Predicting Information Diffusion on Social Networks with Partial Knowledge

Anis Najar
Laboratoire d'informatique de Paris 6
University Pierre et Marie Curie- UPMC
Paris, France
anis.najar@lip6.fr

Ludovic Denoyer
Laboratoire d'informatique de Paris 6
University Pierre et Marie Curie - UPMC
Paris, France
ludovic.denoyer@lip6.fr

Patrick Gallinari
Laboratoire d'informatique de Paris 6
University Pierre et Marie Curie - UPMC
Paris, France
patrick.gallinari@lip6.fr

ABSTRACT

Models of information diffusion and propagation over large social media usually rely on a *Close World Assumption*: information can only propagate onto the network relational structure, it cannot come from external sources, the network structure is supposed fully known by the model. These assumptions are nonrealistic for many propagation processes extracted from Social Websites. We address the problem of predicting information propagation when the network diffusion structure is unknown and without making any closed world assumption. Instead of modeling a diffusion process, we propose to directly predict the final propagation state of the information over a whole user set. We describe a general model, able to learn predicting which users are the most likely to be contaminated by the information knowing an initial state of the network. Different instances are proposed and evaluated on artificial datasets.

Categories and Subject Descriptors

I.m [**Computing Methodologies**]: Miscellaneous

Keywords

Diffusion, Social Networks, Machine Learning

1. INTRODUCTION

The diffusion and propagation of information over large social media has been an active research domain recently. Propagation models are often inspired from earlier work on epidemiology or marketing. Most of them consider that a node is either active or inactive and that active nodes can contaminate or propagate the information to other nodes. The different models differ on the assumptions upon the way information spreads from one node to another. Besides providing models for the propagation process, these techniques can be used for tasks like opinion leader detection [3]. Most models rely on strong assumptions:

1. the propagation network is completely known,

2. information can only propagate onto this network and cannot come from external sources,

3. only one type of information is considered.

For many practical cases, these assumptions are hardly met. The network might be partly known or even completely unknown. Most often, it is impossible to characterize the whole network due to its size and to the difficulty of tracking the different events characterizing the relational structure. In social networks, information often comes from external sources so that the same information may appear at different places and times in the network without being propagated through the network [1]. Users in a network will propagate information differently according to their profile or to their domain of interest. They will then play different roles according to the nature of the information to be propagated. Models generally attempt to mimic the diffusion/ propagation behavior at the node level in order to reproduce cascades of information observed at different places in the network or to reproduce the global contagion mechanism. Their main goal is then to explain or analyze the actual diffusion process.

We consider here the problem of information propagation prediction on social media: given a state of contamination of the network at a given time t what will be the state of contamination at time $t' > t$? This is related to, but different from the problem of diffusion modeling. For the former, the goal is to predict the state of the network at a given time, while for the later, the goal is to model each step of the diffusion process. We present a prediction model which does not rely on hypothesis 1 and 2. The model does not require the knowledge of the network structure, although it could take benefit from a partial or complete knowledge of this structure. This model is based on a regression framework. It can incorporate the effect of external information sources so that the information is not restricted to propagate onto the network only. Extensions of this model can also deal with multiple sources (hypothesis 3) but are not detailed here. The paper is organized as follows. In Section 2 we introduce notations and define the prediction task. In Section 3 we present our model. In Section 4 we describe large scale experiments made to evaluate the model. In section 5 we review related work.

2. NOTATIONS AND TASKS DEFINITION

2.1 Notations

We introduce here notations used throughout the paper.

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

- A social network is modeled as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = (n_1, ...n_N)$ is a set of nodes or users, and $\mathcal{E} = \{e_{i,j} \in [0;1]\}$ denotes edges representing relations between users, such that $e_{i,j}$ is the weight of the relation between nodes. i and j, $e_{i,j} = 0$ means that there is no link between user i and user j.

- Information propagation is modeled here as a discrete process, so that at each step of this process, the network state may be represented as a vector of contamination. Let M^k be a contamination matrix representing the propagation of an information[1]:

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

  $m^k_{i,t}$ is the contamination of user i at time t. Classically $m^k_{i,t} \in 0;1$, i.e. user is contaminated or not. We will also consider the case $m^k_{i,t} \in [0;1]$ when our knowledge about the contamination is uncertain. T^k corresponds to the duration of the contamination process: after time T^k no more individuals will be contaminated. We use here a relative datation i.e $t = 1$ corresponds to the first time a user has been contaminated for information k - e.g. the date of the first apparition of a particular *tweet* on a microblog site for example and T^k is the time where the propagation has finished. For a given network, different information cascades will be observed corresponding to different M^ks.

The model parameters will be estimated from samples of propagation cascades. We denote by $\left(M^1, ..., M^\ell\right)$ a set of training propagation matrices used for estimating the model parameters and by $\left(M^{\ell+1}, ..., M^M\right)$ a set of test matrices used for evaluation. For example, $\left(M^1, ..., M^\ell\right)$ may correspond to past observations and $\left(M^{\ell+1}, ..., M^M\right)$ to future observations to be predicted.

2.2 Prediction task

Existing propagation models are used to model *how information spreads* over a social network. These models may also be used for predicting the information propagation: given an initial state of the network, the model is run and predicts at each step the propagation at each node. They usually make *Closed World Assumptions 1 and 2*: they consider that the diffusion network is known and that information can only propagate through the network without interaction from the external world. In many cases, this is not realistic. We propose here a prediction model which does not rely on these assumptions. For this, we will focus on the following task: predict the final contamination state of the network given an initial contamination. This task amounts at learning a correspondence between the initial and final states of the network without considering the intermediate steps:

$$\left(\mathcal{G}, \mathbf{m}_1^k\right) \Rightarrow m^k_{1,T^k} \qquad (1)$$

where $\mathbf{m}_1^k = (m^k_{1,1}, ..., m^k_{N,1})^T$ [2] is the vector of initial contamination i.e the contamination of all the users by a given information the first time this information appears in observed data, $\mathbf{m}^k_{T^k} = m^k_{1,T^k}, ..., m^k_{N,T^k})^T$ is the vector representing the final state of contamination we want to predict, $\mathcal{G}$ denotes the social network, which might be partially observed or even completely unknown.

3. PROPOSED APPROACH

3.1 General Model

The proposed approach directly predicts the final contamination values without going through the modeling of the whole diffusion process at each time step and at each node like most models do. To compare the two methodologies, one can make an analogy with the predictive and modeling or generative approaches for discrimination or regression problems. Predictive approaches take a direct route to the prediction problem, while modeling ones learn the generative process of the data. Both approaches have their own advantages: generative methods work better when able to capture the real data distribution or when only few training data is available. If there is no hint about the distribution of the data - predictive models generally achieve a better performance. We will come back to this point in the experiments section. As far as we know, this model is the first attempt to use direct predictive models in the context of information diffusion.

Let us denote by f_θ a parameterized regression model where θ is a set of parameters to be learned on the training set. f_θ will be trained to associate a final contamination state, to any initial contamination state:

$$f_\theta\left(\mathcal{G}, \mathbf{m}_1^k\right) \Rightarrow m^k_{1,T^k} \qquad (2)$$

Different types of predictors may be used. We focus here on a family of predictors with the following form:

$$f_{\theta,j}(\mathbf{m}_1) = g\left(\sum_{i=1}^{N} w_{j,i}\theta_{j,i}m_{i,0}\right) \qquad (3)$$

where $w_{j,i}$ is a predefined parameter that may be used to represent the graph structure of the network when available, g is a transfer function and $\{\theta_{i,j}\}_{i,j\in[1;N]^2}$ is the set of parameters to be learned.

3.2 Instances of the General Model

We present here different variants of the model that have been used in the tests presented in the experimental section. They correspond to different assumptions made on the propagation schema that has to be learned.

- Linear Model (LM) The simplest model is a classical linear regression:

$$f^{LM}_{\theta,j}(\mathbf{m}_1) = \sum_{i=1}^{N} \theta_{j,i}m_{i,1} \qquad (4)$$

[1] For simplification, we consider that all information follow the same propagation process (assumption (iii), i.e. we do not differentiate the propagation according to the message content.

[2] T is the transpose operator

This model learns one *influence weight* $\theta_{j,i}$ between each couple of users (u_j, u_i), these weights have real values and can represent either positive or negative influence between users. The number of parameters is N^2.

- Logistic Model (LoM) LoM is the logistic version of the LM model. It can be written as:

$$f_{\theta,j}^{LoM}(\mathbf{m}_1) = logit\left(\sum_{i=1}^{N} \theta_{j,i} m_{i,1}\right) \quad (5)$$

where *logit* is the classical logistic function. This model forces the predicting contamination to be between 0 and 1. The number of parameters is also N^2

- Positive Linear Model (PLM) The positive linear model is a constrained version of the linear model where all *influence weights* are forced to be positive. Here we have used the following implementation of the constraint:

$$f_{\theta,j}^{PLM}(\mathbf{m}_1) = \sum_{i=1}^{N} \theta_{j,i}^2 m_{i,1} \quad (6)$$

where the $\theta_{j,i}$ are as before real values. The influence of a user over another one can only be positive here The number of parameters is again N^2. The PLoM model is the Logistic equivalent to this model.

- Graph Based Positive Linear Model (GPLM) None of the above model considers the structure of the social network $\mathcal{G}$. The knowledge of this structure may be easily taken into account in our general predictive formulation. In the case of the PLM model, this variant, denoted GPLM, takes the following form:

$$f_{\theta,j}^{GPLM}(\mathbf{m}_1) = \sum_{i=1}^{N} w_{j,i} \theta_{j,i}^2 m_{i,1} \quad (7)$$

where $w_{j,i}$ is the weight of the edge between u_i and u_j in the network. When there is no edge, this weight is 0 and there is no propagation between the graph nodes.

GPLM thus restricts the propagation to the known graph structure. One advantage of this model is that it only learns $|\mathcal{E}|$ parameters instead of N^2, resulting in a faster algorithm. A drawback is that when the closed world assumption is false, it will usually do worse than the more general models above.

3.3 Learning

Learning the predictor is performed classically by minimizing a loss function on the training set over the model parameters. Let $\Delta(a, b)$ measure the cost of predicting a when the target is b, the global loss function takes the form:

$$\mathcal{L}(\theta) = \sum_{k=1}^{l} \Delta(f_\theta(\mathbf{m}_1^k), \mathbf{m}_{T_k}^k) + \lambda \|\theta\|^2$$

where l is the number of cascades (examples) in the training set, $\mathbf{m}_1^l$ is the initial state of the network for cascade l, $\mathbf{m}_{T_k}^l$ is the corresponding target state, and $\lambda \|\theta\|$ is a regularization term. Here again different loss functions $\Delta(.,.)$ could be used. In the experiments we have been using a classical square loss. The training problem then amounts at solving

$$\theta^* = argmin_\theta \mathcal{L}(\theta)$$

This is solved using a gradient-descent method. While most of our models have a complexity $O(N^2)$, note that this computation can be easily done using $GPU - based$ computers, resulting in models that are able to learn quickly with a very large amount of training data.

3.4 Complexity

We give here an overview of the learning complexity of the different models and discuss the consequences. Concerning the general model, we aim at learning one parameter for each pair of users resulting in N^2 parameters. On a network of 1,000 users, it means that our approach needs the evaluation of 1 million parameters. This is a major drawback of the proposed model which resulting complexity is $\mathcal{O}(N^2)$ and thus which cannot be applied on very large networks, even using implementation tricks or GPU. The GPLM model, which is based on the structure of the graph as a lower complexity of $\mathcal{O}(|\mathcal{E}|)$ where $|\mathcal{E}|$ is the number of edges in the original network. It is easier and faster to train, but it cannot modeled complex diffusion processes due to the closed world assumption it is based on. The work presented here is a preliminary work and we plan to study different way to reduce the complexity of the general model. The first simple idea is to consider a GPLM model that learns one parameter for each pair of users that are connected by a path of length maximum $= L$. In this case, the number of parameters to estimate is greater than $|\mathcal{E}|$, but lower than N^2 - depending on the structure of the graph - and the model is able to learn long-term propagation. In that case, the value of $L > 1$ is used to determines the mix between the complexity and the expression power of the model. The other perspective is to rewrite the model using sparse L_1 regularizers that will encourage the algorithm to find a sparse solution, where many of the $\theta_{.,.}$ parameters will be set to 0. This can be done by writting the objective function as:

$$\mathcal{L}(\theta) = \sum_{k=1}^{l} \Delta(f_\theta(\mathbf{m}_1^k), \mathbf{m}_{T_k}^k) + \lambda|\theta|$$

where λ is the meta-parameter that is used to choose the sparsity of the model. The higher λ, the more sparse and fast will be the resulting solution. This solution will be explored in a future paper.

4. EXPERIMENTS

4.1 Datasets

Experiments have been performed using artificially generated cascades over real social networks. Since capturing real representative cascades is not trivial, this allows performing extensive experiments on many different situations and to compare with baseline propagation models in situations where they can be used. We have captured social graphs from different Web sites - We provide here results for two of these sites: UsAir97 (direct flights between airports) and PolBlogs (Political blogs). The statistics for the two sites are provided in Table 1 - these graphs are then representative of real diffusion structures. We have then used classical Independent Cascade models (ICM) and Linear Threshold Models (LTM)in order to generate artificial cascades over these structures. Different structures and different parameterization of the *Independent Cascade Model (IC)* and the

Network	Nb. Nodes	Nb. Links
UsAir97	332	2 126
PolBlogs	1 493	19 091

Table 2: Statistics over the UsAir97 and PolBlogs datasets

Linear Threshold Model (LTM) [3] have been used in order to provide a variety of training and testing situations. The generated cascades are then considered as the gold standard and the goal is to predict the propagation on these data. For each experiment, the ICM or LTM model is used to generate 2 000 contamination matrices: 1 000 for training and 1 000 for testing. From these graphs we have also extracted partial graphs by keeping $50\%, 75\%$ et 100% of the original nodes - 100% corresponds to the full network. This will allow us to compare generative ICM and LTM approaches with predictive models in situations where the known graph only imperfectly reflects the true diffusion structure usec for generating the data and also to analyze how the ICM and LTM models degrade when the graph structure is only imperfectly known. These partial graphs are generated as follows. Out of a complete graph $\mathcal{G}$ with N nodes, one selects $N' \leq N$ nodes and build a subgraph $\mathcal{G}'$ consisting of these nodes plus the edges between the nodes from the original graph $\mathcal{G}$.

By varying the parameters of the models and of the generation processes, we have performed a large set of experiments. We will present here only some representative results. Note that the behavior of the models is very similar on the different datasets.

4.2 Evaluation

Prediction models produce scores at each node. The predictive models directly produce real final contamination scores. ICM and LTM models can be used with Monte Carlo simulation in order to estimate the final probability for each user to be contaminated: starting from an initial contamination state for a given graph, the diffusion process for the model is simulated on the graph structure until it stabilizes and produces the final contamination state. LTM models are deterministic so that they produce only one final state for a given initial condition. The contaminated nodes will have a value of 1 while for the others it will be 0. ICM models are stochastic (see Section 5) so that different runs from the same initial state will produce different propagation values on the nodes. For estimating the node scores, 1000 propagations are run for the same initial contamination state and the scores obtained at each node after stabilization are averaged over all the runs, giving a probability of contamination. The scores produced are thus real values which will play the same role as the scores obtained with the predictor models.

For the performance measure we have been using precision-recall (P-R) curves [5]. The node scores obtained with a given model are ordered in decreasing order of their values (in the case of binary scores all nodes with score 1 are above the nodes with score 0), and then P-R curves are computed from these ranked lists as it is classically done for example for lists returned by search engines. This avoids defining decision thresholds for the contamination values and provides a richer information on the systems' behavior. P-R curves reflect the ability of the prediction model to produce a high rank for users that are susceptible to be contaminated given an initial network state.

4.3 Experiments

Figure 1 illustrates the performance of the predictive models for the UsAir97 dataset for cascades generated by an ICM model and Figure 2 for the Polblogs dataset with cascades generated with an LTM model. In all the figures, are plotted the R/P curves for the model used for generating the cascade (respectively ICM and LTM), for the best predictive model, for the alternative generative model (LTM if data have been generated via ICM an vice versa). For comparison, we have also plotted the performance of a random prediction model which predicts a random score and of an Identity model which predicts the initial input state. Figure 1-1 to 1-3 give respectively the performance for the partial graphs with 50% and 75% of the initial graph nodes and for the complete graph. In all cases, the predictive models does not take into account the graph structure and learns or predict only by considering the initial and final contaminations.

For ICM generated data, the predictive model is almost as good as the ICM model for the complete graph (Figure 1-3). ICM performance slightly degrades on partial graphs but remains good meaning that on these datasets, ICM is robust to a degradation of the graph structure. The predictive model performance on the other sides does not degrade and progressively becomes higher than the one of the generating model (Figures 1-2 and 1-1). The less is known about the graph, the higher is the difference between the two models. Although the predictive model has learned on complete graphs, it is extremely robust to missing information and predicts well in all situations. The performance of the best LTM model is rather low for all the situations: LTM cannot predict cascades generated by ICM and is sometimes worse at that than the simple identity model.

Figure 3-1, 3-2 and 3-3 compare different predictive models on the ICM and LTM generated cascades. Models with positive constraints behave better than unconstrained predictors. There is no negative interaction between nodes in all the data generated for the experiments and the positivity constraints help the model to learn solutions that generalize better. Note that our predictive models are however able to learn both positive and negative interactions which is an interesting property since negative interactions do happen in many cases [2]. ICM or LTM inspired models can also handle negative interactions [2], however they must be adapted for such case, whereas the same predictive model can handle both cases since it only relies on the data to be predicted and does not make any hypothesis on the way it propagates.

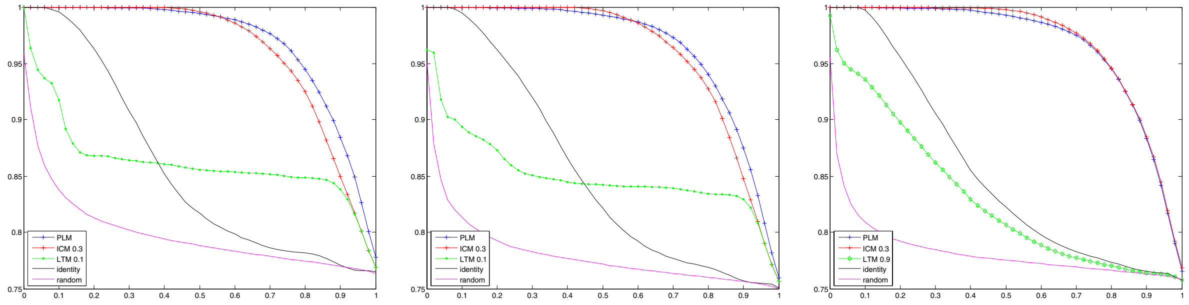
Table 2 provides F1 scores for the UsAir data set for the different models. It can be seen that predictive models are better than ICM or LTM as soon as the network information is incomplete and close to the performance of the model used for generating the data when the full graph is known to this model.

Another set of experiments has been performed by incorporating the knowledge of the graph or of the partial graph in the predictive model. Figures 5-1 to 5-3 compare prediction with and without this graph knowledge for predictive models. Performances are quite similar for both models for all cases and even slightly lower sometimes in the case of

[3] A description of the IC and LTM models used in the experiments is provided in the Section 5

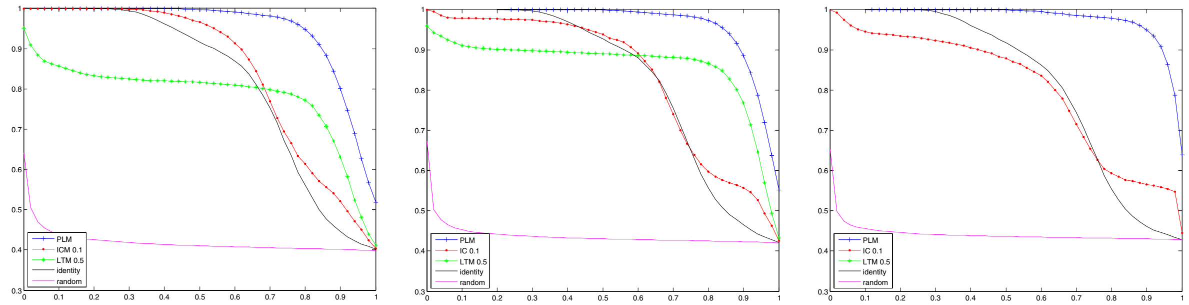
Prediction Model	Partial Network 50%	Partial Network 75 %	Full Network
IC 0.1%	62.7	64.9	74.7
IC 0.3%	83.1	85.2	**89.0**
IC 0.5%	87.0	87.1	87.4
LTM 0.1%	86.9	86.9	86.1
LTM 0.3%	69.9	74.3	71.4
LTM 0.5%	47.2	47.6	44.1
Identity	36.4	37.6	37.0
Random	60.2	59.9	60.0
LM	87.8	86.6	86.6
PLM	**88.7**	88.2	88.6
LoM	**88.7**	**88.8**	88.1
PLoM	86.4	85.6	86.1

Table 1: F1 measure on the UsAir97 Corpus with a generating model IC 0.3 (i.e. the probability for a node to activate any of its neighbors is 0.3 - see Section 5)



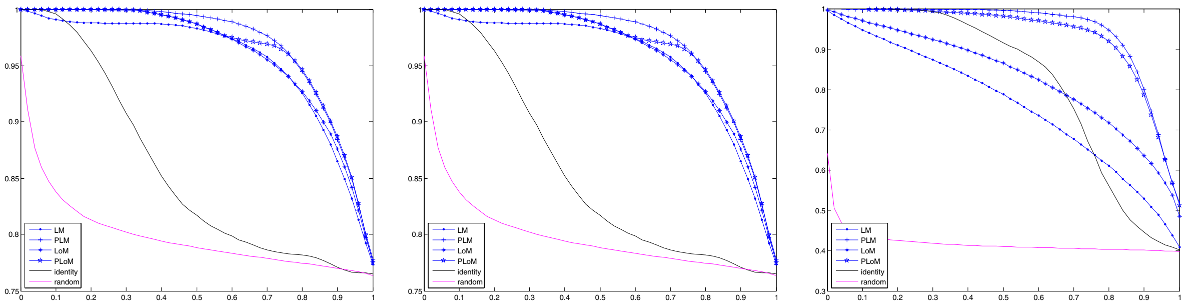
Partial Network 50% Partial Network 75 % Full Network

Figure 1: P-R Curves obtained on the UsAir97 network, with a simulation model IC 0.3 for different size of the partial network. Only the best model of each family(IC,LTM and Discriminant) has been illustrated.



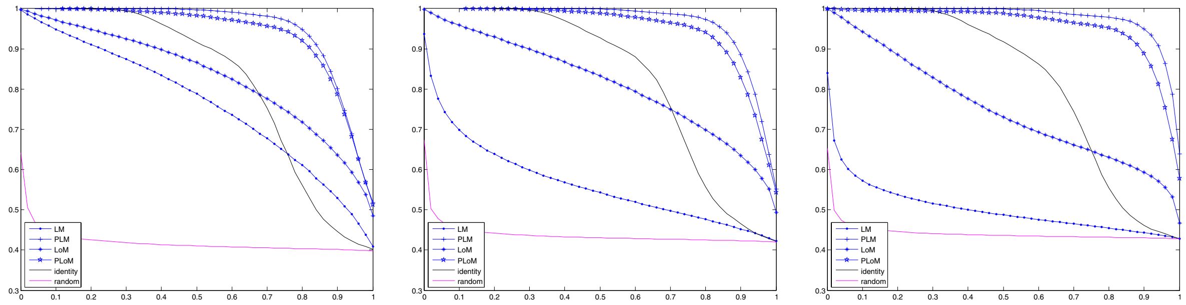
Partial Network 50% Partial Network 75 % Full Network

Figure 2: P-R Curves obtained on the PolBlogs network, with a simulation model LTM 0.5 for different size of the partial network (i.e. the threshold for being contaminated at a node is 0.5 - see Section 5). Only the best model of each family(ICM,LTM and Discriminant) has been illustrated. For the full network, LTM 0.5 gives perfect results.



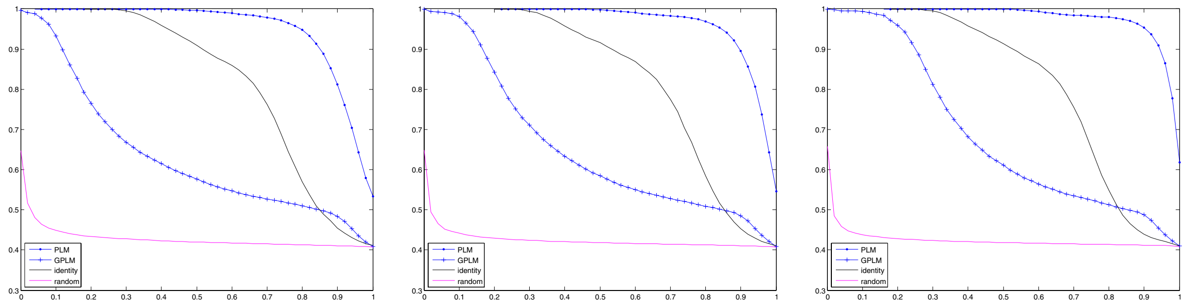
Partial Network 50% Partial Network 75 % Full Network

Figure 3: P-R Curves obtained on the UsAir97 network, with a simulation model IC 0.3 for different size of the partial network and for the different discriminant models described in the paper.



Partial Network 50% Partial Network 75 % Full Network

Figure 4: P-R Curves obtained on the PolBlog network, with a simulation model LTM 0.5 for different size of the partial network and for the different discriminant models described in the paper.



Partial Network 50% Partial Network 75 % Full Network

Figure 5: P-R Curves obtained on the PolBlogs network, with a simulation model LTM 0.5 for different size of the partial network. Comparison between PLM and the graph based GPLM

partial graphs. There is no performance gain obtained by exploiting the graph structures for predictive models. However the complexity of the graph-based predictive model is lower than the one of full predictive models which might become advantageous in the case of large or very large graphs.

5. RELATED WORK

Most LTM and ICM inspired models make use of predefined parameters. Recently, some papers have proposed to learn the model parameters from data using maximum likelihood [9, 4, 8]. An interesting problem also recently addressed by learning from observations is the inference of diffusion networks [7, 6]. Finally, modeling the diffusion over unknown networks is addressed in [10], this paper focuses on modeling the temporal dynamics of the diffusion and global statistics like the volume of infection.

We provide here a brief description of the Independent Cascade (ICM) and Linear Threshold (LTM) models used in this paper. ICM and LTM are two basic reference models which have been widely studied and for which many extensions have been considered. [3] describes a unified view of these models and several extensions. Both models operate on a directed graph G. A node may be *active* or *inactive*. Starting from an initial set of active nodes, a discrete process is unfolded in time where at each time step more nodes may become active under the influence of their neighbours. In both models, an active nodes remains active, although in variants or in related models, a node may recover and become again inactive? Information propagates on the graph until no more node can become active.

ICM operates in a push mode. It start from a set of active nodes $A(0)$. When a node v becomes active at time t, it will get a unique chance to active each of its neighbours w. w will become active at time $t+1$ according to a probability $p_{v,w}$. Whether or not w becomes active, v is not allowed to attempt activating w in later steps. The $p_{v,w}$ are parameters of the model. In the experiments performed here, all nodes have the same probability p to contaminate their neighbors. ICM 0.3 for example will denote a model with $p = 0.3$.

LTM operates on a pull mode. Each node v is given a threshold t_v which may be chosen at random uniformly in $[0, 1]$, although related models use fixed threshold values. Edges (w, v) in G, with w a parent of v are weighted by a positive value b_{vw} such that $\sum_{w \in N(v)} b_{vw} \leq 1$. Starting from an initial set of active nodes A_0, the contamination process unfolds as follows: at time step t, if v is active, it remains so, otherwise it will become active if the weight sum of its parents in G is above its threshold t_v: $\sum_{w \in N(v)} b_{vw} \geq t_v$. The b_{vw} are parameters of the model. In the experiments all the nodes have the same threshold t and $LTM0.3$ denotes a model with $t = 0.3$.

6. CONCLUSION

For predicting the final information propagation state over information networks, we have introduced a new approach which directly aims at predicting this final state without modeling the whole diffusion process over the network. This approach does not make closed world assumptions familiar to most information diffusion models. We have proposed a general predictive model implementing this approach and different instances of this model. Tests have been performed on different artificially generated cascades over real social network structures. These experiments have shown that the predictive approach is able to learn predicting final contamination states from data generated by different models and outperforms these models as soon as the information about the network structure becomes unreliable. Future work will examine the behavior and performance of theses predictive models over real propagation processes observed from large size datasets.

7. AKNOWLEDGMENTS

This work was partially supported by the French National Agency of Research (ExDeuss/Cedres and MLVIS Projects).

8. REFERENCES

[1] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *In Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM).*

[2] Wei Chen, Alex Collins, Rachel Cummings, Te Ke, Zhenming Liu, David Rincón, Xiaorui Sun, Yajun Wang, Wei Wei, and Yifei Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. In *SDM*, pages 379–390, 2011.

[3] David Kempe, Jon Kleinberg, and Éva 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 - KDD '03*, page 137, New York, New York, USA, August 2003. ACM Press.

[4] Masahiro Kimura, Kazumi Saito, Kouzou Ohara, and Hiroshi Motoda. Learning information diffusion model in a social network for predicting influence of nodes. *Intell. Data Anal.*, 15(4):633–652, 2011.

[5] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval.* Cambridge University Press, 2008.

[6] Seth A. Myers and Jure Leskovec. On the convexity of latent social network inference. In *NIPS*, pages 1741–1749, 2010.

[7] Manuel Gomez Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 561–568, New York, NY, USA, June 2011. ACM.

[8] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. Prediction of information diffusion probabilities for independent cascade model. In *KES (3)*, pages 67–75, 2008.

[9] Kazumi Saito, Kouzou Ohara, Yuki Yamagishi, Masahiro Kimura, and Hiroshi Motoda. Learning diffusion probability based on node attributes in social networks. In Marzena Kryszkiewicz, Henryk Rybinski, Andrzej Skowron, and Zbigniew W. Ras, editors, *ISMIS*, volume 6804 of *Lecture Notes in Computer Science*, pages 153–162. Springer, 2011.

[10] Jaewon Yang and Jure Leskovec. Modeling information diffusion in implicit networks. In *ICDM*, pages 599–608, 2010.