Online Social Networks and Media 5 (2018) 61–75

Contents lists available at ScienceDirect

Online Social Networks and Media

journal homepage: www.elsevier.com/locate/osnem

User-centered probabilistic models for content diffusion in the blogosphere

Cédric Lagnier[a,b,*], Eric Gaussier[a,b], François Kawala[c]

[a] University Grenoble Alpes, CNRS, Grenoble INP, LIG, France
[b] Skopai, France
[c] Société Karos, France

ARTICLE INFO

Article history:
Received 30 September 2017
Revised 23 January 2018
Accepted 23 January 2018

Keywords:
Information diffusion
Modelization
Social networks

ABSTRACT

Predicting the diffusion of information in social networks is a key problem for applications like Opinion Leader Detection, Buzz Detection or Viral Marketing. Many diffusion models are direct extensions of the *Cascade* and *Threshold* models, initially proposed for epidemiology and social studies. In such models, the diffusion process is based on the dynamics of interactions between neighbor nodes in the network (the social pressure), and largely ignores important dimensions as the content diffused and the active/passive role users tend to have in social networks. We propose here a new family of models that aims at predicting how a content diffuses in a network by making use of additional dimensions: the content diffused, user's profile and willingness to diffuse. In particular, we show how to integrate these dimensions into simple feature functions, and propose a probabilistic modeling to account for the diffusion process. These models are then illustrated and compared with other approaches on two blog datasets. The experimental results obtained on these datasets show that taking into account the content diffused is important to accurately model the diffusion process. Lastly, we study the influence maximization problem with these models and prove that it is NP-hard, prior to propose an adaptation of the greedy algorithm to approximate the optimal solution.

1. Introduction

Propagation models in content networks, i.e. social networks in which content are shared and diffused among users, aim at reproducing the diffusion of information between users. Being able to accurately model this diffusion has several practical applications, as the identification of influence hubs, the choice of initial diffusers for a maximal diffusion, or the identification of links one has to remove in order to limit the diffusion (e.g. for stopping rumors).

Most of the models proposed in the domain of information diffusion are extensions of the Independent Cascade model (IC) [1] and the Linear Threshold model (LT) [2]. The IC model is based on the following simple principle: as soon as a user (i.e. a node in the social network) n_j is infected, she has a unique chance to infect each of her direct neighbors n_i with a probability P_{ji} that depends on both n_j and n_i. The LT model considers that a node n_i of the social network (i.e. a user) is contaminated if the sum of the weights on its incoming edges are above a threshold θ_i specific to n_i, this threshold being chosen randomly in many instances of the model [3]. They nevertheless fail to take into account for two important elements:

- They ignore the content of the information diffused even though, in a given social network, two different pieces of information will not propagate in the same way;
- They tend to ignore users characteristics even though the interest of a particular user plays a major role in the diffusion process.

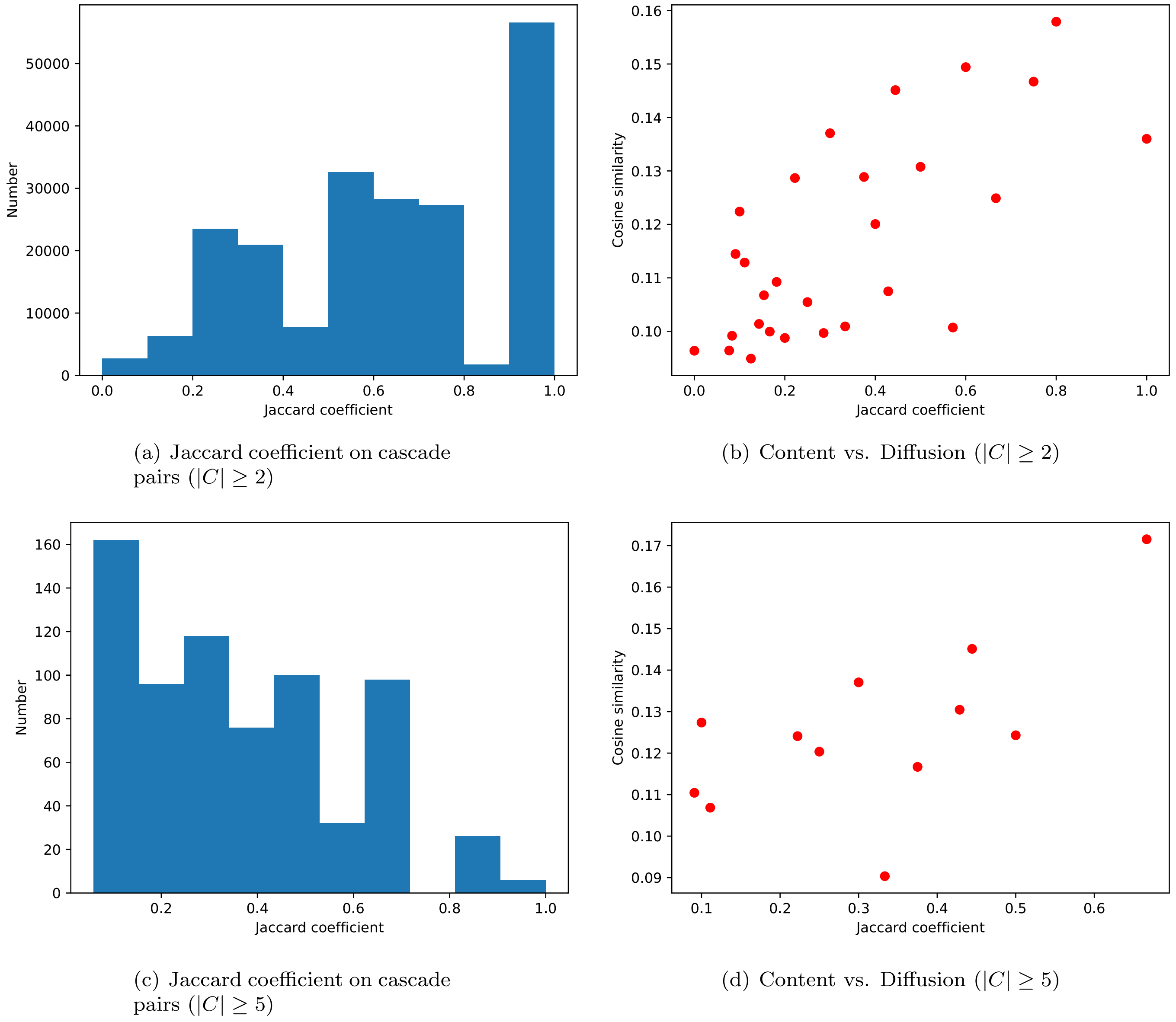
Ignoring the content being diffused entails that, in these models, different contents issued from the same user will diffuse in the same manner. In other words, in content-agnostic models, the diffusion cascades[1] originating from a given user are the same, regardless of the content being diffused.

However, this does not correspond to what is happening in real social networks. To illustrate that, we compared diffusion cascades

[*] Corresponding author at: Skopai, France.
E-mail addresses: cedric.lagnier@skopai.com (C. Lagnier), eric.gaussier@imag.fr (E. Gaussier), francois@karos.fr (F. Kawala).

[1] A cascade corresponds to the production of a content by one user of the network, as well as the ensuing re-diffusions of this content by other users. It is thus characterized by the set of users involved in the diffusion of a content. A cascade of size n involves the initial diffuser and $n - 1$ re-diffusers.

https://doi.org/10.1016/j.osnem.2018.01.001
2468-6964/© 2018 Elsevier B.V. All rights reserved.



(a) Jaccard coefficient on cascade pairs ($|C| \geq 2$)

(b) Content vs. Diffusion ($|C| \geq 2$)

(c) Jaccard coefficient on cascade pairs ($|C| \geq 5$)

(d) Content vs. Diffusion ($|C| \geq 5$)

Fig. 1. Importance of content in diffusion: histograms of the different values (discretized with bins of size 0.1) taken by the Jaccard coefficient on pairs of cascades issued from the same user and involving re-diffusion (left), and relation between content similarity (computed with the cosine) and the Jaccard coefficient on cascade pairs.

issued by the same users in the ICWSM blog dataset[2] The comparison is based on the Jaccard coefficient, that measures here the proportion of users involved in two different cascades. Let A and B be two sets, the Jaccard coefficient between those two sets is defined by:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where $|X|$ denotes the cardinal of set X. Thus, for any pair of diffusion cascades issued from the same user, one obtains a score between 0 and 1 indicating whether the users involved in the cascade are the same or not. If the score is close to 0, then the users involved in the two cascades are different; if the score is 1, then the same users are involved in the two cascades. The scores obtained can then be binned into fixed-size intervals so as to better visualize their behavior.

Fig. 1(a) displays such an histogram for the ICWSM blog dataset using only cascades of size 2 or more (*i.e.* involving the initial diffuser and at least one re-diffuser) , where the bins considered are of size 0.1, starting from 0 to 1. As one can note, this coefficient takes on very different values. If many cascade pairs get a score close to 1 (last bin on Fig. 1(a)), the majority of cascade pairs (74.5%) get a score strictly lower than 0.8. This shows that cascade pairs do not diffuse in the same way. This finding is similar to the one reported in [4] for the diffusion of hashtags related to different topics in Twitter. It has to be noted that, on average, the longer the cascades are, the smaller the Jaccard coefficient between two cascades is. This is illustrated in Fig. 1(c) that corresponds to the same experiment as before but using only cascades of size 5 or more. One can see that there are even less cascades with a high Jaccard coefficient here.

Having established that cascade pairs do not diffuse in the same way, we now consider the question whether or not the diffusion process is influenced by the content being diffused. Purely probabilistic models, as IC or LT, can generate cascade pairs that behave as in Fig. 1(a) and (c). They are however agnostic to content and would fail to accurately model cascades if they depend on the content being diffused. In order to study the relation between

[2] This dataset is fully described in Section 5.

the content diffused and the diffusion cascades, we measured the similarity between two contents being diffused by the same user and checked whether similar contents are more likely to be diffused in a similar manner than dissimilar contents. We rely here on the standard bag-of-words representation and cosine similarity to measure how similar two contents are [5]. Fig. 1(b) and (d) plot the average similarity value between two contents issued from the same user and the Jaccard coefficient computed on the two sets of users having re-diffused the two contents (to get reliable estimates, only values of the Jaccard coefficient corresponding to more than 20 cascade pairs are kept). As one can note, the points are dispersed around the first diagonal with an overall increase along this diagonal. This indicates that the more similar (resp. the more dissimilar) two contents are, the more similar (resp. the more dissimilar) their diffusion cascades are. Furthermore, the Spearman's correlation coefficient, that measures to which extent two sets of values are correlated, amounts to 0.67 for cascade pairs of size 2 or more and to 0.54 for cascade pairs of size 5 or more, indicating a positive, non-negligible correlation between content and diffusion. This shows that content plays an important role in the diffusion process: *A model of content diffusion in social media has to take into account the content being diffused*.

Another problem associated with IC or LT-based models is the fact that they rely on many parameters. Indeed, these models typically use as many parameters as there are edges in the social network. In many cases, however, the diffusion of a piece of information only involves a few users, which entails that the majority of links between users are only rarely observed. For example, most posts in blogs are only cited by authors, as illustrated in Section 5 and [6]. Relying on a lot of parameters in such cases is problematic as it is often impossible to accurately estimate these parameters.

We present here a family of models in which we integrate some of these elements. In particular, these models include:

(a) the social pressure measured using the number of neighbors who already shared the information,
(b) the interest of a user in the information diffused, measured using the similarity between the content and the profile of the user,
(c) the role played by the user, defined by her tendency to share the information she sees.

In content networks, one can observe two main types of diffusion: a user willing to share some piece of information either broadcasts it to all of her neighbors, or explicitly targets a subset of them. The first case is typical of social networks like blogs, conferences or Facebook and Twitter in their main usage, while the second one is more representative of e-mails and phone calls. In this work, we focus on the broadcast type of diffusion, but the models we are going to introduce can be easily adapted to the other type. We furthermore focus here on blogs as they provide enough context to fully capture the content being diffused (this is in contrast with social networks as Twitter in which the content being diffused is sometimes difficult to characterize).

The remainder of this paper is organized as follows: the next section describes related work on diffusion in social networks, in order to position our approach. We give some notations and formally state the problem we address in Section 3. Section 4 describes the user-centered models we introduce in this article. In Section 5, we present some experiments to validate our approach and compare it with other approaches. Section 6 deals with the influence maximization problem in the context of user-centered models. Lastly, Section 7 concludes our study, summarizing our main contributions and presenting some perspectives.

2. Related work

Information diffusion models can be classified into two main categories: contagion models and influence models. A third category, corresponding to social learning models [7], is sometimes considered; it relies on the fact that, in some cases, the adoption of a product by a user depends on the observed utility of this product on other users [8]. If some studies aim at integrating a utility parameter in contagion and influence models, they rely on assumptions (as the fact that the utility of a product for a user is known) that are not met for content diffusion in social network. A comprehensive survey on information diffusion models in social networks can be found in [9].

In contagion models, users are contaminated as soon as they are in contact with a contaminated user. Such models were originally proposed in epidemiology to understand how diseases spread, based on differential equations for the global population for going from a safe state to a contaminated one, as the *SI* (Susceptible, Infected) model, or its extensions with additional states, as the *SIR* (Susceptible, Infected, Recovered) model. Trottier et al. [10–12] provide a good description of these models; Young et al. [13] or [7] present several variants of these models in various settings. It is possible to adopt the same approach locally, by making use of infection rates that are specific to the users in contact. In this case, the SI model becomes close to the *IC* (Independent Cascade) model, as shown in [14]. In [15], SIS-like (Susceptible, Infected, Susceptible) diffusion models are used to study the coexistence of two competitive spreadings. These models are also used to find the source of a diffusion in [16].

The IC model [1] is based on the following simple principle: as soon as a user (i.e. a node in the social network) n_j is infected, she has a unique chance to infect each of her direct neighbors n_i with a probability P_{ji} that depends on both n_j and n_i. That this infection succeeds or not, n_j will make no further attempt to infect n_i. The parameters P_{ji} can be learned through maximum likelihood from observed diffusions [17]. As for the SI model, the IC model is equivalent to a bond percolation process on the graph of the social network [3,12,14].

The IC model has been extended to integrate a time variable in the diffusion model and to account for the fact that diffusion/contamination can be delayed (the contamination of neighbors does not necessarily happen as soon as a user has been contaminated). To do so, the *ASIC* (Asynchronous IC), introduced in [18], makes use of an exponential probability distribution to model the delay between the contamination of a user and its attempt to contaminate her neighbors, the contamination probability decreasing with this delay (a similar "latence" phenomenon is used in [19]). The EM algorithm is then used to estimate the parameters of this model.

In a similar way, in [20], they introduce IC-like models based where the diffusion probability between two users is defined using the proportion of actions they both performed. One particular model they propose give the credit of the contamination of a user equally distributed on all her previously contaminated neighbors. They also introduce a continuous time model in which the decay is based on an exponential distribution.

More recently, Gomez-Rodriguez et al. [21] consider different probability distributions for the delay in the contamination: exponential, power law and Rayleigh distributions. The family of models thus defined is called *NetRate*. The version based on the exponential distribution is in fact a special case of the ASIC model (obtained when setting the $k_{v,\,w}$ parameter of ASIC to a constant). An interesting point with NetRate is that it leads to a maximum likelihood optimization problem with positivity constraints in which the likelihood function is strictly concave and which has a unique solution that can be determined through standard optimization techniques. It thus does not suffer from the local maxima problem of the ASIC likelihood function. The same property can also be found in [22] for the Connie model, which aims at inferring the underlying network from diffusion cascades. The original purpose of *NetRate* and Connie is thus link prediction (as such, it can be seen as an extension of *NetInf* [23], described in [21]); however, the links predicted are based on diffusions existing in the training set, and this family of models can directly be used for information diffusion purposes, the probability of contamination between two users being equivalent to the one of predicting a link between them. Along the same direction, some models have been proposed to predict the links based on the observed cascades. In [24] or [25] for example, information diffusion models are used to infer the underlying structure of the social network. In the same family of problems, Weng et al. [26] predict link creations using diffusion cascades; Taxidou and Fischer [27] reconstruct cascades structure using information diffusion and the Twitter graph.

Influence models, also called threshold models, consider that a user is contaminated if the number or the proportion of her incoming neighbors already contaminated is above a threshold, specific to the user. It is thus the “social pressure” that determines whether a user will be contaminated or not. The first studies on such models are described in [28] and [2][3]. They have since been extended in [29–37]. The prototype of all these models is the LT model which, in its most used form, considers that a node n_i of the social network (i.e. a user) is contaminated if the sum of the weights on its incoming edges are above a threshold θ_i specific to n_i, this threshold being chosen randomly in many instances of the model [3]. As for the IC model, one can show kempe-2003) that the LT model is equivalent to bond percolation on the graph of the social network, which places this model in the same general class as the one of contagion models. A slightly different influence model, LIM (Linear Influence Model), is presented in [38], aiming at finding the global influence of infected users on the rest of the network, and at predicting the volume of the final set of contaminated users.

Lastly, generalized versions of both the IC and LT models have been proposed [39]. The generalized version of IC allows one to integrate the “social pressure”: when a node n_j is contaminated and attempts to contaminate one of its neighbors n_i, it does so with a probability $P_{ji}(S)$ that takes into account the set of all neighbors of n_i which have already tried to contaminate it and failed. $P_{ji}(S)$ is chosen so that it increases with the cardinality of S and is independent from the order in which elements of S are considered. The IC model is a specific case of this generalized cascade obtained by considering that $P_{ji}(S)$ does not depend on S. The generalization of the LT model amounts to consider a set of functions (and not only the sum) to collect the contributions of incoming neighbors already contaminated: a node n_i is contaminated if $f_i(S) \geq \theta_i$ where S is the set of incoming neighbors of n_i already contaminated and f_i is a function strictly increasing with the cardinality of S. The LT model corresponds to the case where f_i is the sum of the weights of the incoming links between n_i and its incoming neighbors already contaminated. These two generalizations lead to models that take into account different aspects (direct contagion and social pressure). They also lead to models that are equivalent: each generalized cascade model can be reformulated as a generalized threshold model, and reciprocally [39].

The re-diffusion of a piece of information, in all the above-mentioned models, solely relies on the “social pressure” (i.e. the fact that a user is more likely to re-diffuse a piece of information if it comes from many different sources) and the willingness of users to diffuse (which measures the active/passive roles users tend to have in social networks). In IC models, the social pressure is implicitly captured in the diffusion process through the fact that each active neighbor of a user tries to diffuse the information to that user: the more active neighbors a user has, the more likely she will re-diffuse the information. On the other hand, in LT models, the social pressure is explicitly modeled through the threshold θ_i associated to each user. In both models, the willingness to diffuse is captured through the weights on the outgoing edges of a particular user. The articulation between these two factors, social pressure and willingness to diffuse, furthermore varies from one model to another: in IC, social pressure arises from repeated trials based on the willingness to diffuse and, in LT, re-diffusion takes place when the combination of the two factors (captured through the sum of weights on the edges linking a particular user to her active neighbors) exceeds a certain threshold. The extension of the ASIC model proposed in [40] also takes into account users’ profile, in order to model the fact that if two users have similar profiles, then they are more likely to diffuse a piece of information to each other. This said, as the other models, it ignores the content of the information being diffused and the interest of users for this content. The models we introduce here account for this additional dimension and rely on an explicit representation of all the factors mentioned above.

Lastly, a certain number of studies have focused on the influence maximization problem given a diffusion model. This problem amounts to finding, for a given network, a given content, a given diffusion model and a given number κ, the κ initial diffusers to select so as to maximize the diffusion of the content over the network. This problem was first studied, to our knowledge, in [41] and later in [3], [14] or [42]. It is known to be NP-hard for the above-mentioned models, and to involve a diffusion function which is sub-modular, thus allowing the use of the greedy algorithm described in [43]. Similar results for the NetRate model can be found in [44]. [45] introduces an algorithm for influence maximization over continuous-time diffusion models that performs better than the standard greedy-algorithm. The models considered extend the independent cascade model. Lastly, similar problems are considered in [46] and [47]: respectively finding how much a content will diffuse using some early stage information, and assessing an influence score independently to a particular diffusion model.

As we will see, the influence maximization problem is also NP-hard for the user-centered models we are going to introduce; however, the diffusion function is no longer sub-modular and some of the properties of the greedy algorithm are lost.

A first version of the models we describe here is presented in [48]. The current contribution differs from this previous work in the following aspects. Firstly, we fully justify the need for models taking into account the content being diffused through an experimental comparison of how different contents, issued by the same user, diffuse in social networks. Secondly, we present a broader comparison to existing models using multiple measures. Thirdly, we give new illustrations aiming at providing intuitive explanations on how the models behave. Fourthly, we conduct a theoretical analysis on the relation (in this case non-equivalence) between our model and standard information diffusion models. Finally, we introduce a study of the influence maximization problem that is linked to the diffusion problem.

3. Notations and problem statement

3.1. Notations

We consider here social directed graphs $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ composed of a set of nodes/users $\mathcal{N} = \{n_1, \ldots, n_N\}$ and a set of edges/links $\mathcal{E}$. A user n_i is linked to another user n_j if $(n_i, n_j) \in \mathcal{E}$. We will furthermore use the following notations:

[3] The *LT (Linear Threshold) model is often associated with Granovetter's model.*

- $\mathcal{B}(n_i)$ will denote the set of all in-neighbors of user n_i (users who have a link to n_i):
  $\mathcal{B}(n_i) = \{n_j/(n_j, n_i) \in \mathcal{E}\}$
- Every user has a profile based on what she likes or dislikes. $\mathcal{P}$ is the set of all possible user profiles and $\forall i$, $1 \leq i \leq N$, p^i will refer to the profile of user n_i. Typically, p^i is a vector of features computed for each user.
- $\mathcal{Q} = (q^1, \ldots, q^K)$ is the set of all possible information that will propagate through the network. A piece of information (or content) is defined in the same feature space as users profiles.
- $\mathcal{M} = (M^1, \ldots, M^K)$, a set of diffusion matrices for each content q^k, of the form:

$$M^k = \begin{pmatrix} m^k_{1,0} & m^k_{1,1} & m^k_{1,2} & \cdots & m^k_{1,T^k} \\ m^k_{2,0} & m^k_{2,1} & m^k_{2,2} & \cdots & m^k_{2,T^k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m^k_{N,0} & m^k_{N,1} & m^k_{2,2} & \cdots & m^k_{N,T^k} \end{pmatrix}$$

  where $m^k_{i,t} \in \{0, 1\}$; $m^k_{i,t} = 1$ indicates that user n_i has diffused content q^k before or at time t. T^k corresponds to the duration of the diffusion of content q^k, in time steps. $M^k_{.,t}$ will denote the t^{th} column of M^k. Lastly, the set $\mathcal{M}$ is divided into two disjoint subsets: a set of training matrices, $\{(M^1, q^1), \ldots, (M^\ell, q^\ell)\}$, and a set of test matrices, $\left\{(M^{\ell+1}, q^{\ell+1}), \ldots, (M^K, q^K)\right\}$. Training matrices will be used to learn diffusion models, whereas test matrices will be used for evaluation.
- In all our models, we look at users who diffuse an information, and when it is done we consider them as active (or contaminated). A user is considered inactive if she doesn't have seen (or diffused) the content. In the processes we consider, a user who has diffused a content is necessarily active.
- $C^k(t)$ the set of all users having been contaminated before or at time t by content q^k.

For convenience in our discussion, we will also make use of the set $C^k(n_i, t)$ of incoming neighbors of n_i who have already diffused content q^k before or at time t.

3.2. Problem statement

We are interested here in the step-by-step evolution of the diffusion process, as well as in its result after a given time. We denote by $\mathcal{F}$ the function that predicts the diffusion of an information at time t given the diffusion status of the network at time $t-1$. With the elements defined above:

$$m^k_{i,t} = \mathcal{F}(n_i, \mathcal{G}, \mathcal{P}, q^k, M^k_{.,t-1}) \tag{1}$$

The function $\mathcal{F}^{(t)}$ predicting the result of the diffusion process after a given time can be constructed from $\mathcal{F}$ by "unfolding" it over time, where $^{(t)}$ denotes the composition of $\mathcal{F}$ t times. In previous studies, $\mathcal{F}$ depends neither on $\mathcal{P}$ nor on q^k, and we make here the assumption that exploiting information from $\mathcal{P}$ and q^k will result on a better prediction of how information diffuses. Furthermore, we also want to rely on few features (and thus parameters) so as to be able to rely on models robust to several diffusion scenarios, involving either few or many diffusions. The goal of the present study is thus twofold:

1. Learn "simple" mappings $\mathcal{F}$, from $\mathcal{G}$, $\mathcal{P}$, q^k and the training set $((M^1, q^1), \ldots, (M^\ell, q^\ell))$, where "simple" means that such mappings should rely on few parameters;
2. Assess whether relying on few parameters and exploiting $\mathcal{P}$ and q^k lead to better diffusion models.

4. Content diffusion models

As mentioned before, previously proposed information diffusion models merely rely on the "social pressure" (i.e. the number or proportion of incoming neighbors who have already diffused the information) in order to determine whether a given user is likely to diffuse the information or not. They thus ignore two aspects that may be crucial for information diffusion in content networks, namely the interest of users in the content being diffused, and the active/passive role each user has in the social network. We show in this section how these different aspects can be captured through simple feature functions, prior to present the basis functions defined on top of these features and used to estimate $\mathcal{F}$.

4.1. User-based features

The *thematic interest* of each user in the content diffused can be modeled as a proximity between user profiles (describing their interests) and the content diffused. A general form for this proximity is:

$$S(n_i, \mathcal{P}, q^k, \theta_s) = sim(p^i, q^k) - \theta_s$$

where θ_s is a threshold and $sim(p^i, q^k)$ represents a similarity between the content diffused and the user profile. Setting θ_s to 0 amounts to relying solely on the similarity between the user profile and the content diffused; higher values of θ_s allow one to "discourage" diffusion when the user interest in the content is not sufficient. We use in this study the *cosine* similarity for *sim*, but other choices are possible.

The *activity*, or active/passive role, can directly be measured, on the training set, through the ratio between the number of contents received and diffused by a user and the number of contents received by that user:

$$Act(n_i, \mathcal{G}, \mathcal{D}) = \frac{\sum_{k=1}^{l} I(|C^k(n_i, T^k - 1)| > 0) m^k_{i,T^k}}{\sum_{k=1}^{l} I(|C^k(n_i, T^k - 1)| > 0)}$$

where $I()$ denotes the indicator function. This measure can be generalized by introducing a threshold, through:

$$W(n_i, \mathcal{G}, \mathcal{D}, \theta_w) = Act(n_i, \mathcal{G}, \mathcal{D}) - \theta_w$$

$W(n_i, \mathcal{G}, \mathcal{D}, \theta_w)$ represents the willingness of user n_i to diffuse information, and θ_w plays a role similar to the one of θ_s above.

Lastly, the *social pressure* on each user, i.e. the fact that many different neighbors have diffused a given content, is traditionally measured, either implicitly or explicitly, through the number of incoming neighbors having already diffused the information. We denote the associated measure:

$$SP(n_i, \mathcal{G}, M^k, t)$$

The way this feature is computed is model dependent and we detail it below, but intuitively it corresponds to the number of incoming neighbors of user n_i who have diffused the content q^k before or at time t.

Each user can thus be represented by a vector of three features evolving over time for each content q^k, omitting, for readability reasons, the other arguments $(\mathcal{P}, \mathcal{G}, M^k_{.,T^k-1}, \theta_s, \theta_w)$:

$$\begin{pmatrix} S(n_i, q^k) \\ W(n_i) \\ SP(n_i, t) \end{pmatrix}$$

Given the three proposed features, one can define basis functions for each user, content and timestamp, that will be used to build the prediction functions. We focus in this study on basis functions that are simple linear combinations of the above features:

$$f_\lambda(n_i, q^k, t) = \lambda_0 + \lambda_1.S(n_i, q^k) + \lambda_2.W(n_i) + \lambda_3.SP(n_i, t) \tag{2}$$

Parameters $\lambda_0 \lambda_1, \lambda_2, \lambda_3$ are controlling the influence of each dimension of the contamination. The estimation of all parameters, including θ_s and θ_w, is described in Section 4.3.

The linear model above has the advantage of being easy to estimate and scalable. If it does not account for potential dependencies between the three dimensions considered (thematic interest, activity and social pressure), it still represents a good basis for integrating them and capturing the effect that the more important each dimension is, the more likely the user will re-diffuse.

4.2. Probabilistic modeling

Probabilistic models for information diffusion allows one to model the uncertainty inherent to the diffusion process. In this case, one does not consider that each user has either diffused a given content or not, but rather that each user has a certain probability of having diffused the given content. Two quantities are useful here:

- $P(n_i, q^k, t)$, the probability that user n_i diffuses content q^k at time t;
- $P(n_i, q^k, \leq t)$, the probability that user n_i has diffused content q^k before or at time t.

These two quantities are related through:

$$P(n_i, q^k, \leq t+1) = P(n_i, q^k, \leq t) + (1 - P(n_i, q^k, \leq t))P(n_i, q^k, t+1) \quad (3)$$

A user having diffused before or at time $t+1$ has either diffused before time t, or has not and has diffused at time $t+1$. Furthermore, because of the definition of $P(n_i, q^k, \leq t)$:

$$\mathcal{F}^{(t)}(n_i, \mathcal{G}, \mathcal{P}, q^k, M^k_{.,0}) = P(n_i, q^k, \leq t)$$

When the thematic interest of the user is high, or when her willingness to diffuse or her social pressure is high, $P(n_i, q^k, t)$ should be high; conversely, when thematic interest, willingness to diffuse and social pressure are low, $P(n_i, q^k, t)$ should be low. Such a behavior is naturally captured in the logistic function, which acts as a soft thresholding process and yields valid probability functions. Furthermore, a user cannot diffuse a content if no incoming neighbor has already diffused it. Because of the probabilistic setting retained here, one does not have a direct access to $|C^k(n_i, t)|$, the number of incoming neighbors having already diffused, but rather to an expectation of it ($E[|C^k(n_i, t)|]$). Hence:

$$SP(n_i, \mathcal{G}, M^k, t) = E[|C^k(n_i, t)|]$$

and:

$$P(n_i, q^k, t+1) = \begin{cases} (1 + e^{-f_\lambda(n_i, q^k, t)})^{-1} \text{ if } E[|C^k(n_i, t)|] > 0 \\ 0 \text{ otherwise} \end{cases} \quad (4)$$

with $(\lambda_1, \lambda_2, \lambda_3)$ positive or null (when a feature has no impact on the diffusion).

The expectation $E[|C^k(n_i, t)|]$ is defined as: $E[|C^k(n_i, t)|] = \sum_{m=0}^{|\mathcal{B}(n_i)|} m\, P(|C^k(n_i, t)| = m)$, where $P(|C^k(n_i, t)| = m)$ is the probability that the number of incoming neighbors who have diffused the content is m. It is easy to show that (we skip here the derivation which is purely technical):

$$E[|C^k(n_i, t)|] = \sum_{n_j \in \mathcal{B}(n_i)} P(n_j, q^k, \leq t) \quad (5)$$

The dynamics of the diffusion thus evolves, from one time step to another, through:

1. Initialization: $P(n_i, q^k, \leq 0) = 1$ for initial diffusers, 0 otherwise;
2. Iteratively compute (from $t = 0$):
   - $E[|C^k(n_i, t)|]$ using Eq. (5)
   - $P(n_i, q^k, t+1)$ using Eq. (4)
   - $P(n_i, q^k, \leq t+1)$ using Eq. (3)

The main problem with the above model, however, is that, at any timestamp, $P(n_i, q^k, t+1) \geq P(n_i, q^k, t)$. Then if $P(n_i, q^k, t)$ is strictly positive at some point in time, $\lim_{t \to +\infty} P(n_i, q^k, \leq t) = 1$. This is due to the fact that users are "aware" of the content they have already diffused at all time steps, and that their probability of diffusing will be reinforced by subsequent receptions of a given content (for this reason, we refer to this model as *RUC*, for *Reinforced User-Centered*). The following model corrects this drawback.

4.2.1. A time-decaying extension

The quantity $P(n_i, q^k, t)$ becomes strictly positive as soon as $E[|C^k(n_i, t)|]$ is strictly positive, and one would like, in this latter measure, that the influence of users having diffused an information a long time ago be less important than the one of users having diffused the information recently. One can thus replace Eq. (5) by the following equation:

$$E[|C^k(n_i, t)|] = \sum_{n_j \in \mathcal{B}(n_i)} \rho(n_j, q^k, t) \quad (6)$$

where $\rho(n_j, q^k, t)$ is a function of the influence n_j has on her outgoing neighbors at time t wrt content q^k, penalizing "old" diffusions:

$$\rho(n_j, q^k, t+1) = \delta \times \rho(n_j, q^k, t) + (1 - P(n_j, q^k, \leq t))P(n_j, q^k, t+1) \quad (7)$$

By definition, $\rho(n_j, q^k, t = 0) = 1$ for initial diffusers and 0 otherwise. $\delta, 0 \leq \delta \leq 1$ is a decaying parameter controlling the penalization on old diffusions. When $\delta = 1$, $\rho(n_j, q^k, t) = P(n_j, q^k, \leq t)$ and one recovers the RUC model. The other quantities of the RUC model remain unchanged. We will refer to the model with a decaying parameter as *DRUC*, for *Decaying Reinforced User-Centered*.

4.3. Parameter estimation

Setting θ_s and θ_w: We now turn to the problem of setting the thresholds θ_s and θ_w. A user having a similarity with the query above θ_s is more likely to diffuse an information; conversely, a user with a similarity below θ_s is more likely to not diffuse the information. The global similarity function S defined above is positive in the first case and negative in the second one. θ_s thus corresponds to a threshold on the similarity function above which a user is more likely to diffuse an information, and can be obtained, from the training set, through a line search on the cosine similarity values between content diffused and user profiles. This line search process is here initialized at 0, with an increment of 0.05, and is stopped as soon as the number of users (with this specific similarity with a content) re-diffusing a content is greater than the number of users not re-diffusing it. For example, the value of θ_s is 0.35 for the MT (dense) dataset. A similar reasoning for the willingness to diffuse ($W(n_i, \mathcal{G}, \mathcal{D}, \theta_w)$) directly leads to $\theta_w = 0.5$. Indeed, if a user shares more than 50% of the content she sees, the willingness should have a positive effect on the diffusion probability (and a negative effect if she shares less than 50%).

The parameters $\lambda_0, \lambda_1, \lambda_2$ and λ_3 can be learned through maximum likelihood, with positivity constraints using the gradient ascent method.

Let $\mathcal{L}(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ denote the likelihood of the training set. The learning problem we face can be formulated as:

$$\begin{cases} \text{argmax}_{\lambda_0, \lambda_1, \lambda_2, \lambda_3} \mathcal{L}(\lambda_0, \lambda_1, \lambda_2, \lambda_3) \\ \text{subject to: } \lambda_1 \geq 0,\ \lambda_2 \geq 0,\ \lambda_3 \geq 0 \end{cases}$$

where the positivity constraints are dictated by the choice of the diffusion function at the basis of our models. As these contraints are "interval" constraints, one can resort to projected gradient approaches to estimate the parameters, in which each gradient descent/ascent step is followed by a projection of the parameters on the admissible intervals. Its application here leads to the following update formulas (in between iterations $(p+1)$ and (p)), in which $\mathcal{L}$ stands for the log-likelihood function:

$$\forall i \in \{0,1,2,3\} : \begin{cases} \lambda_i^{(p+1)} = \lambda_i^{(p)} + \alpha \frac{\partial \mathcal{L}(\lambda_0^{(p)},\lambda_1^{(p)},\lambda_2^{(p)},\lambda_3^{(p)})}{\partial \lambda_i} \\ \text{If } \lambda_{i(i\neq 0)}^{(p+1)} < 0, \text{ then } \lambda_{i(i\neq 0)}^{(p+1)} = 0 \end{cases}$$

where α controls the step of the descent along the gradient of $\mathcal{L}$.

The likelihood, on the training set, for the RUC and DRUC models, takes a simple form as it is directly based on the probability of each user to be active at each timestamp:

$$\mathcal{L}(\lambda_0,\lambda_1,\lambda_2,\lambda_3) = \prod_{k=1}^{l}\prod_{t=1}^{T^k}[\prod_{n_i \in C^k(t)} P(n_i,q^k,\leq t) \prod_{n_i \notin C^k(t)} (1 - P(n_i,q^k,\leq t))]$$

For efficiency reasons, we make use of the recurrence equation (Eq. (3)) to compute the partial derivatives, and store, for each user, the current values of $P(n_i,\, q^k,\, \leq t)$ and its derivative at each timestamp. The exact form of these derivatives is given in appendix A.

4.4. Non equivalence with previous models

Standard models (IC and LT) have been proved equivalent to a bond percolation process by Kempe et al. [3]; we show here that such an equivalence does not hold for the models defined previously.

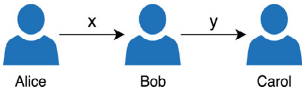
Bond percolation was originally introduced to model water flowing in networks through the following, simple stochastic process: water flows through an edge e with probability p_e and does not with probability $(1-p_e)$. By letting water flows according to this process, one can determine at the end which edges were used. Such edges are called *open* (or *occupied*), whereas the other edges are called *closed* (or *unoccupied*). In the context of information diffusion, an open link corresponds to a diffusion of the information between the involved users, whereas a closed link corresponds to no diffusion between them.

More formally, given a graph $\mathcal{G} = (\mathcal{N},\mathcal{E})$, let $|\mathcal{E}|$ denote the number of edges in $\mathcal{G}$, let v be any $|\mathcal{E}|$-dimensional vector taking values in $\{0,1\}^{|\mathcal{E}|}$, and let $\mathcal{V}$ be the set of all possible vectors v. Each vector v describes a possible open/closed allocation to all the links in the graph. A bond percolation process on $\mathcal{G}$ is fully determined by a probability distribution $q(v)$ on $\mathcal{V}$. We will now prove the following theorem.

Theorem 1. *The model RUC (and by extension DRUC) defined on a graph $\mathcal{G} = (\mathcal{N},\mathcal{E})$ is not equivalent to a bond percolation process on $\mathcal{G}$.*

Proof. To prove the above statement, it is sufficient to exhibit a graph on which an instance of RUC is not equivalent to a bond percolation process. Such an example is described on the simple graph given in Fig. 2(a). It contains 3 users, connected through 2 edges, and the initial diffuser is A, so that, after 2 time steps, all the users have a non-null probability to have received the information diffused by A. The set $\mathcal{V}$ contains 4 vectors, and a bond percolation process on this graph is characterized by 3 probability values (p_1,p_2,p_3), the fourth probability p_4 being defined by: $p_1+p_2+p_3+p_4=1$. Fig. 2(b) gives the associated probability table.

Let us assume that the RUC model is equivalent to a bond percolation on the graph displayed in Fig. 2(a). The probabilities that users B and C are active after 2 time steps with the RUC model are then given by:



(a) Social graph

probability	x	y
p_1	closed	closed
p_2	closed	open
p_3	open	closed
p_4	open	open

(b) Probabilities table

Fig. 2. Link percolation example.

- $P(Bob, q^k, \leq 2) = p_3 + p_4$
- $P(Carol, q^k, \leq 2) = p_4$

which leads to:

$$\begin{aligned} p_3 &= P(Bob,q^k,\leq 2) - P(Carol,q^k,\leq 2) \\ &= P(Bob,q^k,1) + (1 - P(Bob,q^k,1)) \times P(Bob,q^k,2) \\ &\quad - P(Carol,q^k,2) \end{aligned}$$

where the last equality is based on the definitions of $P(Bob,\, q^k,\, \leq 2)$ and $P(Carol,\, q^k,\, \leq 2)$. This value can be negative if, for example, user *Carol* has a profile closer to the content diffused than user *Bob*. Setting $\lambda_0 = \lambda_2 = \lambda_3 = 0,\ \lambda_1 = 1,\ Act(Bob,\mathcal{G},\mathcal{D}) = Act(Carol,\mathcal{G},\mathcal{D}) = \theta_w = 0,\ sim(p^{Bob},q^k) = 0,\ sim(p^{Carol},q^k) = 1$ and $\theta_s = 0.5$, indeed leads to: $P(Bob,\, q^k,\, \leq 2) \approx 0.61$, $P(Carol,\, q^k,\, \leq 2) \approx 0.62$ and then $p_3 < 0$, which comes as a contradiction since p_3 should be non-negative. □

Theorem 1 shows that the models introduced here radically differ from the models proposed previously.

5. Experimental validation

In order to evaluate the user-centered models, we make use of two datasets:

- The *ICWSM* [49] dataset is composed of blog posts and links between them. Each user corresponds to a blog and diffusion of information is observed through links between blogs: if post p_2 of blog b_2 contains an hyperlink to post p_1 of blog b_1, then we consider that b_2 has diffused the content coming from b_1;
- The *MemeTracker* (MT) [50] dataset is composed of blog posts and links between them. Contrary to the ICWSM dataset, no blog url is attached to a post. We thus inferred blogs using post urls (a post url contains the url of the blog it belongs to). To do so, we cut post urls at the first "/" character after "http://" and assume that the string obtained corresponds to the url of the blog. As for the ICWSM dataset, we consider that information diffuses from one user (blog) to another if there is a link from a post of the former to a post of the latter.

For each dataset, we have extracted two different corpora:

- *Sparse* corpora have been built by selecting randomly 100,000 cascades of blog posts. In this case, many of the selected cascades do not diffuse over the network resulting in a case where the models can only be trained on a few number of diffusions. These corpora are used to evaluate the models in a context of low diffusion.
- *Dense* corpora have been built by focusing on a subset of the 5000 users that are the most active. We have only kept the cascades over these active users which have been linked at least one time. These two corpora are used to evaluate the models in the context of a dense diffusion.

The number of users, cascades and the mean size of the cascades are given in Table 1. The length of a cascade is 1 if the information diffuses once from an initial user to another one. As one

Table 1
Main statistics of datasets for the *Sparse* and *Dense* versions. MT denotes MemeTracker.

Dataset	#nodes	#links	#terms	#casc.	avge_size	max_size
MT (dense)	5000	4373	24, 482	2977	1.21	4
ICWSM (dense)	5000	17, 746	173, 014	23, 738	1.075	11
MT (sparse)	39, 427	10, 816	70, 602	104, 973	0.006	10
ICWSM (sparse)	40, 268	62, 657	262, 290	104, 980	0.018	33

can see, *Sparse datasets* are composed of low length cascades – i.e. many cascades do not diffuse – while *Dense datasets* are composed of larger cascades.

For each corpus we performed the following normalization operations:

- Taking posts during only one month;
- Filtering out of non-English posts;
- Removal of empty words using an empty words list;
- Stemming using Porter stemming;
- Filtering out of words appearing less than five times.

The above preprocessing then yields a standard word vector for each post. The vector for a cascade is then computed by averaging the vectors of all the posts that compose a cascade. The profile of each user is computed by averaging the vectors of the cascades diffused by the user on the training set. In order to evaluate the different models, we use a 5-fold cross validation scheme (4 blocks for training, one for testing). The 4 first blocks are used to (a) build the graph between blogs, by considering that two blogs n_i and n_j are connected if a piece of information has diffused between n_i and n_j, and (b) to estimate model parameters. The last block is used for evaluation purposes. All the results presented below are averaged over the 5 different splits.

5.1. Evaluation measures

We use the following standard measures to evaluate different diffusion results:

- *Precision/recall curves (PRC)*: for each cascade, users are first ranked by their probability to be active (as predicted by the model under evaluation). The precision at each recall point (corresponding to a user who is really active in the dataset) is then computed, and averaged over all cascades [51]. As all cascades do not have the same number of recall points, the variance, computed over all cascades, is higher for the last recall points;
- The *Mean Average Precision (MAP)* is a global measure of quality over recall points. It is computed as follows:
$$MAP = \frac{1}{K-l}\sum_{k=l+1}^{K}\frac{1}{|C^k(T^k)|}\sum_{n_i\in C^k(T^k)} Precision(R_{k,i})$$
where R_{ki} is the set of users who have a probability of being active greater than or equal to the one of user n_i (note that $n_i \in R_{ki}$);
- *Relative Volume Error:* The relative volume error, introduced in [38], aims at measuring to which extent a model is able to predict the number of users that will be active at the end of the diffusion process (or after a given time). It thus represents a global measure that captures the tendency to diffuse of a model (regardless of whether the diffusion is correct or not). It is measured by the difference between the expected number $\hat{V}_k$ (given by a model) and the real number of active users V_k, averaged over all cascades:
$$RVE = \frac{\sqrt{\sum_{k=l+1}^{K}(V_k(T^k)-\hat{V}_k(T^k))^2}}{\sqrt{\sum_{k=l+1}^{K}V_k(T^k)^2}}$$

5.2. Models tested

We compare the models presented above with several baseline diffusion models used in previous studies. We have retained here the most widely used models. Some of them have been extended in different directions, without changing their main characteristics. Relying on the standard versions of these models allows us to simply assess how much the new dimensions considered in the user-centered family of models are useful for content diffusion. The models we have retained are the following:

1. The *Independent Cascade Model* (IC). Its parameters are learned through the EM algorithm proposed in [17];
2. The *Asynchronous Independent Cascade Model* (ASIC) which is described in [18] and represents an asynchronous version of the IC model. Its parameters are also learned through an EM algorithm;
3. The *NetRate* (NR) model [21], with the exponential distribution;
4. The *RUC* and *DRUC* models presented in Section 4.2; In this study, we have arbitrarily set the parameter δ to 0.9, which amounts to consider a small decay over time.

In addition, we will make use of a simplified version of the user-centered models, called **UC** (for *User-Centered model*), that will allow a more direct comparison with the independent cascade models. In UC, each user n_j active at time t has a unique chance to contaminate each of its outgoing neighbors n_i at time $t+1$ based on $P(n_i, q^k, t+1)$ (see Section 4.2). If it succeeds, the contamination value associated to n_i at time $t+1$ is 1, and 0 otherwise. In this setting, $SP(n_i, \mathcal{G}, M^k, t)$ is equal to the number of active incoming neighbors ($|C^k(n_i, t)|$). This behavior is similar to the one of the *Independent Cascade* model, except that n_j, in case of failure, will still contribute to the contamination of n_i through $SP(n_i, \mathcal{G}, M^k, t)$ if another incoming neighbor of n_i is contaminated at time τ, $\tau \geq t$. The contamination stops once the contamination values of each user do not change.

5.3. Results

The Relative Volume Error for each model is available in Table 2. We compare baseline models – IC, ASIC and NetRate – with proposed models. Except on Dense ICWSM, User-Centered models obtain similar results as baseline models. They are all near 1, but baseline still tends to be better. On Dense ICWSM, UC obtain really poor results. The particularity of this dataset compared to the others is that models are trained on more diffusing cascades. Table 3 shows the average diffusion volume predicted by each model. For this particular dataset and model, we can see that UC predict a lot more diffusion than the other models and obviously a lot more than necessary occurring in some error. Looking at this table, we see that, except on Sparse MemeTracker which is the dataset with the less diffusing cascades, baseline models diffuse really few compared to RUC and DRUC. Their relative volume error close to 1 is due to a lack of diffusion while RUC and DRUC tend to diffuse too much to wrong users.

Table 4 shows the Mean Average Precision for all models. Except IC model on Dense ICWSM, on all datasets User-Centered models out-perform baseline models. To link up with the error,

Table 2
Relative volume error for the different methods. MT denotes MemeTracker and NR NetRate.

Dataset	0-diff	IC	ASIC	NR	UC	RUC	DRUC
MT (dense)	1	0.916	0.954	**0.884**	1.391	1.412	1.203
ICWSM (dense)	1	0.756	0.893	0.957	6.259	0.901	**0.719**
MT (sparse)	1	1.009	**1.005**	1.041	1.009	1.043	1.069
ICWSM (sparse)	1	**0.996**	1.000	1.012	1.185	1.071	1.206

Table 3
Average diffusion volume for the different methods. MT denotes MemeTracker and NR NetRate.

Dataset	0-diff	IC	ASIC	NR	UC	RUC	DRUC
MT (dense)	0	0.098	0.050	0.309	0.856	0.575	0.719
ICWSM (dense)	0	0.435	0.211	0.124	1.810	0.592	0.811
MT (sparse)	0	0.002	0.001	0.004	0.003	0.003	0.004
ICWSM (sparse)	0	0.002	0.001	0.002	0.011	0.013	0.020

Table 4
Mean Average Precision for the different methods. MT denotes MemeTracker and NR NetRate.

Dataset	IC	ASIC	NR	UC	RUC	DRUC
MT (dense)	0.283	0.134	0.147	0.608	**0.627**	0.622
ICWSM (dense)	0.712	0.313	0.100	0.787	0.817	**0.836**
MT (sparse)	0.015	0.020	0.012	0.121	0.574	**0.620**
ICWSM (sparse)	0. 088	0.043	0.019	0.191	0.691	**0.709**

Table 5
Values of the parameters after learning on *Dense ICWSM*. These values illustrate the importance of the different features for predicting diffusion.

Model	UC	RUC	DRUC
λ_0 (bias)	−3.13	−6.77	−2.61
λ_1 (thematic interest)	7.06	6.77	10.75
λ_2 (activity)	3.33	5.52	4.27
λ_3 (social pressure)	1.94	3.23	1.55

Table 6
Values of the parameters after learning on *Dense MemeTracker*. These values illustrate the importance of the different features for predicting diffusion.

Model	UC	RUC	DRUC
λ_0 (bias)	−2.22	−5.47	−3.33
λ_1 (thematic interest)	6.21	7.01	9.49
λ_2 (activity)	2.22	5.92	3.99
λ_3 (social pressure)	1.39	2.78	0.95

User-Centered models tends to diffuse more than baseline models but “choose” better the users who diffuse. In other words, User-Centered models give a higher probability to correct users while they still give a non null probability to some wrong users. A discussion on the utility of the similarity parameter for the RUC model is proposed in appendix B.

This difference is explained by two points of content-aware models:

(a) They rely on fewer parameters and are thus more robust to sparsity in the training set;
(b) They take into account the content of the information diffused and thus can explain why different contents issued from the same users diffuse in different ways.

Concerning the first point, Figs. 3 and 4 displays the results for the PRC on the sparse datasets. As one can note, the difference between content-aware diffusion models (UC, RUC and DRUC) and the other models (IC, ASIC, NetRate) is really important on all recall points, while the increase in MAP reaches 60% (in absolute value) between the best content-aware model and the best standard model on both datasets. If taking into account the content diffused may be beneficial in the context of sparse datasets, it is clear that all standard models rely on a large number of parameters (they have at least as many parameters as there are edges in the network) whereas the training sets contain few observations. Indeed, as displayed in Table 1, the number of parameters/edges is greater than the number of observations, that is the number of observed diffusions, which amounts to the number of cascades multiplied by the the average cascade length. This means that few edges, compared to the number of actual edges, are observed in the training dataset. Most parameters are thus very poorly estimated in these models, resulting on poor diffusion prediction. In contrast, the content-aware models introduced here rely on only 4 parameters that can still be learned from sparse training data.

Concerning the second point, Figs. 5 and 6 displays the results for the PRC on the dense datasets. The difference between standard models (IC, ASIC, NetRate), on the one hand, and the content-aware ones (UC, RUC and DRUC), on the other hand, is less important in this case. The increase in MAP between the best standard model and the best content-aware model reaches 10% on ICWSM and 35% on MemeTracker. On ICWSM, the number of observed edges (number of cascades multiplied by average cascade length) is larger than the number of edges in the network. Standard models can, in this case, be correctly learned but still fall behind content-aware models, on all recall points. On the dense dataset extracted from MemeTracker, in addition to the content problem, one also encounters a sparsity problem as the number of parameters to be learned is still important with respect to the number of observations.

In order to further analyze the importance of content in modeling information diffusion, we provide in Tables 5 and 6 the values of the parameters learned by the three content-aware models models (UC RUC and DRUC) for the two dense datasets. In these tables, λ_0 is the bias of the model, λ_1 is the weight associated with the *thematic interest,* λ_2 is the weight associated with the *activity* and λ_3 is the weight associated with the *social pressure*. As one can note, the value for λ_1 is always higher than that of the other parameters, which confirms the importance of the thematic interest for information diffusion.

Within the standard models, IC is the one that performs best on all datasets. We believe that this is due to the fact that ASIC and Netrate introduce a strong decay in the diffusion through an exponential model. As the number of diffusions in each dataset

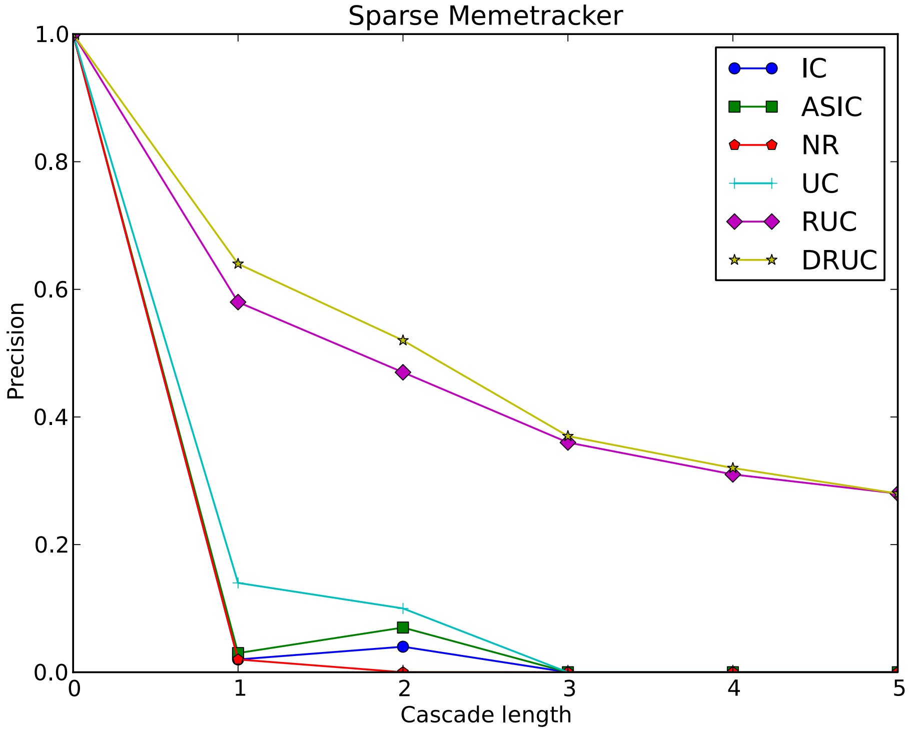


Fig. 3. Precision curves on the *Sparse MemeTracker* dataset.

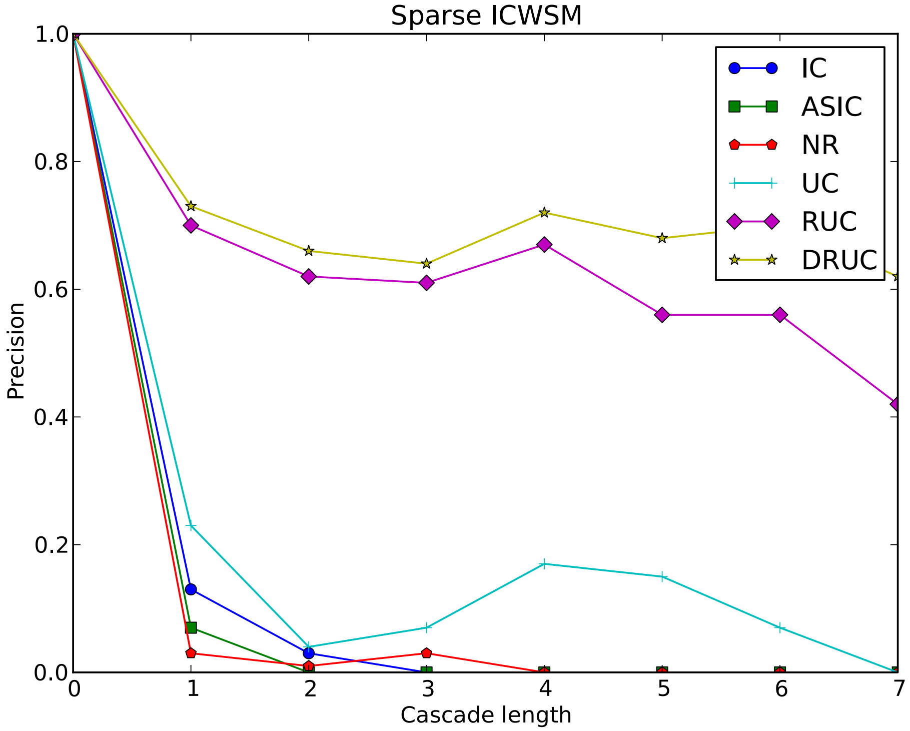


Fig. 4. Precision curves on the *Sparse ICWSM* dataset.

is still low, the probability predicted by these models is also low and dominated by the exponential term (of the form $e^{-P_{ij}(t-t_0)}$). The difference between these values is thus small and the models fail to differentiate between diffusions and non-diffusions. Indeed, when the number of training data is higher (as for the dense dataset extracted from MemeTracker), the difference between IC and ASIC or NetRate becomes less important, the parameters of these models being better estimated in this case.

Within the content-aware models, DRUC outperforms RUC on three over four datasets. This is particularly true over the large cascades as DRUC is better for modeling long diffusions - see Section 4.2 for explanation. Furthermore RUC obtain better results compared to UC. These facts show an improvement when using the reinforcement which allows users to diffuse later and an other improvement using the delay parameter, which makes a decay of the diffusion probability over time.

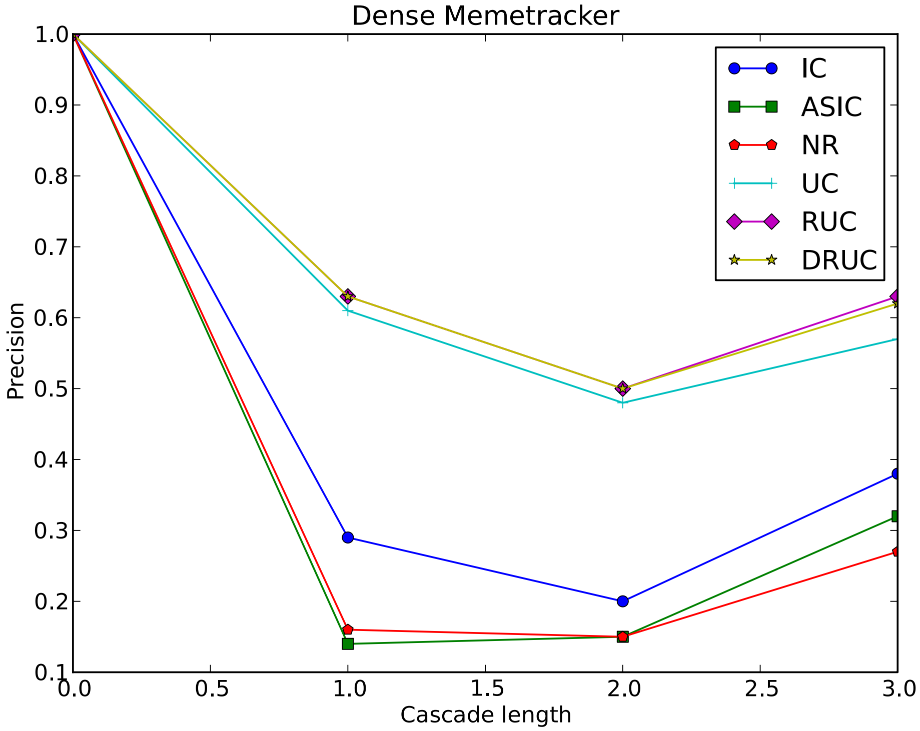


Fig. 5. Precision curves on the *Dense MemeTracker* dataset.

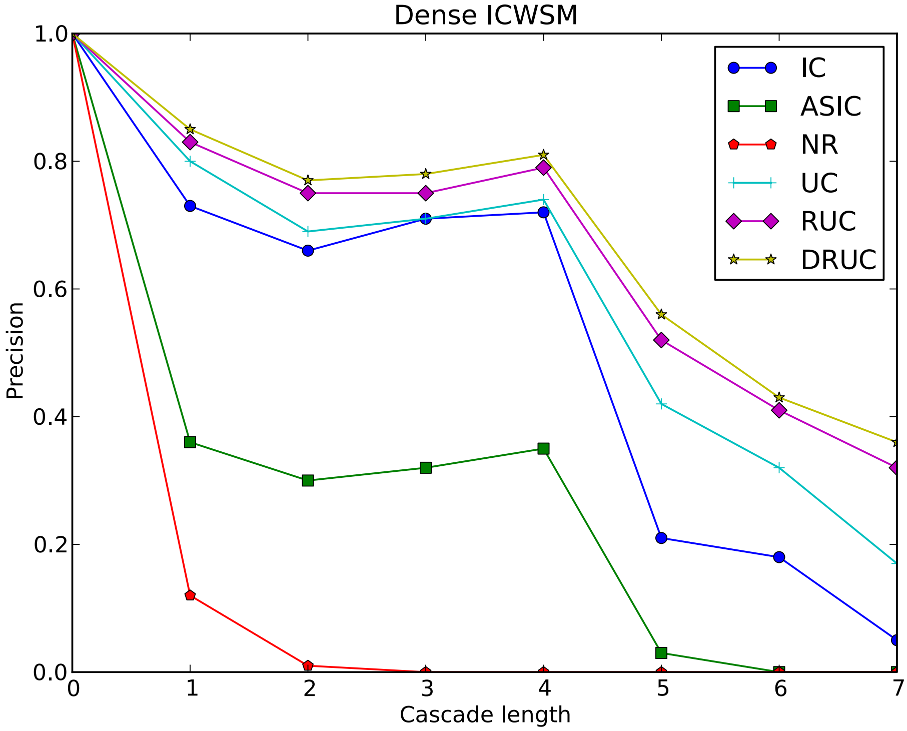


Fig. 6. Precision curves on the *Dense ICWSM* dataset.

6. Influence maximization (IM)

The influence maximization problem is an optimization problem defined on a “social-graph” $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and a diffusion model, the goal of which being to determine the κ most influential users. We will refer to this problem for the model RUC as *IM*(κ, *RUC*). The influence maximization problem is known to NP-hard for the IC and LT models [3], and we show here that this is also the case for RUC (and by extension DRUC).

Theorem 2. *IM*(κ, *RUC*) *is NP-hard.*

Proof. To prove Theorem 2, we show that the set cover problem, one of Karp’s 21 NP-complete problems [52], can be answered through the IM decision function.

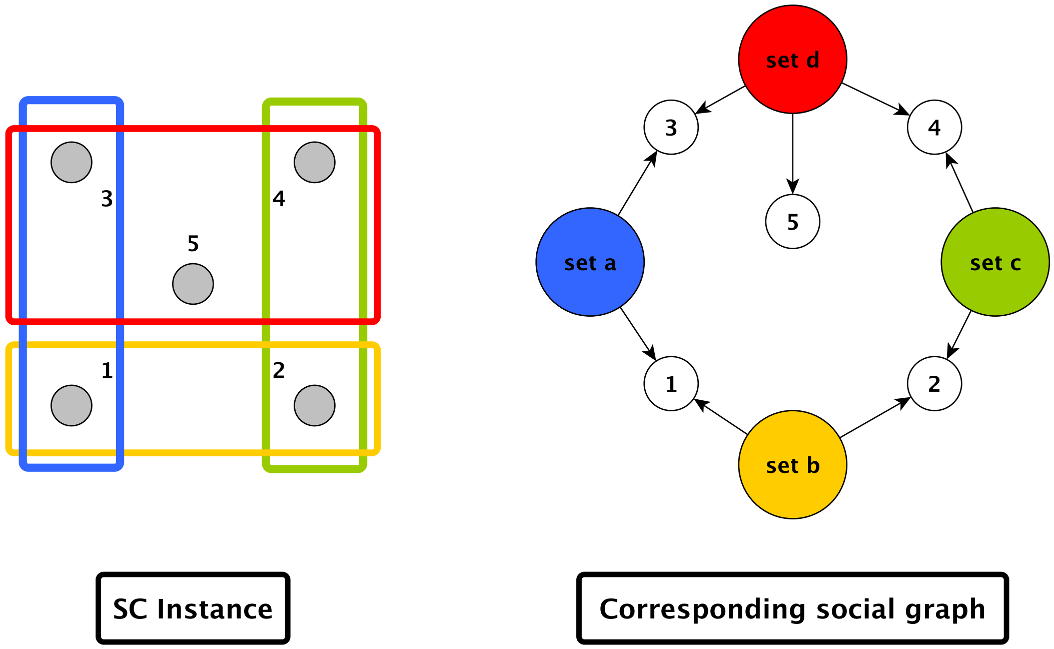


Fig. 7. Γ application used to reduce SC to IM. Nodes {1,2,3,4,5} are element nodes. Nodes $set_{\{a,\ b,\ c,\ d\}}$ are set nodes.

The set cover (SC) decision problem takes as input a collection of subsets $\mathcal{C}$ of a finite set $\mathcal{U}$ called the universe, and an integer κ such that $\kappa \leq |\mathcal{C}|$. The SC problem answers true if and only if there exists a family $\mathcal{F}$ of subsets in $\mathcal{C}$ such that $|\mathcal{F}| \leq k$ and $\bigcup_{f\in\mathcal{F}}(f) = \mathcal{U}$ (*i.e.* the family $\mathcal{F}$ covers the universe). Let us now consider the mapping Γ that (a) associates a node to each element of $\mathcal{U}$ (we will refer to these nodes as *element nodes*) and each subset of $\mathcal{C}$ (we will refer to these nodes as *set nodes*) and (b) links, through a directed edge, a set node to all the element nodes it contains. Fig. 7shows an example of the mapping Γ and the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ obtained from the original SC problem.

Now, let us consider a particular RUC model in which all parameters are set to 0. That is, for all non initial diffusers:

$$P(n_i, q^k, t+1) = \begin{cases} (1+exp(0))^{-1} = \frac{1}{2} & \text{if } R_t(n_i) \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

where $R_t(n_i)$ denotes n_i's reachability ($R_t(n_i)$ is true if and only if at least one of n_i's incoming neighbors has a non null probability to be active before or at time t). For initial diffusers, $P(n_i, q^k, \leq t) = 1, \forall t$, and, by recurrence over t, one obtains, for non initial diffusers:

$$P(n_i, q^k, \leq t+1) = \begin{cases} 1 - (\frac{1}{2})^{(t+1)} & \text{if } R_t(n_i) \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

If one selects, for example, the nodes set_a and set_d in Fig. 7 as initial diffusers, then the diffusion process will result in defining nodes {1, 3, 4, 5} as active and node 2 as inactive.

By choosing initial diffusers among the set nodes and letting the diffusion process unfolds over a sufficiently long period of time, the nodes "covered" by the subsets selected have a probability of having received the information as close to 1 as desired. Furthermore, all the other element nodes have a probability of having received the information equal to 0. Selecting subsets in SC or their corresponding initial diffusers in the IM reduction thus yields the same coverage. This allows one to define $\mathcal{Q}$, a minimal quantity of active users ensuring that the IM decision problem answers positively if and only if every node representing one of $\mathcal{U}$'s element is active. This situation, according to our social-graph, corresponds to the existence of a complete coverage of the universe. One can thus answer the SC problem using the IM decision problem through the mapping:

$$(\mathcal{U}, \{set_0, \dots, set_n\}, \kappa) \rightarrow \left(\mathcal{G}, \kappa, \mathcal{Q} = \kappa * 1 + |\mathcal{U}| * \left(1 - (\frac{1}{2})^t\right)\right)$$

The IM decision problem thus answers the SC problem. □

6.1. Greedy hill climbing applied to user centered models

Because of its NP-hardness, the IM problem is usually addressed through the greedy algorithm defined in [43] and adapted for diffusion purposes as displayed in Algorithm 1. $\sigma(A,\ q^k,\ t)$ is the influence function and corresponds to the expected number of users who have diffused content q^k at time step t if all users of $A, A \subset \mathcal{N}$ are initial diffusers.

```
Set t to a given value
Set A = ∅
for i = 1 to $\kappa$ do
    for all $n_i \in \mathcal{N} \setminus A$ do
        compute $d = \sigma(A \cup \{n_i\}, q^k, t) - \sigma(A, q^k, t)$
        if $d$ is maximal then
            $n_{max} = n_i$
        end if
    end for
    $A \leftarrow A \cup \{n_{max}\}$
end for
```

Algorithm 1: The greedy algorithm.

It is known [43] that the greedy algorithm yields a $(1-1/e)$-approximation of the optimal value if the function σ is a non-negative, monotone submodular function, which is the case for the IC and LT models [3]. For the user-centered models, we nevertheless have the following property.

Property 1. *σ_{RUC} is not sub-modular.*

Proof. The proof simply amounts to exhibiting a graph on which the sub-modularity condition can not hold.

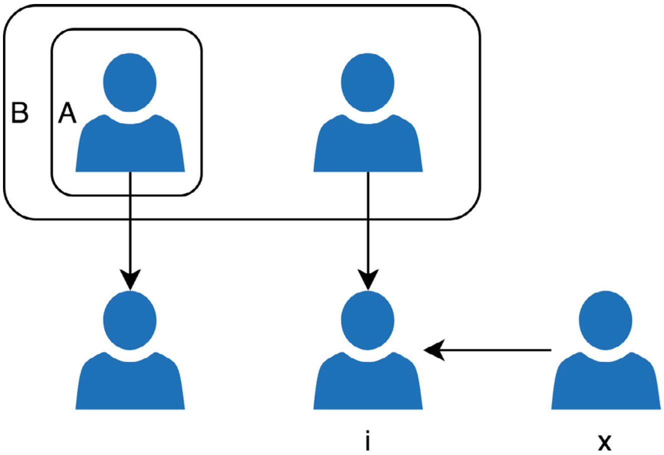


Fig. 8. Social graph used with 5 users and 2 sets of users (A and B) such that $A \subseteq B$.

Let us consider the social graph presented in Fig. 8: If σ_{RUC} is submodular, then $\forall x \notin B,\ A \subseteq B$:

$$\sigma_{\text{RUC}}(A \cup \{x\}) - \sigma_{\text{RUC}}(A) \geq \sigma_{\text{RUC}}(B \cup \{x\}) - \sigma_{\text{RUC}}(B)$$

$$\Leftrightarrow \quad \frac{1}{1+e^{-\lambda_0-\lambda_1\times s-\lambda_2\times w-\lambda_3}} \geq \frac{1}{1+e^{-\lambda_0-\lambda_1\times s-\lambda_2\times w-2\lambda_3}} - \frac{1}{1+e^{-\lambda_0-\lambda_1\times s-\lambda_2\times w-\lambda_3}}$$

$$\Leftrightarrow \quad 2 \times \frac{1}{1+e^{-\alpha-\lambda_3}} \geq \frac{1}{1+e^{-\alpha-2\lambda_3}}$$

$$\text{with } \alpha = \lambda_0 - \lambda_1 \times s - \lambda_2 \times w$$

$$\Leftrightarrow \quad \frac{1+e^{-\alpha-2\lambda_3}}{1+e^{-\alpha-\lambda_3}} \geq \frac{1}{2}$$

Setting $\lambda_3 = 2$ and $\alpha = -4$, the above inequality entails $\frac{2}{1+e^2} < \frac{1}{2}$, which is false. Hence σ_{RUC} is not submodular. □

Despite the fact that one can not theoretically conclude, using sub-modularity arguments, that the greedy algorithm yields a good approximation of the optimal solution for user-centered models, it can still be used in practice to find an approximate solution to the IM problem. We illustrate its use in the next section and compare it to naive heuristics.

6.2. Illustration

In order to further assess the use of the greedy algorithm with user-centered models, we compared its results, on the sparse MemeTracker dataset, for the RUC model, with the following standard heuristics[4]:

- Higher out-degree: the top κ nodes according to the number of outgoing edges (out-degree) are selected as the κ initial diffusers;
- Distance centrality: the distance centrality is the average distance of that node to all the other nodes in the network; if there is no path between two nodes, their distance is taken to be the number of nodes in the graph. As before, the top κ nodes according to the distance centrality are selected as the κ initial diffusers;
- Random100: a set of κ nodes in the graph is randomly chosen a hundred times; the final number of users reached is the average of the numbers of users reached at each trial.

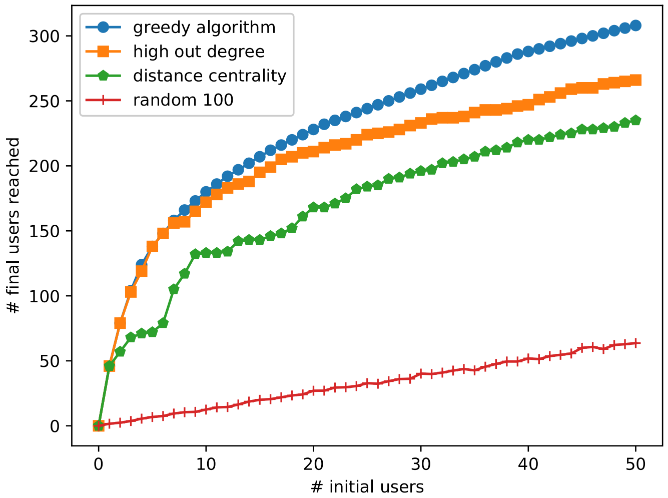


Fig. 9. Influence maximization illustrations: number of users reached at the end of the diffusion depending on the number of initial users.

In all our experiments, the diffusion process is run for 30 steps, which corresponds to the number of days in the dataset. Fig. 9 displays the result of the diffusion process using the RUC model and the different methods for initial users choice. As one can note, and unsurprisingly, the random choice is the worse method, as around 60 users are reached with 50 initial diffusers. The greedy algorithm outperforms the other methods over all ranges of κ, the difference being significant above a certain number of initial diffusers. When κ is very small, the greedy algorithm performs similarly to the higher out-degree and distance centrality heuristics, which can be explained by the fact it is important in this case to select highly connected nodes (hubs), which is done by all three methods.

7. Conclusion

We have introduced in this paper a new family of models for content diffusion in social networks which takes into account the content diffused through its similarity with user profiles. As we have seen, the models proposed differ from the ones previously proposed in that:

1. They integrate an additional, content-based dimension;
2. They explicit the different features used. In particular, both the social pressure and the willingness to diffuse are features which can be weighed differently, as opposed to what is done in IC-based (or LT-based models);
3. They rely on few parameters, which make them robust to the arrival of new users in the network, unlike models which require (at least) as many parameters as there are links in the network;
4. They are not equivalent to a bond percolation process on the social graph, contrary to IC-based, LT-based and SI-based models.

We have furthermore compared these models with previously proposed ones on two blog datasets. The results of these experiments show that taking into account the content of the information diffused leads to more precise predictions. Lastly, we have proven that the influence maximization problem is NP-hard for these new models (as is the case for previously proposed ones), and illustrated how of the greedy algorithm behaves compared to standard heuristics.

The use made of the content diffused in this study is restricted to a similarity function with user profiles, and the topical or semantic networks underlying this content have been ignored. Such

[4] Similar heuristics are used in [3]. Note that, because of the size of the social graphs considered here, the optimal solution can not be computed.

networks can be, at least partly, automatically inferred and could be of great help to predict the evolution of a content network and to develop tools to monitor them. In particular, it is very likely that certain contents are more diffused than others. Qualifying such contents, e.g. with respect to the topics covered, could lead to different models adapted to different types of information. This would help predict, for example, that pieces of information of general interest (as the results of presidential elections) tend to be largely diffused, whereas others are restricted to specific users. We believe that exploiting all these dimensions simultaneously is a promising avenue for the study of information diffusion in content networks.

Appendix A. Derivatives for estimation of model parameters

Let us recall the form of the likelihood, on the training set, for the RUC and DRUC models:

$$\mathcal{L}(\lambda_0,\lambda_1,\lambda_2,\lambda_3)=\prod_{k=1}^{l}\prod_{t=1}^{T^k}\left[\prod_{n_i\in C^k(t)}P(n_i,q^k,\leq t)\prod_{n_i\notin C^k(t)}(1-P(n_i,q^k,\leq t))\right]$$

As mentioned before, we make use of the recurrence equation (Eq. (3)) to compute the partial derivatives, and store, for each user, the current values of $P(n_i,\ q^k,\ \leq t)$ and its derivative at each timestamp:

$$\frac{\partial P(n_i,q^k,\leq t+1)}{\partial\lambda_i}=\frac{\partial P(n_i,q^k,\leq t)}{\partial\lambda_i}(1-P(n_i,q^k,t+1)) \quad \text{(A.1)}$$

$$+\frac{\partial P(n_i,q^k,t+1)}{\partial\lambda_i}(1-P(n_i,q^k,\leq t)) \quad \text{(A.1)}$$

The derivatives of the equations for the update of the probabilities at each step, if $SP(n_i,\mathcal{G},M^k,t)>0$, are:

$$\frac{\partial P(n_i,q^k,t+1)}{\partial\lambda_0}=\frac{\left(1+\frac{\partial E[|C^k(n_i,t)|]}{\partial\lambda_0}\right)e^{-\lambda_0-\lambda_1(S(n_i,q^k;\theta_s)-\lambda_2E[|C^k(n_i,t)|]-\lambda_3W(n_i;\theta_w)}}{(1+e^{-\lambda_0-\lambda_1(S(n_i,q^k;\theta_s)-\lambda_2E[|C^k(n_i,t)|]-\lambda_3W(n_i;\theta_w)})^2} \quad \text{(A.2)}$$

$$\frac{\partial P(n_i,q^k,t+1)}{\partial\lambda_1}=\frac{\left(S(n_i,q^k;\theta_s)+\frac{\partial E[|C^k(n_i,t)|]}{\partial\lambda_1}\right)(e^{-\lambda_0-\lambda_1(S(n_i,q^k;\theta_s)-\lambda_2E[|C^k(n_i,t)|]-\lambda_3W(n_i;\theta_w)})}{(1+e^{-\lambda_0-\lambda_1(S(n_i,q^k;\theta_s)-\lambda_2E[|C^k(n_i,t)|]-\lambda_3W(n_i;\theta_w)})^2} \quad \text{(A.3)}$$

$$\frac{\partial P(n_i,q^k,t+1)}{\partial\lambda_2}=\frac{\left(E[|C^k(n_i,t)|]+\frac{\partial E[|C^k(n_i,t)|]}{\partial\lambda_2}\right)(e^{-\lambda_0-\lambda_1(S(n_i,q_k;\theta_s)-\lambda_2E[|C^k(n_i,t)|]-\lambda_3W(n_i;\theta_w)})}{(1+e^{-\lambda_0-\lambda_1(S(n_i,q_k;\theta_s)-\lambda_2E[|C^k(n_i,t)|]-\lambda_3W(n_i;\theta_w)})^2} \quad \text{(A.4)}$$

$$\frac{\partial P(n_i,q^k,t+1)}{\partial\lambda_3}=\frac{\left(W(n_i;\theta_w)+\frac{\partial E[|C^k(n_i,t)|]}{\partial\lambda_3}\right)(e^{-\lambda_0-\lambda_1(S(n_i,q^k;\theta_s)-\lambda_2E[|C^k(n_i,t)|]-\lambda_3W(n_i;\theta_w)})}{(1+e^{-\lambda_0-\lambda_1(S(n_i,q^k;\theta_s)-\lambda_2E[|C^k(n_i,t)|]-\lambda_3W(n_i;\theta_w)})^2} \quad \text{(A.5)}$$

If $SP(n_i,\mathcal{G},M^k,t)=0$, then $\frac{\partial P(n_i,q^k,t+1)}{\partial\lambda_i}=0$.

Table A.1
Comparison of RUC and RUC without similarity on the *Dense-ICWSM* dataset.

Model	RUC	RUC no sim
MAP	**0.817**	0.780
Relative volume error	**0.901**	1.842

Appendix B. Analysis of the similarity parameter

We present here the results for a version of the RUC model without taking into account the similarity between the content and the user profile (where λ_1 is set to 0).

Table A.1 shows the average precision and the relative volume error for both RUC and a RUC version without the similarity parameter. The results of RUC with λ_1 learned are better for the average precision. The average precision is computed over all the recall points. A system can still have a relatively high average precision if the last recall point is not ranked too low in the list of diffusers.

To assess this last part, one can use the relative volume error. As we can see, the relative volume error is important when λ_1 is set to 0, showing that this model over diffuses.

References

[1] J. Goldenberg, B. Libai, E. Muller, Talk ofthe network: a complex systems look at the underlying process of word-of-mouth, Market. Lett. 12 (2001) 211–223.
[2] M. Granovetter, Threshold models of collective behavior, Am. J. Sociol. 83 (6) (1978) 1420–1443.
[3] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, 2003, pp. 137–146.
[4] D.M. Romero, B. Meeder, J. Kleinberg, Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter, in: Proceedings of the 20th International Conference on World Wide Web, in: WWW '11, ACM, New York, NY, USA, 2011, pp. 695–704, doi:10.1145/1963405.1963503.
[5] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., New York, NY, USA, 1986.
[6] J. Leskovec, M. McGlohon, C. Faloutsos, N.S. Glance, M. Hurst, Patterns of cascading behavior in large blog graphs, in: Proceedings of the SDM, 2007.
[7] H.P. Young, Innovation diffusion in heterogeneous populations: contagion, social influence, and social learning, Am. Econ. Rev. 99 (5) (2009) 1899–1924.
[8] K. Munshi, Social learning in a heterogeneous population: technology diffusion in the indian green revolution, J. Dev. Econ. 73 (1) (2004) 185–213.
[9] A. Guille, H. Hacid, C. Favre, D.A. Zighed, Information diffusion in online social networks: a survey, SIGMOD Rec. 42 (2) (2013) 17–28, doi:10.1145/2503792.2503797.
[10] H. Trottier, P. Philippe, Deterministicmodeling of infectious diseases: theory and methods, Internet J. Infect. Diseases 1 (2) (2001).
[11] F. Brauer, C. Castillo-Chavez, Mathematical Models in Population Biology and Epidemiology, Springer, 2001.
[12] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2) (2003) 167–256.
[13] D. López-Pintado, Diffusion in complex social networks, Games Econ. Behav. 62 (2) (2008) 573–590.
[14] M. Kimura, K. Saito, R. Nakano, Extracting influential nodes for information diffusion on a social network, Proc. Nat. Conf. Artif. Intell. 22 (2) (2007) 1371.
[15] F.D. Sahneh, C. Scoglio, Competitive epidemic spreading over arbitrary multilayer networks, Phys. Rev. E 89 (2014) 062817, doi:10.1103/PhysRevE.89.062817.
[16] N. Antulov-Fantulin, A. Lancic, H. Štefancic, M. Šikic, T. Šmuc, Statistical inference framework for source detection of contagion processes on arbitrary network structures, in: Proceedings of the IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems Workshops, in: SASOW '14, IEEE Computer Society, 2014, pp. 78–83, doi:10.1109/SASOW.2014.35.
[17] K. Saito, R. Nakano, M. Kimura, Prediction of information diffusion probabilities for independent cascade model, in: Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part III, in: KES '08, Springer-Verlag, 2008, pp. 67–75.
[18] K. Saito, M. Kimura, K. Ohara, H. Motoda, Learning continuous-time information diffusion model for social behavioral data analysis, Learning 5828 (2009) 322–337.
[19] D. Liben-Nowell, J. Kleinberg, Tracing information flow on a global scale using Internet chain-letter data, Proc. Nat. Acad. Sci. 105 (12) (2008) 4633–4638.
[20] A. Goyal, F. Bonchi, L.V. Lakshmanan, Learning influence probabilities in social networks, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, in: WSDM '10, ACM, 2010, pp. 241–250.

[21] M. Gomez-Rodriguez, D. Balduzzi, B. Schölkopf, Uncovering the temporal dynamics of diffusion networks, in: L. Getoor, T. Scheffer (Eds.), Proceedings of the 28th International Conference on Machine Learning (ICML), ICML '11, ACM, 2011, pp. 561–568.

[22] S.A. Myers, J. Leskovec, On the convexity of latent social network inference, in: Proceedings of the 23rd International Conference on Neural Information Processing Systems, NIPS'10, vol. 2, 2010, pp. 1741–1749.

[23] M. Gomez-Rodriguez, J. Leskovec, A. Krause, Inferring networks of diffusion and influence, ACM Transactions on Knowledge Discovery from Data (TKDD) 5 (4) (2012) Article No. 21.

[24] H. Daneshmand, M. Gomez-rodriguez, L. Song, Estimating diffusion network structures: Recovery conditions, sample complexity and soft-thresholding algorithm, in: Proceedings of the 31st International Conference on Machine Learning (ICML), 2014.

[25] J. Pouget-Abadie, T. Horel, Inferring graphs from cascades: A sparse recovery framework, in: Proceedings of the 32nd International Conference on Machine Learning, (ICML), 2015, pp. 977–986.

[26] L. Weng, J. Ratkiewicz, N. Perra, B. Gonçalves, C. Castillo, F. Bonchi, R. Schifanella, F. Menczer, A. Flammini, The role of information diffusion in the evolution of social networks, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '13, ACM, 2013, pp. 356–364, doi:10.1145/2487575.2487607.

[27] I. Taxidou, P.M. Fischer, Online analysis of information diffusion in twitter, in: Proceedings of the 23rd International Conference on World Wide Web, in: WWW '14 Companion, ACM, 2014, pp. 1313–1318.

[28] T. Schelling, Dynamic models of segregation, J. Math. Sociol. 1 (1971) 143–186.

[29] M. Granovetter, R. Soong, Threshold models of diversity: Chinese restaurants, residential segregation, and the spiral of silence, Sociol. Methodol. 18 (1988) 69–104.

[30] M.W. Macy, Chains of cooperation: threshold effects in collective action, Am. Sociol. Rev. 56 (6) (1991) 730–747.

[31] T.W. Valente, Network Models of the Diffusion of Innovations (Quantitative Methods in Communication Subseries), Hampton Press, NJ, 1995.

[32] T. Valente, Social network thresholds in the diffusion of innovations, Soc. Netw. 18 (1) (1996) 69–89.

[33] E. Abrahamson, L. Rosenkopf, Social network effects on the extent of innovation diffusion: a computer simulation, Org. Sci. 8 (3) (1997) 289–309.

[34] M. Richardson, P. Domingos, Mining knowledge-sharing sites for viral marketing, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data mining, in: KDD '02, ACM, 2002, pp. 61–70.

[35] P. Dodds, D. Watts, Universal behavior in a generalized model of contagion, Phys. Rev. Lett. 92 (21) (2004).

[36] D. López-Pintado, D.J. Watts, Social influence, binary decisions and collective dynamics, Rationality Soc. 20 (4) (2008) 399–443.

[37] A. Borodin, Y. Filmus, J. Oren, Threshold models for competitive influence in social networks, in: Proceedings of the WINE, Springer, 2010, pp. 539–550.

[38] J. Yang, J. Leskovec, Modeling information diffusion in implicit networks, in: Proceedings of the IEEE International Conference on Data Mining, 2010, pp. 599–608.

[39] D. Kempe, J. Kleinberg, E. Tardos, Influential nodes in a diffusion model for social networks, in: Proceedings of the ICALP, Springer Verlag, 2005, pp. 1127–1138.

[40] K. Saito, K. Ohara, Y. Yamagishi, M. Kimura, H. Motoda, Learning diffusion probability based on node attributes in social networks., in: M. Kryszkiewicz, H. Rybinski, A. Skowron, Z.W. Ras (Eds.), ISMIS, *Lecture Notes in Computer Science*, vol. 6804, Springer, 2011, pp. 153–162.

[41] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '01, ACM, 2001, pp. 57–66.

[42] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost–effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD 07, ACM, 2007, pp. 420–429.

[43] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher, An analysis of approximations for maximizing submodular set functions-i, Math. Program. 14 (1) (1978) 265–294.

[44] M. Gomez-Rodriguez, B. Schölkopf, Influence maximization in continuous time diffusion networks, in: Proceedings of the 29th International Conference on Machine Learning (ICML), 2012.

[45] N. Du, L. Song, M. Gomez Rodriguez, H. Zha, Scalable influence estimation in continuous-time diffusion networks, in: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 3147–3155.

[46] P. Bao, H.-W. Shen, J. Huang, X.-Q. Cheng, Popularity prediction in microblogging network: a case study on sina weibo, in: Proceedings of the 22nd International Conference on World Wide Web, in: WWW '13 Companion, ACM, 2013, pp. 177–178, doi:10.1145/2487788.2487877.

[47] N. Du, Y. Liang, M. Balcan, L. Song, Influence function learning in information diffusion networks, in: T. Jebara, E.P. Xing (Eds.), Proceedings of the 31st International Conference on Machine Learning (ICML), JMLR Workshop and Conference Proceedings, 2014, pp. 2016–2024.

[48] C. Lagnier, L. Denoyer, É. Gaussier, P. Gallinari, Predicting information diffusion in social networks using content and user's profiles, in: Proceedings of the ECIR, 2013.

[49] K. Burton, A. Java, I. Soboroff, The ICWSM 2009 Spinn3r dataset, in: Proceedings of the Third Annual Conference on Weblogs and Social Media (ICWSM), 2009.

[50] J. Leskovec, L. Backstrom, J. Kleinberg, Meme-tracking and the dynamics of the news cycle, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '09, ACM, 2009, pp. 497–506.

[51] C.D. Manning, P. Raghavan, H. Schütze, An Introduction to Information Retrieval, Cambridge University Press, 2008.

[52] R. Karp, Reducibility among combinatorial problems, in: Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.