this reason, we generate artificial BA networks with sizes $N = [1 \times 10^3, 5 \times 10^3, 1 \times 10^4, 5 \times 10^4, 8 \times 10^4]$. The other network's properties are the same as the BA network described in Table II.

We employ a dedicated machine with Ubuntu18, CPU frequency of 2.6 GHz, four cores and eight threads, and 32 GB of RAM. In this case, we did not use GPUs. In all the simulations, we fixed $\gamma = 0.2$, $\mu = 1.0$, and NSTEPS to 15 time steps, in order to reproduce the short exploration. All the codes of the prior art methods and the proposed PD are in Python 2.7. We optimize the PD code in memory and processing use. The source code for ALL methods will be freely available in a public repository of the first author upon acceptance of this work.

TABLE IV
MEMORY AND TIME SPENT WHEN EXECUTING THE PD (PD-AS) METHOD IN DIFFERENT NETWORK SIZES N

N	Execution time (sec)		Memory use (MiB)	
	Single-core	Multi-core	Single-core	Multi-core
1×10^3	1.34	0.75	8.6	8.7
5×10^3	18.75	6.91	101.4	101.7
1×10^4	62.45	19.81	391.4	391.5
5×10^4	2169.7	974.58	9580.7	9580.8
8×10^4	5768.1	2345.2	24483.9	24486.1

* Multi-core execution was in four cores with a total of eight threads.

Table IV shows the results of memory and time spent when running the PD-As method on each of the artificial network

TABLE V
PERFORMANCE SUMMARY OF THE LP METHODS

	PD-As	RP	SR	GD	AA	VGAE	CN	JC	NV	DW
Reported computational cost	$\mathcal{O}(N^2)$	$\mathcal{O}(\mathbf{N^3})$	$\mathcal{O}(N^3)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N\langle k\rangle^2)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N\langle k\rangle^2)$	$\mathcal{O}(N\langle k\rangle^2)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$
Spreading capacity	+++	+++	++	+	∘	+	∘	+	+	+
Link prediction AUC ranking	1.6	3.85	3.45	6.5	6.45	7.55	7.6	8.85	9.35	9.6
Execution time (sec) in the Advogato network	22.87	12229.18	2288.48	6436.11	1359.11	237.05*	1230.52	1468.50	1399.66	2164.23
Memory (MiB) in the Advogato network	103.2	257.1	623.4	561.7	255.5	582.8*	254.3	221.8	447.3	269.8

+++ High increase tendency ++ Medium increase tendency + Low increase tendency ∘ ultra-low increase pattern

* The VGAE was the only method that ran in parallel processing.

sizes, grouped by a single-core or multicore execution. It is worth noting that this computational cost is the same independently of the selected PD option. We can also see that memory use is not impacted when PD switch to a multicore scenario, e.g., for a network of size $N = 1 \times 10^4$, the PD execution in a single-core or multicore execution used around 391.5 MiB. In terms of consuming time, the multicore execution is faster than the single-core case, as expected, e.g., for the same network of $N = 1 \times 10^4$, it spent around 20 s compared to 63 s in the single-core scenario. The higher the number of available cores/threads for processing big networks, the faster the parallel execution.

F. Effects of Edge Addition by the LP Methods

Table V summarizes the key results obtained in the spreading analysis and classification task, along with the time complexity of each LP method. We can observe the expected behavior in the case of random inclusion of links (RN), for which the spreading capacity grows as new links are included (see Fig. 2). However, contrary to the expected behavior, including recommended edges for some LP methods will not necessarily improve the network's spreading capacity. The local LP methods CN, AA, and JC seem to increase the evolved networks' spreading capacity slightly. AA and CN are both methods that consider the common neighbors between pairs of nodes, i.e., equivalent to adopt the number of steps equal to 1 in the PD method. Most of the time, RP has an increasing pattern in the spreading capacity when adding new edges, although it needs global information.

Improving the spreading capacity and link classification task at the same time is not a simple task. For instance, although RN is one of the best methods in improving the spreading capacity, the random inclusion of edges has a lower performance in the classification task, as expected. In particular, the proposed PD-A strategies are in the top places in the statistical test rankings (see Fig. 4). This is an interesting result for traditional spreading models and random walk LP methods, for which the works consider the final probability of informed individuals and not the probability of being an active spreader at some intermediate time.

PD-As and PD-Is are also the best ranked methods in the LP classification results [see Fig. 4(b)]. In the same group of top methods, we also have RP and SR strategies. More importantly, these methods are the ones that improve the diffusion capacity of the network when adding the recommended links. This result provides evidence that improving the diffusion of the network can lead to good performance in both the spreading capacity and classification task. In this way, the spreading capacity serves as a relevant metric to be considered.

Concerning the computational cost of the methods, PD considers an intermediate or quasi-local/global scale of the network, where each node iterates until a small number of steps, like the network's diameter, for instances. On the other hand, RP is a global method using a random walk strategy over all nodes. Both methods improve the spreading capacity by adding recommended edges. However, the PD method achieves better spreading and link classification results and is less time-consuming than RP. For example, notice that our PD method spent around 23 s, whereas RP spent 500 times longer in the Advogato network (see the comparison in Table V). Besides, the PD is the method that requires less memory allocation between the evaluated methods. Only the VGAE method used all the available threads in the machine due to the tensor-flow prerogatives during the training stage.

V. CONCLUSION

In this work, we have proposed a PD LP method capable of improving the information diffusion in a network with a low computational cost and practical use for real-world networks (quasi-local/global information, direct parallelization, and amenable to distributed processing). We have analyzed how the network evolution by the addition of new edges affects information spreading. In the numerical study, we considered a diverse set of data sets and two diffusion approaches, the epidemic SIR and information IC models, with different combinations of parameters. As evolutionary rules, we adopted the recommended edges from nine prior art LP methods, based on local and global structural information.

Through this study, we observe an impact of link recommendation on the diffusion process and vice versa. For example, in a real-world scenario, new links change the network's structure and, in turn, the interaction and diffusion process can change back the connections and speed up the network evolution. Therefore, it is relevant to evaluate the impact of LP methods on the network's spreading capacity. This work represents an effort in this direction by analyzing the spreading capacity and link classification results of some state-of-the-art methods and proposing a new LP approach with these concerns from the ground-up.

For some methods, the results indicate that the inclusion of new edges on the networks may not improve the spreading capacity, which is counterintuitive. As examples, we have observed that CN and AA methods have little impact on the

evolved networks' spreading capacity. Some other classical methods (e.g., RP, GD, and SR) tend to increase the network's spreading capacity when adding new links. At the same time, these methods also obtain good performance in the LP classification task. Such findings suggest that LP methods that improve the network's spreading capacity can lead to better classification performance. In this sense, the proposed PD method outperforms the spreading capacity of all methods under comparison, for the adopted data sets. More specifically, we have found that using the probability matrix of active spreaders (also known as the infected state) leads to satisfactory results with fewer iterations than the traditional arrival/recovered probability matrix, which requires more time steps to finish the simulation. The proposed dynamical process and the topological network structures can provide a better understanding of LP research.

Future work possibilities touch upon exploring information flow transmission among heterogeneous, multigraph networks, or time-varying strategies. Moreover, other diffusion models to improve transmission efficiency could also be developed.

REFERENCES

[1] M. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford Univ. Press, 2010.

[2] L. Berton, A. de Andrade Lopes, and D. A. Vega-Oliveros, "A comparison of graph construction methods for semi-supervised learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[3] D. A. Vega-Oliveros, P. S. Gomes, E. E. Milios, and L. Berton, "A multi-centrality index for graph-based keyword extraction," *Inf. Process. Manage.*, vol. 56, no. 6, Nov. 2019, Art. no. 102063.

[4] Y. Zou, R. V. Donner, N. Marwan, J. F. Donges, and J. Kurths, "Complex network approaches to nonlinear time series analysis," *Phys. Rep.*, vol. 787, pp. 1–97, Jan. 2019.

[5] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, Mar. 2011.

[6] Y. Yang, R. N. Lichtenwalter, and N. V. Chawla, "Evaluating link prediction methods," *Knowl. Inf. Syst.*, vol. 45, no. 3, pp. 751–782, Dec. 2015.

[7] L. Weng *et al.*, "The role of information diffusion in the evolution of social networks," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 356–364.

[8] E. Budur, S. Lee, and V. S. Kong, "Structural analysis of criminal network and predicting hidden links using machine learning," 2015, *arXiv:1507.05739*. [Online]. Available: https://arxiv.org/abs/1507.05739

[9] C. Lei and J. Ruan, "A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity," *Bioinformatics*, vol. 29, no. 3, pp. 355–364, Dec. 2012.

[10] J. Li, L. Zhang, F. Meng, and F. Li, "Recommendation algorithm based on link prediction and domain knowledge in retail transactions," *Procedia Comput. Sci.*, vol. 31, pp. 875–881, 2014.

[11] P. M. Chuan, L. H. Son, M. Ali, T. D. Khang, L. T. Huong, and N. Dey, "Link prediction in co-authorship networks based on hybrid content similarity metric," *Int. J. Speech Technol.*, vol. 48, no. 8, pp. 2470–2486, Aug. 2018.

[12] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Rev. Mod. Phys.*, vol. 87, no. 3, pp. 925–979, Aug. 2015.

[13] D. A. Vega-Oliveros, L. Berton, F. Vazquez, and F. A. Rodrigues, "The impact of social curiosity on information spreading on networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Jul. 2017, pp. 459–466.

[14] C. Dong, Y. Zhao, and Q. Zhang, "Assessing the influence of an individual event in complex fault spreading network based on dynamic uncertain causality graph," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1615–1630, Aug. 2016.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.

[16] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.

[17] A. F. Ally and N. Zhang, "Effects of rewiring strategies on information spreading in complex dynamic networks," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 57, pp. 97–110, Apr. 2018.

[18] D. Li, Y. Zhang, Z. Xu, D. Chu, and S. Li, "Exploiting information diffusion feature for link prediction in Sina Weibo," *Sci. Rep.*, vol. 6, no. 1, p. 20058, Apr. 2016.

[19] J. Wu, J. Shen, B. Zhou, X. Zhang, and B. Huang, "General link prediction with influential node identification," *Phys. A, Stat. Mech. Appl.*, vol. 523, pp. 996–1007, Jun. 2019.

[20] Z. Wang, Y. Lei, and W. Li, "Neighborhood attention networks with adversarial learning for link prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 24, 2020, doi: 10.1109/TNNLS.2020.3015896.

[21] R. R. Junuthula, K. S. Xu, and V. K. Devabhaktuni, "Evaluating link prediction accuracy in dynamic networks with added and removed edges," in *Proc. IEEE Int. Conf. Big Data Cloud Comput. (BDCloud), Social Comput. Netw. (SocialCom), Sustain. Comput. Commun. (SustainCom) (BDCloud-SocialCom-SustainCom)*, Oct. 2016, pp. 377–384.

[22] C. Hao Nguyen and H. Mamitsuka, "Latent feature kernels for link prediction on sparse graphs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1793–1804, Nov. 2012.

[23] D. A. Vega-Oliveros, L. Zhao, and L. Berton, "Evaluating link prediction by diffusion processes in dynamic networks," *Sci. Rep.*, vol. 9, no. 1, p. 10833, Dec. 2019.

[24] D. A. Vega-Oliveros, L. da Fontoura Costa, and F. A. Rodrigues, "Influence maximization by rumor spreading on correlated networks through community identification," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 83, Apr. 2020, Art. no. 105094.

[25] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[26] J. Kunegis. (Jan. 2018). *The Koblenz Network Collection–KONECT*. [Online]. Available: http://konect.uni-koblenz.de/networks/

[27] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, no. 6, p. 2003, Dec. 2003.

[28] P. Massa, M. Salvetti, and D. Tomasoni, "Bowling alone and trust decline in social network sites," in *Proc. 8th IEEE Int. Conf. Dependable, Autonomic Secure Comput.*, Dec. 2009, pp. 658–663.

[29] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, "Models of social networks based on social distance attachment," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 5, Nov. 2004, Art. no. 056122.

[30] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 2, pp. 404–409, Jan. 2001.

[31] J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 548–556.

[32] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: Densification laws, shrinking diameters and possible explanations," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2005, pp. 177–187.

[33] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. for Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.

[34] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*. [Online]. Available: https://arxiv.org/abs/1611.07308

[35] Z.-K. Zhang, C. Liu, X.-X. Zhan, X. Lu, C.-X. Zhang, and Y.-C. Zhang, "Dynamics of information diffusion and its applications on complex networks," *Phys. Rep.*, vol. 651, pp. 1–34, Sep. 2016.

[36] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.