RESEARCH Open Access

A novel regularized weighted estimation method for information diffusion prediction in social networks

Yoosof Mashayekhi[1], Alireza Rezvanian[2*] and S. Mehdi Vahidipour[3]

*Correspondence:
rezvanian@usc.ac.ir

[1] Computer Engineering Department, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
[2] Department of Computer Engineering, University of Science and Culture, Tehran, Iran
[3] Computer Engineering Department, Faculty of Electrical and Computer Engineering, University of Kashan, Kashan, Iran

Abstract

In recent years, social networks have become popular among Internet users, and various studies have been performed on the analysis of users' behavior in social networks. Information diffusion analysis is one of the leading fields in social network analysis. In this context, users are influenced by other users in the social network, such as their friends. User behavior is analyzed using several models designed for information diffusion modeling and prediction. In this paper, first, the problem of estimating the diffusion probabilities for the independent cascade model is studied. We propose a method for estimating diffusion probabilities. This method assigns a weight to each individual diffusion sample within a network. To account for the different effects of diffusion samples, several weighting schemes are proposed. Afterward, the proposed method is applied to real cascade datasets such as Twitter and Digg. We try to estimate diffusion probabilities for the independent cascade model considering the continuous time of nodes' infections. The results of our evaluation of our methods are presented based on several datasets. The results show the high performance of our methods in terms of training time as well as other metrics such as mean absolute error and F-measure.

Keywords: Independent cascade model, Information diffusion, Information diffusion prediction, Social network analysis

Introduction

In recent years, online social networks have become a major source of information. The emergence of the importance of social networks is the result of the large amount of data they produce. Users' actions are a major source of information in online social networks. A researcher analyzes data such as likes, shares, comments, and other interactions between social network users to find relationships between them. This information is used to understand users' behavior in social networks and their influence on each other. A number of problems are studied in this area, such as community detection, link prediction, and information diffusion. The study of information diffusion is mostly focused on how content spreads in social networks through interactions between users or external sources.

Millions of people use popular online social networks such as *Facebook*, *Twitter,* and *YouTube*. Such websites play a vital role in the propagation of information between internet users by providing the opportunity to view everyone's viewpoints about different subjects. The role of *Facebook* during the 2010 Arab Spring and *Twitter* during the 2016 U.S. presidential elections (Tan et al. 2013) shows the importance of online social networks in this era. In the analysis of information diffusion in social networks, it is assumed that content propagation is the result of users' interactions. In this regard, information diffusion prediction over social networks has become a priority for researchers. Several studies try to predict which users become aware of the propagating information in a social network. This field of study has many applications in other information diffusion-related research topics like rumor blocking (Wang et al. 2017), influence maximization (Khomami et al. 2018), and crowdsourcing. For example, when a rumor spreads about a company, company leaders can make early decisions to avoid damages in the future. On the other hand, understanding how content propagates more on a social network such as *Twitter* helps advertising find strategies. This results in more information spread in favor of a company (Gao et al. 2017).

The study of information diffusion in social networks was first initiated in epidemiology and social sciences contexts (Najar et al. 2012). In recent years, understanding user behavior in online social networks has become a major research topic. In this context, users influence each other through social relationships like following on *Twitter* or friendship on *Facebook*, resulting in information cascades (Bao et al. 2017). In the study of information diffusion, some research areas include detecting popular topics, identifying influential information spreaders, and modeling information diffusion (Patel and Nanavati 2023). It has been observed that many research studies in the context of information diffusion use independent cascade models (ICM) (Goldenberg et al. 2001) or linear threshold models (LTM) (Granovetter 1978) to analyze how information propagates in social networks. Information diffuses in these models through an iterative process. The probability of reaching information to a user depends on the neighbors, which already have propagated the information (Yang and Leskovec 2010; Gomez Rodriguez et al. 2011; Ver Steeg and Galstyan 2013). Various models have been proposed based on these two models. Some of them consider the continuous time of the diffusion process to be more realistic (Saito et al. 2009, 2010).

The parameters of each diffusion model play a significant role in the reflection of near-realistic modeling. Thus, it can be inferred from the history of users' activities on social networks, for example, by learning parameters of the diffusion model. The problem of learning diffusion probabilities for ICM has been first studied in Saito et al. (2008). Learning the parameters of a diffusion model requires observing users' interactions and behavior in the social network for a while. After learning the parameters, the model can be used in various information diffusion applications, for example, predicting which users will be infected in a diffusion process, maximizing influence propagation, etc. Most similar works, which try to learn diffusion probabilities for ICM, have poor training time performance over large networks. This is because they use iterative algorithms to learn diffusion probabilities. Therefore, we consider the reasonable sequence of activating nodes in such a way that a sequence of diffusions may occur from a node as a parent node to other connected nodes as child nodes. In this regard, we assign different weights as diffusion probabilities for each role of nodes as parents or children. To this end, this paper proposes a novel method for estimating diffusion probabilities for ICM. Moreover, we estimate ICM parameters by considering the continuous time of the diffusion process. The model can be useful for predicting information diffusion in social networks for practical purposes. Furthermore, we design a method to overcome the lack of learning data for many social network links. We extract some features from each link and use regression algorithms to learn a regularization term for each link's diffusion probability. Using this method, it is possible to learn the diffusion probabilities of links from fewer data. Our methods have lower training time and error than iterative methods.

In summary, our main contributions are as follows:

- We propose a basic method called **W**eighted **E**stimation of **D**iffusion **P**robabilities (WEDP) to estimate diffusion probabilities for ICM. In this method, we assign different weights to different types of information obtained from past cascades.
- We propose an improved version of WEDP, called **C**ontinuous **T**ime **W**eighted **E**stimation of **D**iffusion **P**robabilities (CT-WEDP), to learn diffusion probabilities from ICM considering the continuous time of the diffusion process. Similar to WEDP, different weights are obtained for different types of information obtained from cascades.
- We propose an improved version of CT-WEDP, called **C**ontinuous **T**ime **R**egularized **W**eighted **E**stimation of **D**iffusion **P**robabilities (CT-RWEDP), to learn diffusion probabilities from ICM considering the continuous time of the diffusion process and using regression algorithms to learn a regularization term for each link. This helps links with fewer available data to learn their diffusion probabilities.

The paper is organized as follows. Sect. "Related work" reviews previous works on this problem and their strengths and weaknesses. Sect. "Background" explains the background and preliminary requirements for the rest of the paper. Sect. "WEDP: weighted estimation of diffusion probabilities" describes the basic method of WEDP, including diffusion model description, problem definition, and evaluation of WEDP. Sect. "CT_WEDP: continuous time weighted estimation of diffusion probabilities" describes the proposed methods for continuous time. The experimental evaluation of the results of the information diffusion prediction task based on the proposed algorithms is given in Sect. "Experimental evaluation". Finally, Sect. "Conclusion and future work" concludes our work and gives some insights into future works.

Related work

Most social network analysis researchers use graphs to model social networks, where nodes represent users of social networks and edges represent relationships between users, such as friendship, following, etc. In recent years, many research works have studied information diffusion analysis (Yang and Leskovec 2010; Chang et al. 2018; Feng, et al. 2022; Kempe et al. 2003; Yu and Chu 2017; Failed 2016; Sharma and Bajaj 2023). Some researchers focus on influence maximization, which is the study of how to propagate content to reach more users in a social network. On the other hand, some of the studies try to minimize contamination spread in the social network. Other methods focus on learning a diffusion model or predicting information diffusion in an online social network.

Domingos and Richardson (Domingos and Richardson 2001; Richardson and Domingos 2002) were the first to study the spread of influence in social networks and identify the most influential nodes with a data mining approach. There are also some graph heuristics, such as node degree, closeness, etc., for identifying the most influential nodes in social networks. Kempe et al. (Kempe et al. 2003) formally defined the influence maximization problem as an optimization problem. They proved that the optimization problem of selecting the most influential nodes in ICM and LTM is NP-hard. They also designed a greedy method with approximation guarantees for the problem of selecting the most influential nodes in both ICM and LTM.

There have been several studies of information diffusion dynamics (Failed 2023; Rezvanian et al. 2023; Guille and Hacid 2012). A time-based asynchronous independent cascade model was proposed by Guille and Hacid (Guille and Hacid 2012). In this model, diffusion probabilities are a function of propagation time. They used Bayesian logistic regression to learn diffusion probabilities from features extracted from each *Twitter* link. Such studies learn a model based on users' extracted features instead of learning a diffusion probability for each link in a social network.

A number of research studies have studied the prediction of information diffusion as the problem of predicting which users will be infected at the end of the diffusion process. One approach to this problem is to learn the parameters of a diffusion model and use the model to predict information diffusion. Saito et al. (Saito et al. 2008) studied learning ICM diffusion probabilities. They formulated the problem as a likelihood function and used the EM algorithm for learning diffusion probabilities. Moreover, Goyal et al. (Goyal et al. 2010) investigated the learning parameters of a general threshold model. Their work is scalable, unlike Saito et al. (Saito et al. 2008), because their model's parameters are not learned iteratively. Some research has attempted to improve the performance of these two primary works. Continuous time models AsIC and AsLT have been developed to consider the continuous time of the diffusion process (Goldenberg et al. 2001; Granovetter 1978). Lamprier et al. (Lamprier et al. 2016) focused on the partial order of node infections instead of the exact timestamp that each node gets infected. Some other studies tackle the problem of information diffusion prediction by embedding users and contents in continuous latent space (Gao et al. 2017; Bourigault et al. 2016). In most cases, they don't consider the full social network graph, and the underlying network is not known to them.

Some other studies use the diffused content and users' profiles to enhance information diffusion prediction in social networks (Wang et al. 2020, 2016; Varshney et al. 2017; Barbieri et al. 2013). Barbieri et al. (Barbieri et al. 2013) proposed Topic-aware Independent Cascade and Topic-aware Linear Threshold models, which consider the topic of content propagated in learning models' parameters. In similar research, Wang et al. (Wang et al. 2016) used the emotion of the propagated content to learn diffusion probabilities for ICM.

In (Beni et al. 2023), Bouyer et. al., an improved version of the ICM is presented with an improved criterion for calculating the diffusion process is used. This is done by considering different diffusion rates for nodes in different layers with respect to core nodes.

It means that high-influence nodes are close to core nodes and low-influence nodes reside in outer core nodes. This concept is realized by applying the K-shell algorithm. Based on this model the authors provided candidate node selection and showed in simulation that the algorithm has linear time complexity.

A tensor-based independent cascade model is developed by incorporating the similarity matrix by Lin et al. (2023). The model considers the probabilities of information spreading from one node to its neighbors based on node interactions and similarity. The authors also propose an optimization algorithm to estimate the parameters and identify influential links (Lin et al. 2023).

In (Haldar et al. 2023), Haldar et. al., a novel temporal cascade model is introduced for understanding and analyzing information spreading dynamics in evolving networks. They aim to investigate how network structure and temporal patterns affect the spread of contagions. The proposed model takes into account both network topology and the temporal characteristics of interactions between individuals in the network. It considers the sequential nature of information spread, where individuals become infected and spread the contagion to their neighbors. Due to the critical role of learning diffusion probabilities for diffusion models in real-world scenarios, a stochastic diffusion model is proposed by Rezvanian et al. In this model, the authors take into account the probabilistic nature of information diffusion and incorporate the concept of network centrality as stochastic. They aimed to find a seed set of initial users who can trigger a cascading effect leading to maximum influence spread using learning automata theory (Rezvanian et al. 2023).

In this paper, we tackle the problem of learning diffusion probabilities for ICM and information diffusion prediction in social networks. Using information obtained from observed cascades in the social network, we design weighted estimation methods for this purpose. Additionally, we developed a method that learns a regularization term for diffusion probabilities based on some features of social network links. Our methods are more efficient than other methods for the same purpose.

Background

In this study, we learn the parameters of ICM. First, we explain this model. Next, we introduce the notation used in this paper and define the required preliminary.

ICM diffusion model

In the ICM model, a set of seed nodes is activated at time zero. The diffusion process proceeds discretely. At each time, every newly activated node at the previous time has a single chance to activate any of its inactive child nodes. Node v is the child of node u if there is an edge from u to v, and u is the parent of v. If the parent node fails to activate the child node, it will not have another chance to activate it again. When there is no newly activated node, the diffusion process ends. It is not possible to change the state of a node from active to inactive in ICM. In other words, they remain active after their activation time. A parent node activates a child node with a diffusion probability, which should be specified for the corresponding edge between them. Diffusion probabilities are ICM parameters. In this paper, we tackle the problem of estimating these probabilities.

Preliminaries

In this paper, we will follow (Saito et al. 2008)'s notation closely. We represent a social network (a directed network) with a graph $G = (V, E)$. Let V be the set of nodes and E be the set of edges of graph. We also represent each edge of the graph with (u, v), which means that there is a link from node u to node v. Next, we define set of out-neighbors and in-neighbors of node u. $N^{out}(u) = \{v|(u, v) \in E\}$ denotes the set of children of u and $N^{in}(u) = \{v|(v, u) \in E\}$ denotes the set of parents of u.

We define a diffusion episode as $D_s =< D(0), \ldots, D(T) >$, which is a sequence of node sets, where $D_s(t)$ denotes a set of nodes that are activated at the time t. Due to the assumption that nodes cannot change from active to inactive, a node cannot appear more than once in a diffusion episode. It means that nodes can be activated once in a single diffusion episode. In addition, $C_s(t)$ represents a set of nodes activated before time t in diffusion episode D_s. Moreover, $AT_s(u)$ denotes the activation time of node u in the diffusion episode D_s. $AT_s(u)$ will be infinity if u is not activated in D_s. Furthermore, $T(D_s)$ is the activation time of the last node in D_s.

WEDP: Weighted estimation of diffusion probabilities

In this section, we explain our basic method (called WEDP) to estimate diffusion probabilities for ICM, which is similar to our method presented in Mashayekhi et al. (2018). The problem definition is the same as in Saito et al. (2008). We are trying to answer the question: How can we estimate the diffusion probabilities for ICM, given a directed network graph $G = (V, E)$ and a set of diffusion episodes $S = \{D_0, \ldots, D_n\}$?. To answer the question, we will estimate the diffusion probability for an arbitrary edge $e = (u, v)$, denoted by P_{uv}^{WEDP}, using the diffusion episode set S. P_{uv}^{WEDP} is equal to the number of times u has succeeded in activating v divided by the number of times u has attempted to activate v. To estimate P_{uv}^{WEDP}, we use two functions whose inputs are an edge and a diffusion episode. These two functions are *isDiffused* and *weight*.

The function *isDiffused* indicates whether or not node u could have activated v in a diffusion episode D_s. Assume that node v is activated at time $t + 1$. If node u is activated at time t, *isDiffused* returns 1. There is a problem here. If multiple parents of node v are activated at time t, we do not know which parent activated v and which parent failed to activate v; *isdiffused* returns 1 for all of parents. To solve this problem, the second function, i.e. *weight*, is proposed.

The function *weight* calculates how much we can trust the episode D_s about the edge $e = (u, v)$. We assume that if node v is activated at time $t + 1$ and if more than one of its parents is activated at time t, then all of the parents of v with activation time t have successfully activated v. In computing diffusion probability for edge $e = (u, v)$, information obtained from such diffusion episodes will be weighted lower than episodes in which the parent node u has successfully or unsuccessfully activated the child node v.

These two functions are explained in the rest of this section.

Function *isDiffused*

To explain our method, first, we define *isDiffused* function, as shown in Pseudocode 1, which shows that in diffusion episode D_s and for edge $e = (u, v)$, whether or not u could have activated v in a diffusion episode.

```
Function $isDiffused((u, v), D_s)$
  Input: edge $e = (u, v)$, diffusion episode $D_s$.
  Output: 1 or 0
  if $AT_s(u) \leq T(D_s), AT_s(v) \leq T(D_s), AT_s(v) = AT_s(u) + 1$ then return (1)
  else return (0)
```

Pseudocode 1. Function *isDiffused*

In Pseudocode 1, if the parent node (i.e., node u) activates child node (i.e. node v), this function returns one. This means that:

(1) Parent node u must be activated in D_s earlier than the last active node in D_s. (i.e. $AT_s(u) \leq T(D_s)$),
(2) Child node v has the same condition (i.e., $AT_s(v) \leq T(D_s)$), and
(3) Child node v must be made active right after parent node u in D_s (i.e. $AT_s(v) = AT_s(u) + 1$).

If any of these three conditions is false, the *isDiffused* function returns zero.

Function *weight*

In this subsection, we define how to specify the weights of information obtained from a diffusion episode D_s for edge $e = (u, v)$. To this end, the *weight*$((u, v), D_s)$ function is defined in Pseudocode 2.

```
Function $weight((u, v), D_s)$
  Input: edge $e = (u, v)$, diffusion episode $D_s$.
  Output: a weight value for $e = (u, v)$
  #Activated_parents_of_$v$_in_$D_s$ =$|N^{in}(v) \cap D_s(AT_s(v) - 1)|$
  if $isDiffused((u, v), D_s)$ and #Activated_parents_of_$v$_in_$D_s > 1$ then
      return ($W$(#Activated_parents_of_$v$ ))
  elseif $isDiffused((u, v), D_s)$ and #Activated_parents_of_$v$_in_$D_s = 1$ then
      return (1)
  elseif $AT_s(u) \leq T(D_s)$ and $AT_s(v) > AT_s(u) + 1$ then
      return (1)
  else return (0)
```

Pseudocode 2: Function *weight*

Function *weight* will assign a weight to the information obtained from the diffusion episode D_s for the edge $e = (u, v)$.

- The function assigns weight to zero if u did not have any chance to activate v. This can occur if u is not activated at all or if it is activated after v.
- The function assigns weight to one if u had the opportunity to activate v and we are certain that it did not succeed. It can happen if u is activated at time t, and v is activated at time $t\prime > t + 1$, or it is not active at all.

- If u is activated at time t and v is activated at time $t+1$, two possible scenarios can occur:

  (1) If u is the only parent of v activated at time t (i.e., #Activated_parents_of_ v _in_ D_s=1), the function makes sure that u has activated v and assigns weight 1 for this information.

  (2) If there is more than one parent of v activated at time t, the function calls another function W to assign a weight for $e=(u,v)$. This new function is described in the next subsection.

Function W

In the following, we propose two weighting schemes to calculate W: linear decay and exponential decay.

- *Linear decay weighting*: In the linear decay weighting scheme, we define $\mathrm{W}(\mathrm{A})=\frac{1}{\mathrm{A}}$. In other words, suppose that node v is activated at the time-step $t+1$. Thus, as the number of parents of node v that may have activated it increases, information for the edges between them and v would receive lower weight.
- *Exponential decay weighting*: In the exponential decay weighting scheme, we define $\mathrm{W}(\mathrm{A})=\mathrm{e}^{-(\mathrm{A}-1)}$. Consequently, this weighting scheme defines lower weights than linear decay weighting when more than one parent of v could activate it.

Estimate diffusion probabilities for edges

Now we can compute diffusion probabilities for ICM. The diffusion probability of edge $e=(u,v)$ using a set of diffusion episodes $S=\{D_0,\ldots,D_n\}$ is computed by the following formula:

$$P_{uv}^{WEDP}=\frac{\sum_{D_s\in S} weight((uv).D_s)\times isDiffused((uv)D_s)}{\sum_{D_s\in S} weight((uv)D_s)} \tag{1}$$

In Eq. (1), the diffusion probability of edge $e=(u,v)$ is equal to the number of times $e=(u,v)$ may have activated v, divided by the number of times u had the chance to activate v, with applying weights defined beforehand.

To increase the robustness of estimating diffusion probabilities, we consider a special case. If there is no diffusion episode for an edge, we cannot be certain whether the parent node has succeeded or failed in activating the child node. Although we did not find a diffusion episode that shows the parent node activated the child node, the probability would be one in this case. We use a random probability instead of an estimated probability in this situation.

We refer to this method as *Weighted Estimation of Diffusion Probabilities (WEDP).*

Table 1 Activation times for two simple scenarios

$Scenario1, S = \{D_0, D_1\}$	$AT_0(u) = 1, AT_0(j) = 3, AT_0(v) = 2$
	$AT_1(u) = 1, AT_1(j) = 1, AT_1(v) = 2$
$Scenario2, S = \{D_0, D_1\}$	$AT_0(u) = 1, AT_0(j) = 2, AT_0(v) = 3$
	$AT_1(u) = 2, AT_1(j) = 1, AT_1(v) = 3$

Table 2 Calculation of diffusion probabilities for two scenarios

$Scenario1,$	$isDiffused((u, v), D_0) = 1, isDiffused((u, v), D_1) = 1$
	$weight((u, v), D_0) = 1, weight((u, v), D_1) = W(2) = \frac{1}{2}$
	$P_{uv}^{WEDP} = \frac{1\times1+\frac{1}{2}\times1}{1+\frac{1}{2}} = 1$
	$isDiffused((j, v), D_0) = 0, isDiffused((j, v), D_1) = 1$
	$weight((j, v), D_0) = 1, weight((j, v), D_1) = W(2) = \frac{1}{2}$
	$P_{jv}^{WEDP} = \frac{1\times0+\frac{1}{2}\times1}{1+\frac{1}{2}} = 1$
$Scenario2,$	$isDiffused((u, v), D_0) = 0, isDiffused((u, v), D_1) = 1$
	$weight((u, v), D_0) = 1, weight((u, v), D_1) = 1$
	$P_{uv}^{WEDP} = \frac{1\times0+1\times1}{1+1} = \frac{1}{2}$
	$isDiffused((j, v), D_0) = 1, isDiffused((j, v), D_1) = 0$
	$weight((j, v), D_0) = 1, weight((j, v), D_1) = 1$
	$P_{jv}^{WEDP} = \frac{1\times1+1\times0}{1+1} = \frac{1}{2}$

An example with two simple scenarios

In this subsection, we provide a toy example for diffusion probabilities estimation based on the WEDP. Consider$G = (V = \{u, v, j\}, E = \{(u, v), (j, v)\})$. We define two simple scenarios in each of which there are different sets of diffusion episodes$S = \{D_0, D_1\}$. The following Table 1 presents the activation time of each node in each diffusion episode in two scenarios. For example, $AT_0(u) = 1$ indicates that node u is activated at time 1 in the diffusion episodeD_0.

As given in Table 1, for Scenario 1, in D_1, multiple parents of node v are activated at time 1, while node v is activated at time 2. Thus, we need to consider different weights for the estimation of diffusion probabilities of edge (u, v) as P_{uv}^{WEDP} and edge (j, v) as P_{jv}^{WEDP}. On the other hand, in scenario 2, each parent of node v is activated in separate episodes. Therefore, function *isdiffused* returns different values for each edge to estimate diffusion probabilities.

The estimation of diffusion probabilities for two edges (u, v) and (j, v) are shown in Table 2. For calculation in function W the Linear Decay Weighting scheme is used. The calculations related to each scenario are shown in different rows. As presented in Table 2, for each scenario, the calculated diffusion probabilities for edges (u, v) and (j, v) are the same, whereas the returned values from *isdiffused* and *weight* functions are different.

Time complexity

WEDP's time complexity is discussed in this section. This is accomplished by calculating the time required to drive Eq. (2). In every diffusion episode, two functions are executed, *isdiffusion* and *weight*. Each function has a time complexity of $O(m)$, where m is the number of edges in the graph. For one diffusion episode, the time complexity is $O(2m)$. In Eq. (2), because there are n episodes in D_s, the time complexity is $O(2mn)$. Thus, the time complexity of the WEDP is $O(mn)$.

CT_WEDP: Continuous time weighted estimation of diffusion probabilities

Our proposed methods for predicting information diffusion for continuous time are described in this section. We are trying to answer the question: given a set of nodes in a social network as the initial nodes in a diffusion episode. Which users will be infected at the end of the diffusion episode?

One approach to answering this question is to learn a diffusion model and utilize it to simulate the diffusion episode. This will enable us to predict information diffusion. In this paper, we try to learn ICM parameters and use them for this purpose. One problem with the method explained in the previous section is that it does not consider continuous time in diffusion episodes. This will happen in real online social networks' diffusion episodes. Hence, we use the problem definition of the previous section, but we consider continuous time in diffusion episodes.

To overcome this problem, while estimating diffusion probabilities, if a parent node is active at time t, we consider that it can activate its child nodes at any time $t' > t$ instead of $t + 1$. Consequently, if a node is activated at time t, then any of its parents that were activated before this time may have activated it. To include continuous diffusion time, we redefine the *isDiffused* as *isDiffused_CT* and *weight* as *weight_CT* functions in Pseudocodes 3 and 4, respectively.

```
Function $isDiffused\_CT((u,v), D_s)$
  Input: edge $e = (u,v)$, diffusion episode $D_s$.
  Output: 1 or 0
if  $AT_s(u) \le T(D_s)$ and $AT_s(v) \le T(D_s)$ and$AT_s(v) > AT_s(u)$ then return(1)
  else  return(0)
```

Pseudocode 3: Function *isDiffused_CT*

```
FUNCTION $weight\_CT((u,v), D_s)$
  INPUT: EDGE $e = (u,v)$, DIFFUSION EPISODE $D_s$.
  OUTPUT: A WEIGHT VALUE FOR $e = (u,v)$
  #ACTIVATED_PARENTS_OF_$v$_IN_$D_s$ =$|N^{in}(v) \cap C_s(AT_s(v))|$
   IF  $isDiffused\_CT((u,v),D_s)$  AND #ACTIVATED_PARENTS_OF_$v$_IN_$D_s$ $> 1$ then
         return ($W$(#ACTIVATED_PARENTS_OF_$v$))
   elseif $isDiffused\_CT((u,v),D_s)$ AND #ACTIVATED_PARENTS_OF_$v$_IN_$D_s$ $= 1$ then
         return(1)
  elseif $AT_s(u) \le T(D_s)$ and $AT_s(v) > T(D_s)$ then
         return(1)
   else return(0)
```

Pseudocode 4: Function *weight_CT*

In Pseudocode 3, the function returns one if parent node u could have activated child node v. This will happen if v is activated after u. It returns zero otherwise. Pseudocode 4 is similar to Pseudocode 2 but uses the *isDiffused_CT* function instead of *isDiffused*.

Now, the diffusion probability for $e = (u, v)$ can be calculated by (2), similar to the formula proposed for WEDP. We refer to this method as Continuous Time Weighted Estimation of Diffusion Probabilities (CT-WEDP).

$$P_{uv}^{CT-WEDP} = \frac{\sum_{D_s \in S} weight\_CT((uv).D_s) isDiffused\_CT((uv)D_s)}{\sum_{D_s \in S} weight\_CT((uv)D_s)} \tag{2}$$

The sparsity of information diffusion data on real-world social networks

One of the problems with estimating diffusion probabilities in real-world social networks is the lack of learning data for most of the links in the network. More specifically, for many edges in the social network graph, it does not often happen that the parent node becomes active before its children. Additionally, most of the time, the number of a node's parents who have been activated before it is much smaller than the number of times a child is not activated at all or just one of its parents has activated it. Consequently, estimating diffusion probabilities for many edges is not as robust as others.

To get more insight, we collected tweets from a set of *Twitter* users between *January* 1^{st}, 2018, and *March* 19th, 2018. We considered hashtag propagation as diffusion episodes. Further, we considered edges in the graph when at least one parent node activated a child during a diffusion episode (the *isDiffused* function returns one in at least one diffusion episode). In the dataset, we have 6500 nodes, 512,144 edges, and 70,709 diffusion episodes.

Figure 1 represents the number of edges concerning their sum of weights. We used linear decay weighting in Fig. 1. We call the sum of weights of an edge the weight of the edge and refer to it as W_{uv}.

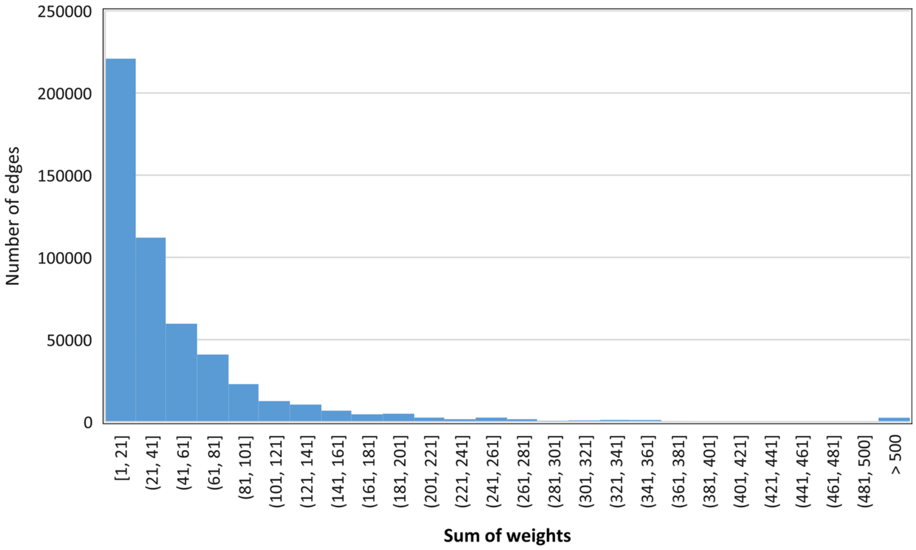

Fig. 1 Number of edges with respect to the sum of weights of edges

According to Fig. 1, the number of edges with high weights is less than edges with low weights. It shows that data available for estimating diffusion probabilities for many edges are rare. Hence, diffusion probabilities in this manner have considerable error.

Improving robustness

To learn diffusion probabilities, we improve the CT-WEDP. To this end, we use various social networking features to estimate the probability of the edges that have less data to learn by those edges that have more data to learn. Thus, the diffusion probability of the edge $e = (u, v)$ using a set of diffusion episodes $S = \{D_0, \ldots, D_n\}$ is computed by the following formula:

$$P_{uv}^{CT-RWEDP} = \lambda_{uv} P_{uv}^{CT-WEDP} + (1 - \lambda_{uv}) P_{uv}^{learn} \tag{3}$$

In (3), $P_{uv}^{CT-WEDP}$ is probability learned from CT-WEDP and P_{uv}^{learn} learn is probability learned from edges with more data. To learn this term, we use edges with more weights. λ_{uv} is a balancing factor. We refer to this method as *Continuous Time Regularized Weighted Estimation of Diffusion Probabilities (CT-RWEDP).*

We use the sum of the weights of the edge to compute λ_{uv}. We suggest a logarithm function as a concave function for λ_{uv}.

$$\lambda_{uv} = Min\left(1 \frac{Log(Max(W_{uv} 1))}{Log\left(Max\left(W_{AvgEstimate} 1+\right)\right)}\right) \tag{4}$$

In (4), W_{uv} is the sum of the weights of the edge (u, v) and $W_{AvgEstimate}$ is the average of W_{zw} for all edges which are used for estimating purposes. We use $1 + \epsilon$ in the denominator to ensure that it is greater than zero. These edges are $\alpha\%$ of all edges, with more W_{zw}.

Consider Eq. (4). If $W_{uv} < W_{AvgEstimate}$, the equation returns a value smaller than 1 for Lambda. It means we need to improve robustness. In this situation, the information obtained from the diffusion episode set for edge (u, v) is not enough, and more learning is needed. The logarithm function for λ_{uv} as a concave function simplifies calculations and means estimation curves downward. The logarithm function is commonly applied to many machine learning techniques such as Maximum Likelihood Estimation (MLE) (Failed 2006).

To learn P_{uv}^{learn}, we extracted some features from the edges of the graph. Next, we applied some regression algorithms (linear regression, decision tree, KNN regression, and SVR) to learn this term. We gained better performance utilizing SVR and used this algorithm in this paper. We used five features described below:

- *Child node activation rate*: this feature specifies how many times the child node has been activated in diffusion episodes; with respect to the times it had the chance to be activated (at least one of these parents should be activated in the diffusion episode). It is computed using the following formula:

$$Child\ Positive\ Rate_{uv} = \frac{\left|\left\{D_s \,|D_s \in S\, AT_s(v) \leq T(D_s)\right\}\right|}{\left|\left\{D_s \,|D_s \in S\, AT_s(v) \leq T(D_s)\ or\ \left|w \in N^{in}(v)\, AT_s(w) < T(D_s)\right| > 0\right\}\right|} \tag{5}$$

- *Parent node diffusion rate*: this feature specifies what fraction of children of the parent nodes have been activated if the parent node has been activated before them. Let $diffusionSet(u)$ be $\{D_s | D_s \in S, AT_s(u) \leq T(D_s), \left| w \in N^{out}(u), AT_s(u) < AT_s(w) \right| > 0\}$. The feature is computed using the following formula:

$$Parent\ Diffusion\ Rate_{uv} = \frac{\sum_{D_s \in diffusionSet(u)} \frac{\left| w \in N^{out}(u)\ AT_s(u) < AT_s(w) < T(D_s) \right|}{\left| w \in N^{out}(u)\ AT_s(u) < AT_s(w) \right|}}{\left| diffusion\ Set(u) \right|} \tag{6}$$

- *Parent node diffusion count*: this feature specifies how many children of the parent nodes have been activated if a parent node has been activated before them. It is computed using the following formula:

$$Parent\ Diffusion\ Count_{uv} = \frac{\sum_{D_s \in diffusionSet(u)} \left| w \in N^{out}(u)\ AT_s(u) < AT_s(w) < T(D_s) \right|}{\left| diffusion\ Set(u) \right|} \tag{7}$$

- *Jaccard similarity coefficient*: this feature specifies how many diffusion episodes both the parent and child nodes have been activated, based on at least one of them being activated. It is computed using the following formula:

$$Jaccard_{uv} = \frac{|\{D_s | D_s \in SAT_s(u) < T(D_s) AT_s(v) < T(D_s)\}|}{|\{D_s | D_s \in SAT_s(u) < T(D_s) or AT_s(v) < T(D_s)\}|} \tag{8}$$

- *Parent node out-degree*: This feature indicates the number of out-degree parent nodes (*i.e.*, the number of followers in a social network). It shows a node is popular among other nodes.

Experimental evaluation

Experiment 1

The purpose of experiment 1 is to evaluate the performance of WEDP. WEDP is compared with EM (Saito et al. 2008) in terms of mean absolute error (MAE), mean relative error (MRE), and training time. In (Saito et al. 2008), Saito et al. formulated the problem

Table 3 Statistics of datasets for Experiment 1

Dataset	***n***	***m***	***avg$_n$***	***max$_n$***	***min$_n$***
Digg	30,398	86,404	296.31	3139	1
Slashdot	51,083	131,175	1018.96	8183	1
UC Irvine	1899	20,296	484.95	896	1
CitHepTh	27,770	352,807	719.41	5513	1
DBLP	12,591	49,743	12.82	270	1
Cora	23,166	91,500	3.45	66	1
Google+	23,628	39,242	2.08	29	1
Twitter	465,017	834,797	3.70	100	1

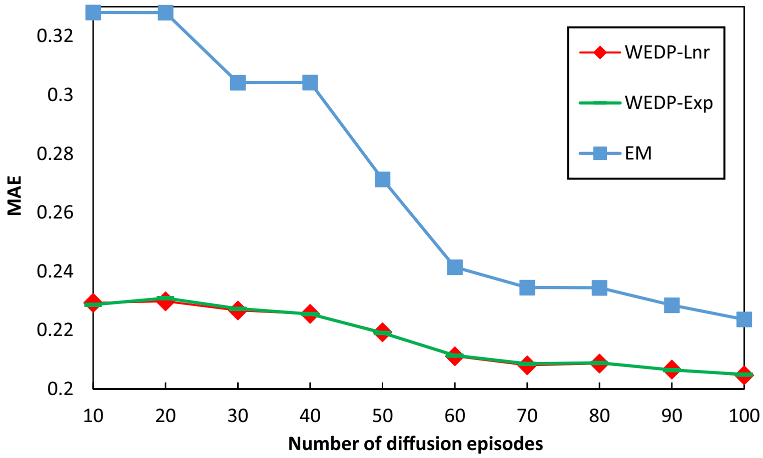

Fig. 2 Mean absolute error against the number of episodes in the Digg dataset

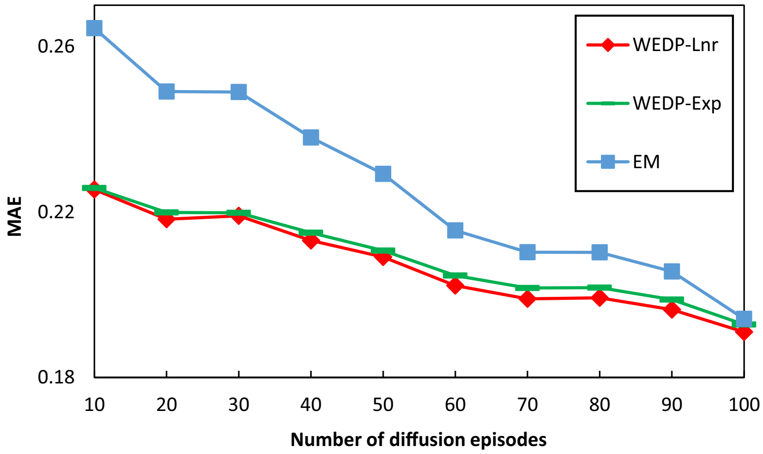

Fig. 3 Mean absolute error against the number of episodes in the Slashdot dataset

of learning diffusion probabilities from a set of diffusion episodes as a likelihood function and used the EM algorithm to optimize it.

We applied the approach presented in Saito et al. (2008) to choose diffusion probabilities and create diffusion episodes. For each dataset, we generated 100 different diffusion episodes with one uniformly randomly selected seed node for each episode. We tested the methods for different sizes of learning episodes. The diffusion probabilities are selected uniformly at random in the range [0.1, 0.3].

Table 3 lists the datasets used in the experiments. Self-edges are removed from datasets. Also, we considered the edges where the parent node activated before the child node in one diffusion episode. The datasets used in this experiment are Digg (Choudhury et al. 2009), Slashdot (Gómez et al. 2008), UC Irvine (Opsahl and Panzarasa 2009), CitHepTh (Leskovec et al. 2007), DBLP (Ley 2002), Cora (Šubelj and Bajec 2013), Google + (Leskovec and Mcauley 2012), and Twitter (Choudhury et al. 2010).

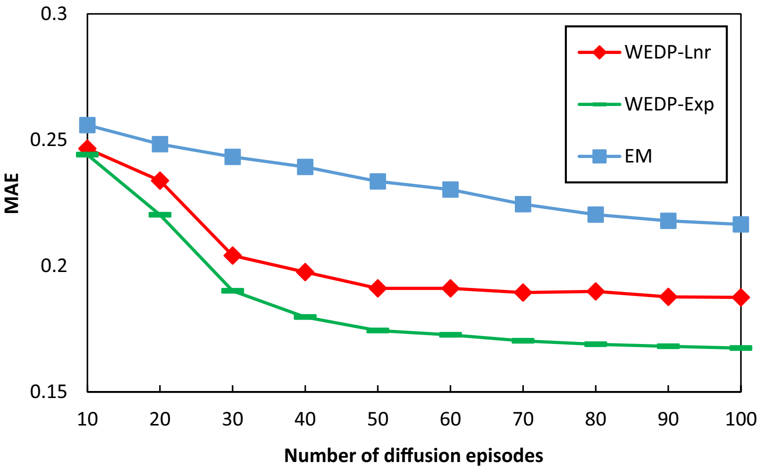

Fig. 4 Mean absolute error against the number of episodes in the UCIrvine dataset

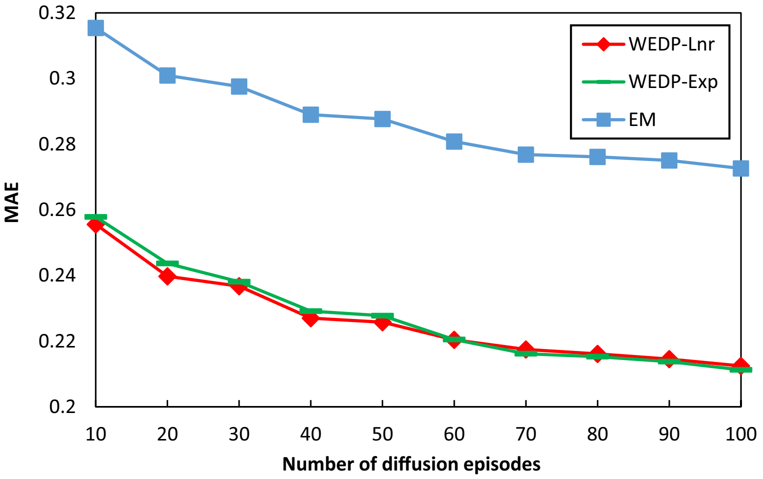

Fig. 5 Mean absolute error against the number of episodes in the CitHepTh dataset

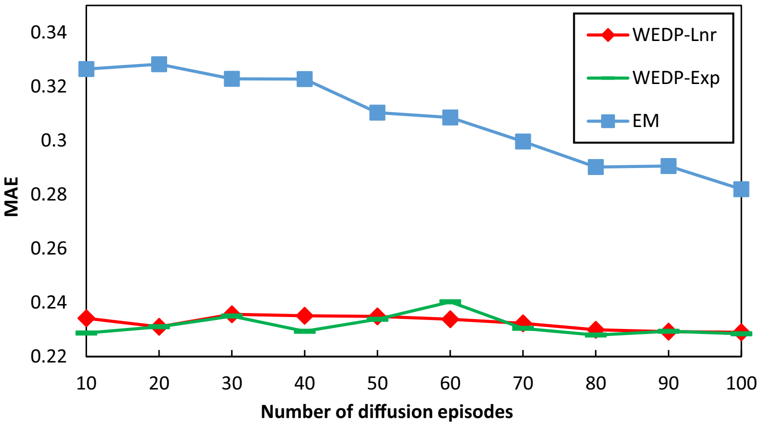

Fig. 6 Mean absolute error against the number of episodes in the DBLP dataset

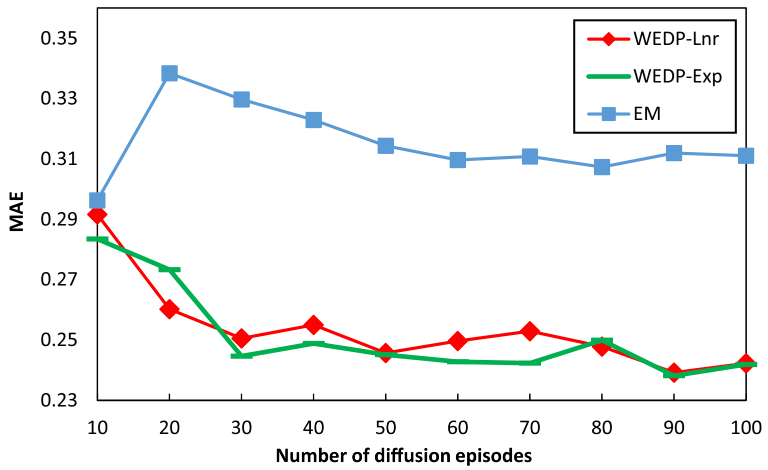

Fig. 7 Mean absolute error against the number of episodes in the Cora dataset

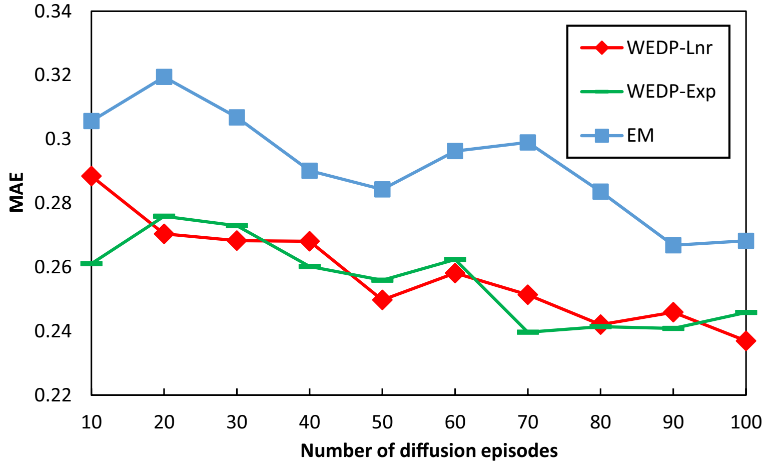

Fig. 8 Mean absolute error against the number of episodes in the Google + dataset

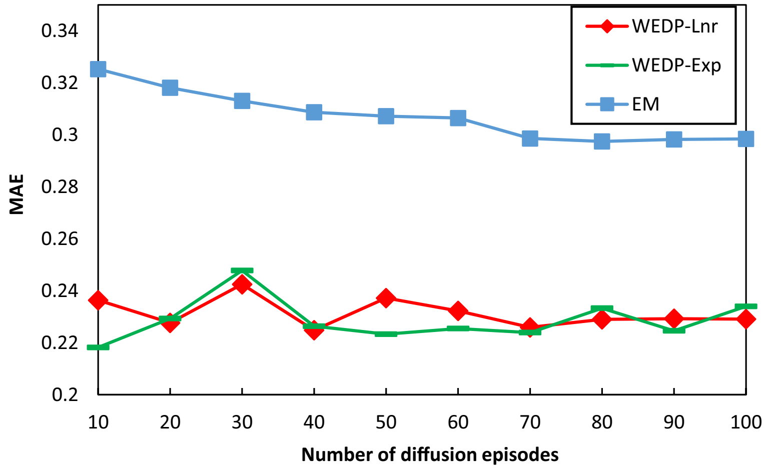

Fig. 9 Mean absolute error against the number of episodes in the Twitter dataset

Table 4 Mean relative error against the number of diffusion episodes in the Digg dataset

Method\S	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
WEDP-Lnr	1.181	**1.180**	**1.168**	1.161	1.123	**1.079**	**1.063**	1.066	1.054	1.044
WEDP-Exp	**1.177**	1.189	1.170	**1.160**	**1.122**	**1.079**	1.065	**1.065**	**1.051**	**1.043**
EM	1.690	1.689	1.580	1.580	1.412	1.262	1.226	1.226	1.195	1.169

Bold values show the best result (the lowest in each column)

Table 5 Mean relative error against the number of diffusion episodes in the Slashdot dataset

Method\S	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
WEDP-Lnr	1.174	**1.137**	**1.141**	**1.109**	**1.090**	**1.053**	**1.037**	**1.038**	**1.022**	0.994
WEDP-Exp	**1.171**	1.140	**1.141**	1.113	1.091	1.059	1.040	1.041	1.023	**0.990**
EM	1.397	1.320	1.320	1.260	1.214	1.143	1.115	1.114	1.09	1.031

Bold values show the best result (the lowest in each column)

Table 6 Mean relative error against the number of diffusion episodes in the UCIrvine dataset

Method\S	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
WEDP-Lnr	1.355	1.292	1.134	1.105	1.073	1.071	1.066	1.072	1.059	1.058
WEDP-Exp	**1.306**	**1.161**	**0.980**	**0.918**	**0.885**	**0.874**	**0.859**	**0.852**	**0.848**	**0.844**
EM	1.377	1.333	1.303	1.279	1.250	1.229	1.198	1.177	1.165	1.158

Bold values show the best result (the lowest in each column)

Table 7 Mean relative error against the number of diffusion episodes in the CitHepTh dataset

Method\S	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
WEDP-Lnr	**1.373**	**1.283**	1.267	1.216	1.206	1.180	1.167	1.160	1.152	1.139
WEDP-Exp	1.379	1.296	**1.262**	**1.210**	**1.201**	**1.158**	**1.133**	**1.129**	**1.119**	**1.104**
EM	1.685	1.603	1.586	1.536	1.528	1.492	1.474	1.469	1.463	1.449

Bold values show the best result (the lowest in each column)

Table 8 Mean relative error against the number of diffusion episodes in the DBLP dataset

Method\S	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
WEDP-Lnr	1.231	**1.206**	1.235	1.238	1.208	**1.205**	1.198	1.192	**1.185**	1.177
WEDP-Exp	**1.207**	1.207	**1.219**	**1.197**	**1.205**	1.233	**1.189**	**1.179**	1.188	**1.175**
EM	1.715	1.731	1.697	1.703	1.619	1.608	1.568	1.517	1.517	1.467

Bold values show the best result (the lowest in each column)

Table 9 Mean relative error against the number of diffusion episodes in the Cora dataset

Method\S	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
WEDP-Lnr	1.636	**1.36**	1.268	1.297	1.251	1.252	1.282	**1.251**	1.218	1.247
WEDP-Exp	**1.600**	1.422	**1.238**	**1.255**	**1.240**	**1.233**	**1.229**	1.264	**1.209**	**1.239**
EM	1.656	1.741	1.653	1.619	1.573	1.549	1.555	1.537	1.592	1.595

Bold values show the best result (the lowest in each column)

Table 10 Mean relative error against the number of diffusion episodes in the Google + dataset

Method\S	10	20	30	40	50	60	70	80	90	100
WEDP-Lnr	1.513	1.380	**1.345**	1.375	**1.286**	**1.346**	1.305	1.257	1.271	**1.218**
WEDP-Exp	**1.392**	**1.368**	1.367	**1.333**	1.328	1.397	**1.235**	**1.240**	**1.242**	1.266
EM	1.608	1.620	1.564	1.481	1.471	1.574	1.561	1.472	1.388	1.390

Bold values show the best result (the lowest in each column)

Table 11 Mean relative error against the number of diffusion episodes in the Twitter dataset

Method\S	10	20	30	40	50	60	70	80	90	100
WEDP-Lnr	1.250	**1.176**	**1.277**	**1.146**	1.206	1.180	1.141	**1.161**	1.159	**1.161**
WEDP-Exp	**1.147**	1.187	1.306	1.157	**1.132**	**1.147**	**1.125**	1.178	**1.131**	1.184
EM	1.715	1.658	1.647	1.564	1.553	1.551	1.503	1.497	1.503	1.505

Bold values show the best result (the lowest in each column)

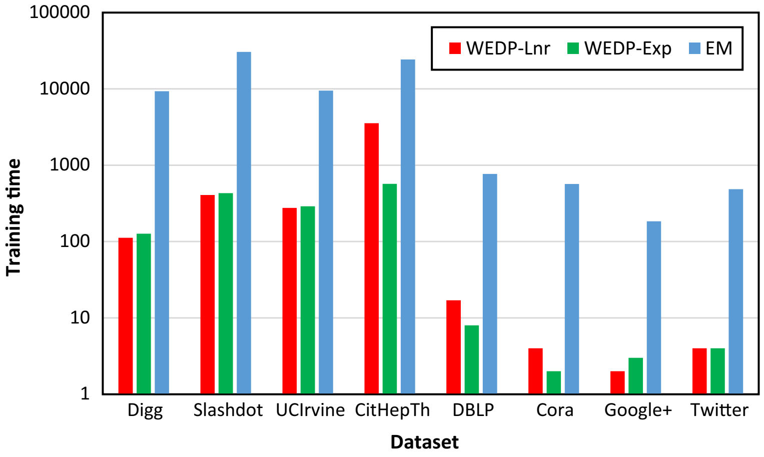

Fig. 10 Training time (Log-Scale) in each dataset

In Table 3, n and m show the number of nodes and edges in the network, respectively. Moreover, avg_n,max_n, and min_n represent the average, maximum, and minimum number of nodes activated in diffusion episodes.

In Figs. 2, 3, 4, 5, 6, 7, 8, and 9, MAE is plotted against the number of diffusion episodes for each dataset.

In Tables 4, 5, 6, 7, 8, 9, 10, and 11, MRE is shown against the number of diffusion episodes for different datasets. The number of diffusion episodes used for learning is S. Bold values show the best result.

As it is clear from the results, WEDP shows better performance in terms of both mean absolute error and mean relative error than EM. We also observe that by increasing the number of diffusion episodes, all methods result in less error in learning diffusion probabilities. Hence, in this setting, WEDP has better performance. Also, we observed that WEDP-Exp had less error than WEDP-Lnr in some datasets, demonstrating the importance of weighing lower the data obtained from diffusion episodes in which multiple parent nodes activated the child node.

Figure 10 shows the training time of methods in each dataset for 100 diffusion episodes.

As shown in Fig. 10, WEDP has less training time than EM, which is a significant improvement in learning diffusion probabilities. The reason is that EM uses an iterative algorithm to learn diffusion probabilities. On the other hand, WEDP estimates diffusion probabilities directly from diffusion episodes. Learning diffusion probabilities can be done through one scan of diffusion episodes. As a result, WEDP has fewer errors than EM, and its training time is much shorter than that of EM.

Experiment 2

In experiment 2, we evaluate CT-WEDP and CT-RWEDP performance. We compare them with the following methods:

- *EM*: Saito et al. (2008) formulated the problem of learning diffusion probabilities from a set of diffusion episodes as a likelihood function and used the EM algorithm to optimize it. We refer to this method as EM.
- *CTIC*: Saito et al. (2009) formulated the problem of learning diffusion probabilities considering the continuous time of nodes' activation times as a likelihood function. They used the EM algorithm to learn diffusion probabilities and time delays of node activations. We refer to this method as CTIC.
- *DAIC*: Lamprier et al. (2016) proposed a method similar to EM to learn diffusion probabilities for ICM by focusing on the partial order of nodes' activation times instead of exact activation times. They also considered exponential prior distributions of the diffusion probabilities to improve robustness. We refer to this method as DAIC.

We evaluate methods for the prediction of information diffusion. In a test diffusion episode, we must predict which nodes will be activated at the end. We use the training time of methods and F1-Score for evaluation, which is the harmonic average of precision and recall. Recall considers the ratio of nodes activated in a diffusion episode that has been retrieved as activated in the simulation. Precision is the ratio of correct activation predictions.

We use three datasets in this experiment. We considered the edges that, at least in one diffusion episode, the parent node activated before the child node:

Table 12 Statistics of Datasets for Experiment 2

Dataset	*n*	*m*	*c*	avg-n	max-n	min-n
Digg	36,031	338,741	3553	564.75	10,404	108
Twitter	6500	512,144	70,708	2.53	1390	1
MemeTracker	16,285	161,385	62,097	5.79	1269	1

Table 13 F-Score for each dataset

Method\dataset	Digg	Twitter	MemeTracker
CT-WEDP-Lnr	0.133	0.231	0.133
CT-WEDP-Exp	0.052	0.316	0.155
CT-RWEDP	**0.150**	**0.330**	**0.203**
EM	0.069	0.214	0.100
CTIC	0.040	0.248	0.049
DAIC	0.126	0.249	0.154

The best results (the lowest in each column) is highlithed as bold

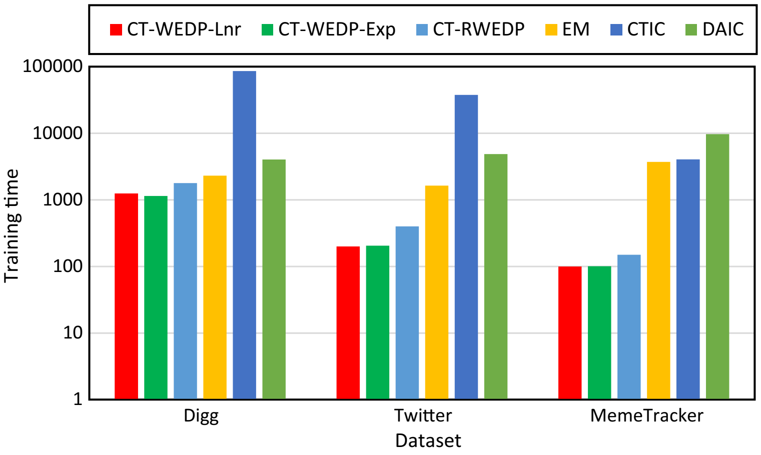

Fig. 11 Training time (Log-Scale) in each dataset

- The Digg news portal lets users vote on stories (articles, blog posts, videos, etc.). The dataset is described in Hogg and Lerman (2012). Voting on each story was considered a diffusion episode.
- *Twitter*: In this dataset, we collected tweets from a set of Twitter users between January 1st, 2018, and March 19th, 2018. We started with ten random users. Next, we took 1000 followers from the collected users, which had more followers within the collected users. We followed this approach 10 times. We considered each hashtag propagation as a diffusion episode.
- *MemeTracker*: The MemeTracker corpus has diffusion episodes of short phrases. This dataset is described in Leskovec et al. (2009). We considered this dataset between August 1st, 2008, and April 1st, 2009. We consider a link from blog A to blog B if blog B has at least one post having a post from blog A in it. In this way, we can conclude that blog B is watching blog A's posts, and hence, diffusion will occur from blog A to B.

Table 12 gives some statistics for experiment datasets.

In Table 12, all parameters are the same as in Table 2. Moreover, parameter c shows the number of diffusion episodes in each dataset.

Table 13 shows the result for the F1-Score for each dataset.

In all datasets, CT-RWEDP shows better performance. In this CT-RWEDP, we used CT-WEDP with the weighting scheme. In the Digg dataset, we employed CT-WEDP-Lnr in CT-RWEDP because it has better performance than other methods. In the Twitter and MemeTracker datasets, we employed CT-WEDP-Exp in CT-RWEDP because it has a higher F1-Score than other methods. These results show that CT-WEDP and CT-RWEDP are suitable methods for learning diffusion probabilities in real-world social networks. As a result, learning regularization terms with reliable diffusion probabilities can be helpful when learning data for some edges in the network is less than others.

Figure 11 shows the training time of methods for each dataset.

CT-WEDP and CT-RWEDP have less training time than EM-based methods (EM, CTIC, DAIC). The reason is that EM-based methods use the EM algorithm in their learning phase, which is an iterative algorithm. Thus, CT-WEDP and CT-RWEDP have less training time and a better F1-Score, which shows that they are appropriate for learning diffusion probabilities in real-world social networks.

Conclusion and future work

In this paper, we tackled the problem of learning diffusion probabilities for a well-known diffusion model, ICM. Next, we proposed a method to learn diffusion probabilities for ICM considering continuous cascade time. Consequently, it can be used in information diffusion prediction. We also used some features of social networks to learn diffusion probabilities from edges with more reliable data. Our methods learn diffusion probabilities from diffusion episodes available for use in learning. The experiments showed that our methods reduce error. As a result of the method proposed in this paper, probabilities are learned in much less time, making it more scalable.

Based on promising results with this approach, we can expect a variety of developments in the proposed method in the future. For instance, we intend to consider the dynamics of social networks. Users and the relationships between them change over time and it will be helpful to design methods that consider these changes. We can also use users' profiles and content, propagated to result in a model close to true users' behaviors.

Author contributions

YM designed, proposed the original idea, developed the code, designed, and performed the simulation experiments. YM, AR and SMV wrote the main manuscript text and guided the research process. All authors planned the work, analyzed the results, and reviewed the manuscript.

Funding

No funding.

Availability of data and materials

The datasets analyzed during the current study are available in the KONECT – The Koblenz Network Collection repository, http://konect.cc/networks/.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 25 July 2023 Accepted: 11 November 2023
Published online: 30 November 2023

References

Bao Q, Cheung WK, Zhang Y, Liu J (2017) A component-based diffusion model with structural diversity for social networks. IEEE Trans Cybern 47:1078–1089

Barbieri N, Bonchi F, Manco G (2013) Topic-aware social influence propagation models. Knowl Inf Syst 37:555–584

Beni HA, Bouyer A, Azimi S, Rouhi A, Arasteh B (2023) A fast module identification and filtering approach for influence maximization problem in social networks. Inf Sci 640:119105

Bishop CM, Nasrabadi NM (2006) Pattern recognition and machine learning. Springer, New York

Bourigault S, Lamprier S, Gallinari P (2016) Representation learning for information diffusion through social networks: an embedded cascade model. In: Proceedings of the Ninth ACM international conference on web search and data mining, pp 573–582

Chang B, Xu T, Liu Q, Chen EH (2018) Study on information diffusion analysis in social networks and its applications. Int J Autom Comput 15:377–401

De Choudhury M, Sundaram H, John A, Seligmann DD (2009) Social synchrony: predicting mimicry of user actions in online social media. In: 2009 International conference on computational science and engineering CSE'09, pp 151–158

De Choudhury M, Lin Y-R, Sundaram H, Candan KS, Xie L, Kelliher A (2010) How does the data sampling strategy impact the discovery of information diffusion in social media? ICWSM 10:34–41

Domingos P, Richardson M (2001) Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp 57–66

Zhang K, Yun X, Liang J, Zhang X-y, Li C, Tian B (2016) Retweeting behavior prediction using probabilistic matrix factorization. In: 2016 IEEE symposium on computers and communication (ISCC), pp 1185–1192

Feng S et al. (2022) H-Diffu: hyperbolic representations for information diffusion prediction. In: IEEE Trans Knowl Data Eng, pp 1–14

Gao S, Pang H, Gallinari P, Guo J, Kato N (2017) A novel embedding method for information diffusion prediction in social network big data. IEEE Trans Industr Inf 13:2097–2105

Goldenberg J, Libai B, Muller E (2001) Talk of the network: a complex systems look at the underlying process of word-of-mouth. Mark Lett 12:211–223

Gómez V, Kaltenbrunner A, López V (2008) Statistical analysis of the social network and discussion threads in Slashdot. In: Proceedings of the 17th international conference on World Wide Web, pp 645–654

Gomez Rodriguez M, Balduzzi D, Schölkopf B (2011) Uncovering the temporal dynamics of diffusion networks. In: Proceedings of the 28th international conference on machine learning (ICML- 11), ICML'11, ACM, 2011, pp 561–568

Goyal A, Bonchi F, Lakshmanan LV (2010) Learning influence probabilities in social networks. In: Proceedings of the third ACM international conference on Web search and data mining, pp 241–250

Granovetter M (1978) Threshold models of collective behavior. Am J Sociol 83:1420–1443

Guille A, Hacid H (2012) A predictive model for the temporal dynamics of information diffusion in online social networks. In: Proceedings of the 21st international conference on World Wide Web, pp 1145–1152

Haldar A, Wang S, Demirci GV, Oakley J, Ferhatosmanoglu H (2023) Temporal cascade model for analyzing spread in evolving networks. ACM Trans Spatial Algor Syst 9(2):1–30

Hogg T, Lerman K (2012) Social dynamics of digg. EPJ Data Sci 1:5

Kempe D, Kleinberg J, Tardos É (2003) Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 137–146

Khomami MMD, Rezvanian A, Bagherpour N, Meybodi MR (2018) Minimum positive influence dominating set and its application in influence maximization: a learning automata approach. Appl Intell 48:570–593

Lamprier S, Bourigault S, Gallinari P (2016) Influence learning for cascade diffusion models: focus on partial orders of infections. Soc Netw Anal Min 6:93

Leskovec J, Mcauley JJ (2012) Learning to discover social circles in ego networks. Adv Neural Inf Process Syst pp 539–547

Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: densification and shrinking diameters. ACM Trans Knowl Discov Data (TKDD) 1:2

Leskovec J, Backstrom L, Kleinberg J (2009) Meme-tracking and the dynamics of the news cycle. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 497–506

Ley M (20025) The DBLP computer science bibliography: evolution, research issues, perspectives. In: International symposium on string processing and information retrieval, pp 1–10

Lin W, Xu Q, Li Y, Xu L (2023) A tensor-based independent cascade model for finding influential links considering the similarity. Chaos Solitons Fract 173:113655

Lin Y, Wang X, Wang L, Wan P (2023) Dynamics modeling and optimal control for multi-information diffusion in Social Internet of Things. Digit Commun Netw. https://doi.org/10.1016/j.dcan.2023.02.014

Mashayekhi Y, Meybodi MR, Rezvanian A (2018) Weighted estimation of information diffusion probabilities for independent cascade model. In: 2018 4th International Conference on Web Research (ICWR), pp 63–69

Najar A, Denoyer L, Gallinari P (2012) Predicting information diffusion on social networks with partial knowledge. In: Proceedings of the 21st international conference on world wide web, pp 1197–1204

Opsahl T, Panzarasa P (2009) Clustering in weighted networks. Social Netw 31:155–163

Patel NA, Nanavati N (2023) A state of the art review on user behavioral issues in online social networks. Recent Adv Comput Sci Commun (Formerly: Recent Patents on Computer Science) 16(2):81–95

Rezvanian A, Vahidipour SM, Meybodi MR (2023) A new stochastic diffusion model for influence maximization in social networks. Sci Rep 13(1):6122

Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 61–70

Saito K, Nakano R, Kimura M (2008) Prediction of information diffusion probabilities for independent cascade model. In: Knowledge-based intelligent information and engineering systems, pp 67–75

Saito K, Kimura M, Ohara K, Motoda H (2009) Learning continuous-time information diffusion model for social behavioral data analysis. Adv Mach Learn, pp 322–337

Saito K, Kimura M, Ohara K, Motoda H (2010) Selecting information diffusion models over social networks for behavioral analysis. In: Joint European conference on machine learning and knowledge discovery in databases, pp 180–195

Sharma K, Bajaj M (2023) DeepWalk based influence maximization (DWIM): influence maximization using deep learning. Intell Autom Soft Comput 35(1)

Šubelj L, Bajec M (2013) Model of complex networks based on citation dynamics. In: Proceedings of the 22nd international conference on World Wide Web, pp 527–530

Tan W, Blake MB, Saleh I, Dustdar S (2013) Social-network-sourced big data analytics. IEEE Internet Comput 17:62–69

Varshney D, Kumar S, Gupta V (2017) Predicting information diffusion probabilities in social networks: a Bayesian networks based approach. Knowl-Based Syst 133:66–76

Ver Steeg G, Galstyan A (2013) Information-theoretic measures of influence based on content dynamics. In: Proceedings of the sixth ACM international conference on Web search and data mining, 2013, pp 3–12

Wang B, Chen G, Fu L, Song L, Wang X (2017) Drimux: dynamic rumor influence minimization with user experience in social networks. IEEE Trans Knowl Data Eng 29:2168–2181

Wang Z, Chen C, Li W (2020) Joint learning of user representation with diffusion sequence and network structure. IEEE Trans Knowl Data Eng 34(3):1275–1287

Wang Z, Zhao J, Xu K (2016) Emotion-based Independent Cascade model for information propagation in online social media. In: 2016 13th international conference on service systems and service management (ICSSSM), pp 1–6

Yang J, Leskovec J (2010) Modeling information diffusion in implicit networks. In: 2010 IEEE 10th international conference on data mining (ICDM), pp 599–608

Yu X, Chu T (2017) Learning the structure of influence diffusion in the independent cascade model. In: 2017 36th Chinese Control Conference (CCC), pp 5647–5651

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.