Infectious Knowledge in a Collaborative News Site

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

3.1 Structure Learning

3.2 Problem Structure

3.3 Inference Problem

3.4 Datasets

3.4.1 Reddit

3.4.2 StackExchange

3.4.3 SomethingAwful

3.5 Experiments

3.6 Results

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.

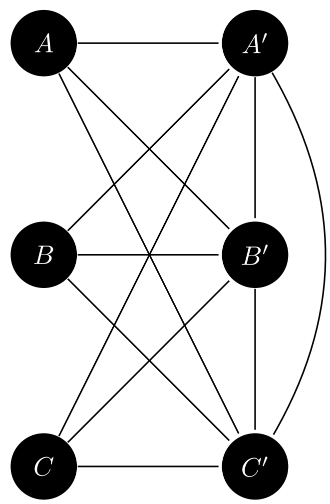
Following the model in [3], we refer to a node as *contagious* for a given phrase if it has had that phrase trend internally within the last timestep. A node that contains a previously trending phrase can be viewed as having become *infected.*

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 $X = \{x_1, x_2, \dots, x_n\}$ representing all of our sites at time t, what is the MLE assignment to the nodes $X^* = \{x_1^*, x_2^*, \dots, x_n^*\}$?
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 [1]

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.

4.6.2 SomethingAwful

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.

4.6.3 Synthetic Dataset

Infections are randomly initialized and spread through sampling via MCMC from generated parameters.

4.7 Experiments

We conducted three sets of experiements, one on each of our three 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 diffcut or

[1] `forums.somethingawful.com`

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.