Infectious Knowledge in Collaborative Social Sites

Eliana Feasley
Department of Computer Science
University of Texas at Austin
elie@cs.utexas.edu

Wesley Tansey
Department of Computer Science
University of Texas at Austin
tansey@cs.utexas.edu

Abstract

In recent years, the attention paid to cascading information in social networks has been increasing in a fashion itself comparable to a social cascade. This makes sense - the way that information infects different spaces of ideas has applications to basic graph theory, epidemiology, predictions of the future, and marketing. In this paper, we examine the dynamics of information spread across subcommunities with overlapping networks in several domains - the question-answering site StackExchange, a news network provided by memetracker, and a synthetic dataset of our own devising. Each of these is structured such that it is possible to track how ideas spread over time, and to discover semi-explicit communities and the connections between them.

1 Introduction

Social cascades capture the concept of new ideas, or *memes*, spreading across influence networks and infecting subcultures. Cascade theory has applications to several areas, including epidemiology, graph theory, machine learning, and marketing. In this project, we examine the dynamics of information spread across sub-communities with overlapping networks in several real-world domains: the social news site *reddit*, a collection of blogs and mainstream media [5], and the question-answer network *StackExchange*. Our goal is to capture the flow of memes by learning a graphical model for each domain. We learn a graphical model describing the transition between timesteps, effectively capturing how the memes cascade through the network.

Information online travels across networks in a variety of configurations, and it is easy to see information spreading across them. Past work such as **Cite** **Cite** and **Cite** all illustrate examples of phrases or terms spreading rapidly across networks. Our work is an extension of the work in [2], in which the authors learn the structure of a graph from observing cascades. These authors learn structure, while ee assume the graph is fully connected, and predict the state of a cascade given the inferred weights on the graph.

In this paper, we explore how modeling this mechanism as a timeseries can help us both to predict future topics and to discover the structure of latent communities. . Section 4.6 explains the structure of the data and the domains from which they are drawn. Section 2.1 delves into past work on cascades in networks. Sections 4.7 and 4.8 explain the experiments and results respectively, and Section 5 explores the broader implications of our results and some future work.

2 Background

2.1 Social Cascades

In [4] the authors use the explicit struture of networks - observing following and friendship relationships in order to explore the effect of actual, instead of inferred, network structure on information

cascades. They observe that the popularity of stories peaks with an age of about one day, and then subsides.

A unique aspect of open networks like reddit, digg, publications, &c is that it is possible for information to latently travel quicly, as opposed toin closed, action oriented networks like the one described in [6], where information, which must individually be spread from one email to another, peters out quickly.

In [7], the authors identify *bursty* keywords that suddently appear, and attempt to align them with trends - entire topics that are becoming more popular. They do this by analyzing new bursts in the queue. One thing we can do in this paper is look at each sub as a queue and see if bursts in one are followed by bursts in another.

2.2 Structure Learning

As internet domains are fully connected, and inference over fully connected graphs is intractable, it is important to

3 Learning Structure

In this section, we present our approach to learning the structure of influence graphs in collaborative sites. We first present an overview of the general problem, followed by our approach to learning structure, and finally validate our approach by learning structure graphs for two real-world datasets, *reddit* and *StackExchange*.

3.1 Challenges for Collaborative Sites

Learning graphical model structure is a particularly difficult task for collaborative sites. Each subsite forms a node in the network, and memes can potentially between any two subsites that share at least one user in common. Typically in social networks, we expect the percentage of mutual friends to be relatively low and edges exist simply as binary friend-or-not connections. However, in collaborative sites with thousands of users, it is reasonable to expect that every subsite shares at least one user in common. A naive approach would thus result in a fully connected graph, which is undesirable as it reveals little insight and may make certain tasks intractable.

3.2 Approximate Structure Learning Algorithm

Since it is possible for a meme to spread between any two subsites, any structure learning algorithm that yields a less-than-fully-connected graph is learning an approximate structure. As noted previously, an approximate structure may be desirable for gaining insights and reducing computational requirements. We next present our approximate structure learning algorithm for collaborative sites.

We first begin by defining an *adjency matrix*, $\mathcal{A}$, over subsites. For each pair, $(s,t) \in \mathcal{A}$, of subsites, we mine the percentage of *active* users that overlap in both s and t. We define a user as *active* if they have posted at least once on both subsites[1]:

$$\mathcal{A}[s,t] = \sum_{\forall u \in \mathcal{U}} \frac{a(u)}{min(|\mathcal{U}^s|, |\mathcal{U}^t|)}$$

$$a(u) = \begin{cases} 1 & u \in P_s \text{ and } u \in P_t \\ 0 & \text{otherwise} \end{cases}$$

Where $\mathcal{U}$ is the set of users and P_i is the set of posts in subsite i.

Rather than focusing on binary co-occurrence edges, our algorithm assesses the strengths of user overlap between two subsites and removes edges that are below a user-specified threshold. We define the relative strength matrix, $\mathcal{A}^*$:

[1]Note that mining inactive users would be infeasible since most collaborative sites have user subsite subscriptions as private.

$$\mathcal{A}^*[s,t] \quad = \quad \frac{\mathcal{A}[s,t] - \mu(\mathcal{A})}{\sigma(\mathcal{A})}$$

Where μ and σ are the mean and standard deviation, respectively. Finally, we define the weighted edge matrix, $\mathcal{E}$:

$$\mathcal{E}[s,t] \quad = \quad \begin{cases} \mathcal{A}^*[s,t] & \text{if } \mathcal{A}^*[s,t] \geq \gamma \\ 0 & \text{otherwise} \end{cases}$$

Where γ is the user-specified strength threshold.

The resulting graph, $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, captures the most likely sources of influence for each node. To predict meme spreads, one could then convert $\mathcal{G}$ to a dynamic graphical model, $\mathcal{G}^*$, that captures the transition from $\mathcal{G}$ to $\mathcal{G}'$:

$$\begin{array}{rcl} \mathcal{G}^* & = & (\mathcal{S}^*, \mathcal{E}^*) \\ \mathcal{S}^* & = & S \cup S' \\ \mathcal{E}^*[s,t] = & = & \begin{cases} 1 & \text{if } s \in \mathcal{S},\, t \in \mathcal{S}', \text{ and } \mathcal{E}[s,t] > 0 \\ 1 & \text{if } s \in \mathcal{S}',\, t \in \mathcal{S}', \text{ and } \mathcal{E}[s,t] > 0 \\ 0 & \text{otherwise} \end{cases} \end{array}$$

Queries to our dynamic graph will always be performed with the nodes in $\mathcal{G}$ being observed, thus connecting them would not affect the MAP inference for nodes in $\mathcal{G}'$. One important point is that while the weights inferred by our algorithm could represent final weights in a graphical model, it is not clear that doing so would produce high accuracy.

3.3 Experiments

To validate our approach, we mined active user adjacency matrices for three real-world collaborative sites: *reddit*, *StackExchange*, and *SomethingAwful*. The results for each site are presented next.

3.3.1 reddit

reddit is a social news site where each subsite[2] represents a different news topic. By default, all users are subscribed to a generic set of subsites such as *politics* and *humor*. The overlap among these default subsites is very high resulting in very little signal in the adjacency matrix; it is also worth noting that these subsites are more likely to be the final destination of memes, given that they are less niche than the non-default subsites. Consequently, we first removed all default subsites from our dataset. To conform to our time constraints and maximize the amount of insight we could draw, we then limited the remaining non-default subsites to only the top 25. The resulting graph generated by our structure learning algorithm is shown in Figure 1.

Two interesting properties of the graph immediately stand out. First, the explicit-content subreddits, *nsfw*, *sex*, and *gonewild*, are clustered together, indicating that memes on these subreddits are not likely to spread to the rest of the network and memes within the cluster will quickly infect the remaining nodes. Conversely, the marijuana-oriented subreddit, *trees*, is a super-node that appears to exert influence on nearly every node and vice-versa. Intuitively, we may thus hypothesize that the users of *trees* are very influential on reddit and would thus expect to see memes spread from and to *trees* rapidly.

3.3.2 StackExchange

StackExchange (SE) is a collaborative question-answer site where each subsite represents a different question topic. A majority of subsites focus on technical topics such as *android* and *security*, though non-technical subsites exist (e.g., *cooking*). Memes in SE are more likely to take the form of rising topics of interest, such as the popularity of iPhone games or Ruby on Rails, rather than cultural

[2] reddit subsites are called subreddits.

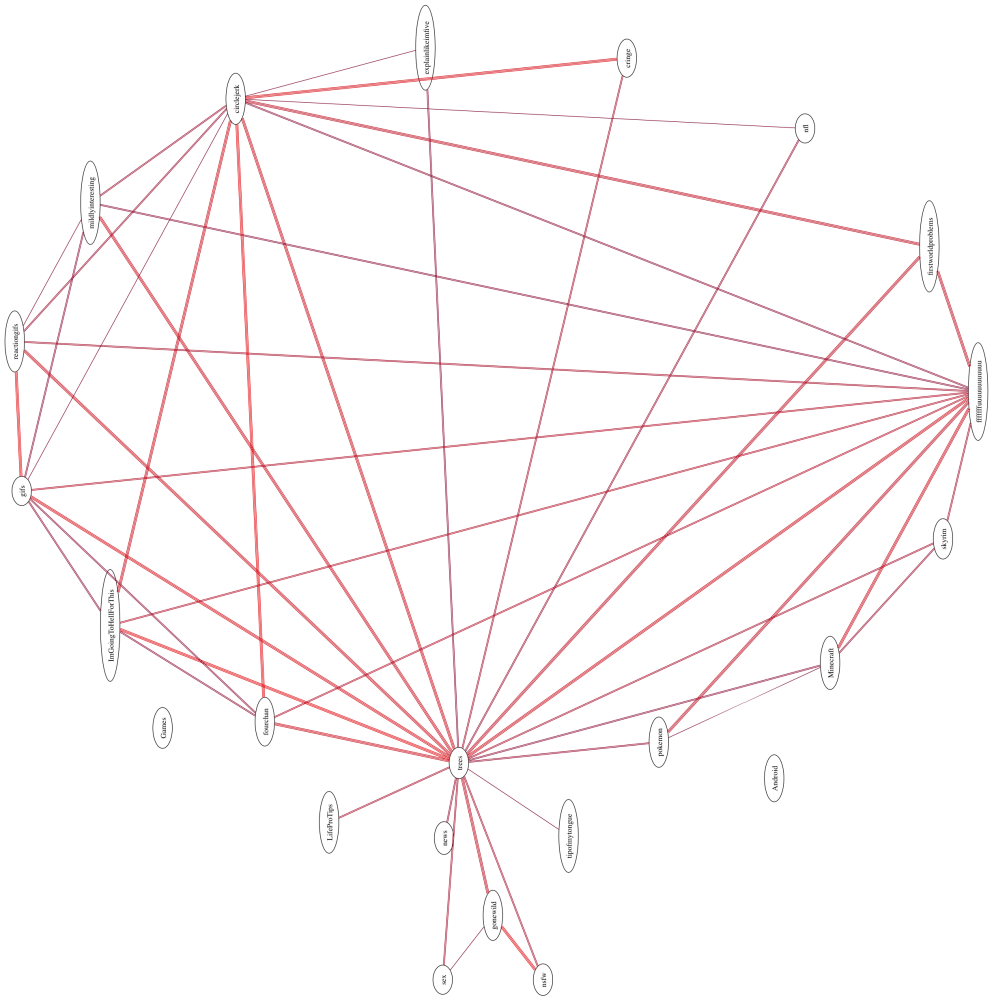


Figure 1: Graph inferred from the top 25 non-default subreddits.

topics like on reddit. Unlike most collaborative sites, SE user subscription data is publicly available. We therefore used the adjacency matrix of all users to build our graph. The resulting graph for SE is shown in Figure 2.

Perhaps counter-intuitively, the *stackoverflow* subsite has two orders of magnitude more overall users than other sites, but exerts no influence on the network. This may be a result of the other sub-sites being relatively unknown but having a strong sub-culture of shared users. Two high-influence pseudo-clusters appear between less technical and strictly technical sites in the upper-left and lower-

right, respectively. Thus, we would expect that localized memes would be frequent among these pseudo-clusters.

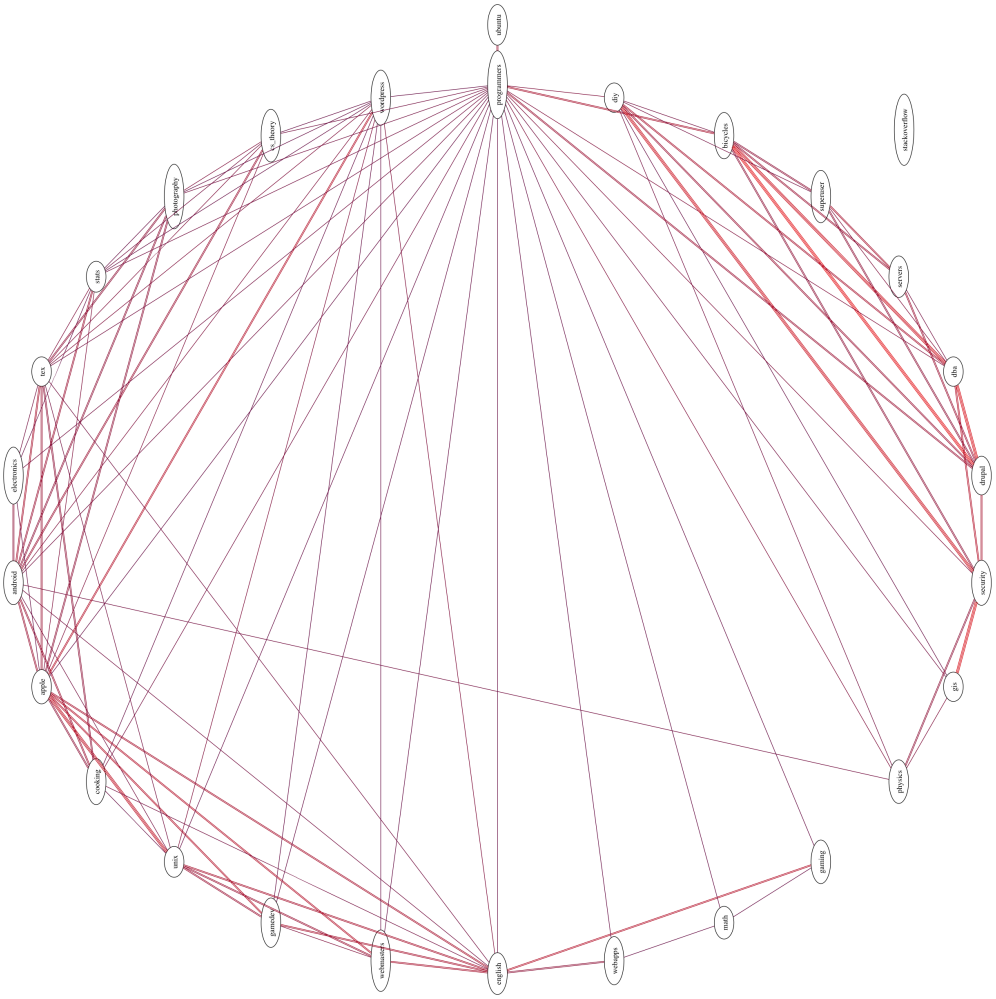


Figure 2: Graph inferred from the 28 StackExchange subsites.

3.3.3 SomethingAwful

4 Predicting Cascades

The second task is a difficult inference problem, detecting cascades in networks of sites. This task has two components - learning the connections between the sites and using the strengths of these connections to predict the sites of future infections. In [2], the authors learn the strength of connections between sites to visualize and explore the data, while we extend this research by using connections to predict future data.

4.1 Problem statement

Given nodes $\mathcal{D}$ with overlapping connections $\mathcal{C}$ via shared users, predict topic vector $v_{d,t+1}$ in document collection d at time $t+1$ given the topics V in all documents at time t?

Following the model in [3], we refer to a node as *contagious* for a given phrase if it has had that phrase trend internally within the last timestep. A node that contains a previously trending phrase can be viewed as having become *infected.*

Given sites $\mathcal{D}$ with overlapping connections $\mathcal{C}$ via shared users, predict topic vector $v_{d,t+1}$ in site d at time $t+1$ given the topics V in all documents at time t.

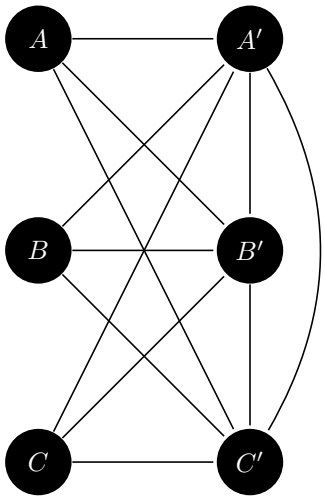
Following the model in [3], we refer to a node as *contagious* for a given phrase if it has had that phrase trend internally within the last timestep. A node that contains a previously trending phrase can be viewed as having become *infected.*

We model this domain as an unfolding Markov Chain, in which at each timestep, nodes become infected or uninfected. See the below graph, in which nodes $\{A, B, C\}$ represent the states of three sites at time t, and the nodes $\{A', B', C'\}$ represent the same nodes at time $t+1$.

4.2 Problem Structure

As infections spread over continuous time but our model involves only discrete timesteps, our model must account for the fact that newly infected nodes may cross-contaminate. Towards this end, we connect all of the nodes in the second timestep.

The number of nodes in both timesteps $|V| = 2|S|$, with S being the set of all sites. The number of edges $|E| = \frac{3}{2}|S|^2 - S$ - S^2 captures influence between each site at time t and each site at time $t+1$, and between each site at time $t+1$ with every other such site.



The number of nodes $|V| = 2|S|$, with S being the set of all sites. The number of edges $|E| = \frac{3}{2}|S|^2$

4.3 Detecting Infections

To detect infections, we use Pointwise Mutual Information [1] to identify salient bigrams in each site in each timestep. Whenever these occur multiple times across the entire dataset, they may be infections. We examine the occurences of each ngram to see if it appears in bursts, and if it does, we designate it an infection.

4.4 Learning Parameters

We used the method for finding the MLE estimate in triangulated graphs described in [8] to learn the weights of our edges. The model is trained on every pairs of steps $t, t+1$. We define indicator functions

$$\mathbb{I}[s] = \begin{cases} 1 & \text{Infection occurs in site } s \\ 0 & s \text{ is not uninfected} \end{cases}$$

and

$$\mathbb{I}[s,t;j,k] = \begin{cases} 1 & \text{if infection is present in site } s \ (j=1) \text{ and } t \ (k=1) \\ 0 & \text{otherwise} \end{cases}$$

Parameters are the logs of the expectations of these indicator functions.

4.5 Inference Problem

Our question Given a cascade at time t, what will be the state of the cascade at time $t+1$ can be restated as the following variational problem:
Given that we have learned our parameters θ_s and $\theta_s t$, and that we have observed half of the nodes $X = \{x_1, x_2, \ldots, x_n\}$ representing all of our sites at time t, what is the MLE assignment to the nodes $X^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$?
This optimization over 2^n possible assignments with n equal to the number of sites under examination is NP-complete.

We define the probability of an assignment x' to our nodes at timestep $t+1$ given our graph G and an assignment x to the nodes at timestep $t = p(x'|x,G) = p(x',x|G)/p(x)$. But as $p(x)$ will be constant for every assignment to x, we use a hillclimbing algorithm letting $\hat{x} = x \cup x'$ and to optimize

$$p_{\hat{\theta}}(\hat{x}) = \exp\left\{ \sum_{s \in V} \hat{\theta}_s(\hat{x}_s) + \sum_{s,t \in E} \hat{\theta}_{st}(\hat{x}_s, \hat{x}_t) \right\}$$

4.6 Datasets

Our algorithm is evaluated on three datasets. These are the Memetracker dataset used in [5], a synthetic dataset generated using the assumptions described above, and a dataset scraped from Something Awful [3]

4.6.1 Memetracker

A collection of popular newssites during the campaign season of 2008. Sites are blogs and major media sources. Memes detected via the Memetracker algorithm [5] are designated infections. These experiments evaluate the 20 most popular webssites and the 40 most popular blogs.

4.6.2 Something Awful

There are 48 public forums on Something Awful, and we scraped the last month of data from all of them. The scraped data is available `cs.utexas.edu/~elie/records`. This data is all from 2012. To identify infections, we use Pointwise Mutual Information [1] to identify salient bigrams. From these, we select the burstiest bigrams to be our infections, and form the infection tree.

[3] `forums.somethingawful.com`

4.7 Experiments

We conducted two sets of experiements, one on each of our datasets. Each of these was similar, in that we used n-fold cross-validation to predict cascades with our algorithm, and with the edge weights learned by NetInf.

Accuracy in these experiments is denoted by the percentage of nodes correctly predicted by our algorithm for timestep $t + 1$ given the nodes in timestep t. We ran a hill-climbing algorithm for five steps to find our assignment to variables. We constructed a model with weighted edges learned by our algorithm and by the NetInf algorithm presented in [2] to assign probabilities and find a MAP assignment to nodes.

4.8 Results

5 Discussion and Future Work

One of the major difficulties of researching infections in social networks is the sparsity questions. If a site is only infected when a meme is present, the graph is very sparse, and phenomena like a user absorbing content at time $t - 1$ and then reproducing it on another site at time $t + 1$ are difficut or impossible to model. However, if a site becomes infected with the first occurance of a meme and stays inected forever, information about when a meme is first flaring or when it hasn't been active in the last several timesteps is lost.

Important future work includes formulating this problem and doing inference over it with real values. In every formulation of the problem thus far, an infection is either present or not present. Real-valued indicators of the presence of an infection will help with the problem of information loss and sparsity.

References

[1] K.W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[2] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):21, 2012.

[3] D. Kempe, J. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. *Automata, Languages and Programming*, pages 99–99, 2005.

[4] Kristina Lerman and Rumi Ghosh. Information contagion: an empirical study of the spread of news on digg and twitter social networks. *CoRR*, abs/1003.2664, 2010.

[5] J. Leskovec, L. Backstrom, and 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*, pages 497–506. ACM, 2009.

[6] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), May 2007.

[7] Michael Mathioudakis and Nick Koudas. Twittermonitor: trend detection over the twitter stream. In Ahmed K. Elmagarmid and Divyakant Agrawal, editors, *SIGMOD Conference*, pages 1155–1158. ACM, 2010.

[8] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.