Contrastive Training for Models of Information Cascades

Shaobin Xu and **David A. Smith**
College of Computer and Information Science
Northeastern University
440 Huntington Avenue
Boston, MA, 02115
{shaobinx,dasmith}@ccs.neu.edu

Abstract

This paper proposes a model of information cascades as directed spanning trees (DSTs) over observed documents. In addition, we propose a contrastive training procedure that exploits partial temporal ordering of node infections in lieu of labeled training links. This combination of model and unsupervised training makes it possible to improve on models that use infection times alone and to exploit arbitrary features of the nodes and of the text content of messages in information cascades. With only basic node and time lag features similar to previous models, the DST model achieves performance with unsupervised training comparable to strong baselines on a blog network inference task. Unsupervised training with additional content features achieves significantly better results, reaching half the accuracy of a fully supervised model.

Introduction

As is their wont, politicians talk—on television, on the floor of the legislature, in printed quotations, and on their websites and social media feeds. If you read and listen to all these statements, you might notice common tropes and turns of phrase that groups of politicians used to describe some issue (Grimmer and Stewart 2013). You might even discover a list of "talking points" underlying this common behavior.[1] Similarly, you might be reading the literature in a scientific field and find that a paper from another discipline starts getting cited repeatedly. Which previous paper, or papers, introduced the new technique? Or perhaps you read several news stories about a new product from some company and then find that they all share text with a press release put out by the company.

In each of these cases, we might be interested in structures at differing levels of detail. We might be interested in individual **links**, e.g., knowing which previous paper it was that later papers were mining for further citations; in **cascades**, e.g., knowing which news stories are copying from which press releases or from each other; and in **networks**, e.g., knowing which politicians are most likely to share talking points or which newspapers are most likely to publish press releases from particular businesses or universities. Depending on our data source, some of these structures could be directly observed. With the right API calls, we might observe retweets (links), chains of retweets (cascades), and follower relations (networks) on Twitter. We might also be interested in inferring an underlying social network for which the Twitter follower relation is partial evidence. In contrast, politicians interviewed on television do not explicitly cite the sources of their talking points, which must be inferred.

Observing the diffusion process often reduces to keeping track of when nodes (newspapers, bills, people, etc.) mention a piece of information, reuse a text, get infected, or exhibit a *contagion* in a general sense. When the structure of the propagation of contagion is hidden and we cannot tell which node infected which, all we have is the result of diffusion process—that is, the timestamp and possibly other information when the nodes get infected. We want to infer the diffusion process itself by using such information to predict the links of underlying network. There have been increasing efforts to uncover and model different types of information cascades on networks (Brugere, Gallagher, and Berger-Wolf 2016): modeling hidden networks from observed infections (Stack et al. 2012; Rodriguez et al. 2014), modeling topic diffusion in networks (Gui et al. 2014), predicting social influence on individual mobility (Mastrandrea, Fournet, and Barrat 2015) and so on.

This work all focuses on using parametric models of the time differences between infections. Such models are useful when the only information we can get from the result of diffusion process is the timestamps of infections. We can hope to make better predictions, however, with access to additional features, such as the location of each node, the similarity between the messages received by two nodes, etc. Popular parametric models cannot incorporate these features into unsupervised training.

In this paper, we propose an edge-factored, conditional log-linear directed spanning tree (DST) model with an unsupervised, contrastive training procedure to infer the link structure of information cascades. After reviewing related work, we describe the DST model in detail, including an efficient inference algorithm using the directed matrix-tree theorem and the gradient of our maximum conditional likelihood optimization problem. We then report experiments on the ICWSM Spinn3r dataset, where we can observe the true

[1] The U.S. State Department, for example, produced a much-discussed set of talking points memos in response to the 2012 attack in Benghazi.

hyperlink structure for evaluation and compare the proposed method to MultiTree (Rodriguez and Schölkopf 2012) and InfoPath (Rodriguez et al. 2014) and some simple, but effective, baselines. We conclude by discussing directions for future work.

Related Work

There has been a great deal of work on trying to infer underlying network structure using information cascades, most of which are based on the independent cascade (IC) model (Saito, Nakano, and Kimura 2008). We evaluate the DST model against a transmission based model from Rodriguez and Schölkopf (2012), which, similar to our work, also uses directed spanning trees to represent cascades, but employs a submodular parameter-optimization method and fixed activation rates. In addition, we compare our work with an advanced model from Rodriguez et al. (2014), which uses a generative probabilistic model for inferring both static and dynamic diffusion networks. It is a line of work starting from using a generative model with fixed activation rate (Gomez Rodriguez, Leskovec, and Krause 2010; Rodriguez and Schölkopf 2012; Myers and Leskovec 2010). Later comes the development of inferring the activation rate between nodes to reveal the network structure (Rodriguez, Balduzzi, and Schölkopf 2011; Gomez Rodriguez, Leskovec, and Schölkopf 2013; Snowsill et al. 2011; Rodriguez, Leskovec, and Schölkopf 2013; Rodriguez et al. 2014). Zhai, Wu, and Xu (2015) use a Markov chain Monte Carlo approach for the inference problem. Linderman and Adams (2014) propose a probabilistic model based on mutually-interacting point processes and also use MCMC for the inference. Gui et al. (2014) model topic diffusion in multi-relational networks. An interesting approach by Amin, Heidari, and Kearns (2014) infers the unknown network structure, assuming the detailed timestamps for the spread of the contagion are not observed but that "seeds" for cascades can be identified or even induced experimentally. Wang, Ermon, and Hopcroft (2012) propose feature-enhanced probabilistic models for diffusion network inference while still maintaining the requirement that exact propagation times be observed and modeled. Daneshmand et al. (2014) and Abrahao et al. (2013) perform theoretical analysis of transmission-based cascade inference models. While the foregoing approaches are all based on parametric models of propagation time between infections, Rong, Zhu, and Cheng (2016) experiment with a nonparametric approach to discriminating the distribution of diffusion times between connected and unconnected nodes. Recently, Brugere, Gallagher, and Berger-Wolf (2016) have compiled a survey about the methods and applications for different network structure inference problems.

Tutte's directed matrix-tree theorem, which plays a key role in our approach, has been used in natural language processing to infer posterior probabilities for edges in nonprojective syntactic dependency trees (Smith and Smith 2007; Koo et al. 2007; McDonald and Satta 2007) and for inferring semantic hierarchies (i.e., ontologies) over words (Bansal et al. 2014).

Method

In this section, we present our modeling and inference approaches. We first present a simple log-linear, edge-factored directed spanning tree (DST) model of cascades over network nodes. This allows us to talk concretely about the likelihood objective for supervised and unsupervised training, where we present a **contrastive** objective function. We note that other models besides the DST model could be trained with this contrastive objective. Finally, we derive the gradient of this objective and its efficient computation using Tutte's directed matrix-tree theorem.

Log-linear Directed Spanning Tree Model

For each cascade, define a set of activated nodes $\mathbf{x} = \{x_1, \ldots, x_n\}$, each of which might be associated with a timestamp and other information that are the input to the model. Nodes thus correspond to (potentially) dateable entities such as webpages or posts, and not aggregates, such as websites or users. Let $\mathbf{y}$ be a directed spanning tree of $\mathbf{x}$, which is a map $\mathbf{y} : \{1, \ldots, n\} \rightarrow \{0, 1, \ldots, n\}$ from child indices to parent indices of the cascade. In the range of mapping $\mathbf{y}$ we add a new index 0, which represents a dummy "root" node x_0. This allows us to model both single cascades and to disentangle multiple cascades on a set of nodes $\mathbf{x}$, since more than one "seed" node might attach to the dummy root. In the experiments below, we model datasets with both single-rooted ("separated") and multi-rooted ("merged") cascades.

A valid directed spanning tree is by definition acyclic. Every node has exactly one parent, with the edge $x_{\mathbf{y}(i)} \rightarrow x_i$, except that the root node has in-degree 0. We might wish to impose additional constraints C on the set of spanning trees: for instance, we might require that edges not connect nodes with timestamps known to be in reverse order. Let $\mathcal{Y}_C$ be the set of all valid directed spanning trees that satisfy a list of rules in constraint set C over $\mathbf{x}$, and $\mathcal{Y}$ be the set of all directed spanning trees over the same sequence of $\mathbf{x}$ but without any constraint being imposed.

Define a log-linear model for trees over $\mathbf{x}$. The unnormalized probability of the tree $\mathbf{y} \in \mathcal{Y}$ is thus:

$$\tilde{p}_{\vec{\theta}}(\mathbf{y} \mid \mathbf{x}) = e^{\vec{\theta} \cdot \vec{f}(\mathbf{x}, \mathbf{y})} \tag{1}$$

where $\vec{f}$ is a feature vector function on cascade and $\vec{\theta} \in \mathbb{R}^m$ parameterizes the model. Following (McDonald, Crammer, and Pereira 2005), we assume that features are **edge-factored**:

$$\vec{\theta} \cdot \vec{f}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \vec{\theta} \cdot \vec{f}_{\mathbf{x}}(\mathbf{y}(i), i) = \sum_{i=1}^{n} s(\mathbf{y}(i), i) \tag{2}$$

where $s(i, j)$ is the *score* of a directed edge $i \rightarrow j$. In other words, given the sequence $\mathbf{x}$ and the cascade is a directed spanning tree, this directed spanning tree (DST) model assumes that the edges in the tree are all conditionally independent of each other.

Despite the constraints they impose on features, we can perform inference with edge-factored models using tractable

$O(n^3)$ algorithms, which is one of the advantages this model brings. Since $\tilde{p}_{\vec{\theta}}(\mathbf{y} \mid \mathbf{x})$ is not a normalized probability, we divide it by the sum over all possible directed spanning trees, which gives us:

$$p_{\vec{\theta}}(\mathbf{y} \mid \mathbf{x}) = \frac{e^{\vec{\theta}\cdot\vec{f}(\mathbf{x},\mathbf{y})}}{\sum_{\mathbf{y}'\in\mathcal{Y}} e^{\vec{\theta}\cdot\vec{f}(\mathbf{x},\mathbf{y}')}} = \frac{e^{\vec{\theta}\cdot\vec{f}(\mathbf{x},\mathbf{y})}}{Z_{\vec{\theta}}(\mathbf{x})} \tag{3}$$

where $Z_{\vec{\theta}}(\mathbf{x})$ denotes the sum of log-linear scores of all directed spanning trees, i.e., the partition function.

If, for a given set of parameters $\vec{\theta}$, we merely wish to find the $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} p_{\vec{\theta}}(\mathbf{y} \mid \mathbf{x})$, we can pass the scores for each $i \rightarrow j$ candidate edge to the Chu-Liu-Edmonds maximum directed spanning tree algorithm (Chu and Liu 1965; Edmonds 1967).

Likelihood of a cascade

When we observe all the directed links in a training set of cascades, we now have the machinery to perform supervised training with maximum conditional likelihood. We can simply maximize the likelihood of the true directed spanning tree $p_{\vec{\theta}}(\mathbf{y}^* \mid \mathbf{x})$ for each cascade in our training set, using the gradient computations discussed below.

When we do not observe the true links in a cascade, we need a different objective function. While we cannot restrict the numerator in the likelihood function (3) to a single, true tree, we can restrict it to the set of trees $\mathcal{Y}_C$ that obey some constraints C on valid cascades. As mentioned above, these constraints might, for instance, require that links point forward in time or avoid long gaps. We can now write the likelihood function for each cascade $\mathbf{x}$ as a sum of the probabilities of all directed spanning trees that meet the constraints C:

$$\mathcal{L}_{\mathbf{x}} = \sum_{\mathbf{y}\in\mathcal{Y}_{\mathbf{C}}} p_{\vec{\theta}}(\mathbf{y} \mid \mathbf{x}) = \frac{\sum_{\mathbf{y}\in\mathcal{Y}_{\mathbf{C}}} e^{\vec{\theta}\cdot\vec{f}(\mathbf{x},\mathbf{y})}}{Z_{\vec{\theta}}(\mathbf{x})} = \frac{Z_{\vec{\theta},C}(\mathbf{x})}{Z_{\vec{\theta}}(\mathbf{x})} \tag{4}$$

where $Z_{\vec{\theta},C}(\mathbf{x})$ denotes the sum of log-linear scores of all valid directed spanning trees under constraint set C.

This is a **contrastive** objective function that, instead of maximizing the likelihood of a single outcome, maximizes the likelihood of a **neighborhood** of possible outcomes contrasted with implicit negative evidence (Smith and Eisner 2005). A similar objective could be used to train other cascade models besides the log-linear the DST model presented above, e.g., models such as the Hawkes process in Linderman and Adams (2014).

As noted above, cascades on a given set of nodes are assumed to be independent. We thus have a log-likelihood over all N cascades:

$$\log \mathcal{L}_{\mathbf{N}} = \sum_{\mathbf{x}} \log \mathcal{L}_{\mathbf{x}} = \sum_{\mathbf{x}} \log \frac{Z_{\vec{\theta},C}(\mathbf{x})}{Z_{\vec{\theta}}(\mathbf{x})} \tag{5}$$

Maximizing Likelihood

Our goal is to find the parameters $\vec{\theta}$ that solve the following maximization problem: problem:

$$\max_{\vec{\theta}} \log \mathcal{L}_N = \max_{\vec{\theta}} \sum_{\mathbf{x}} (\log Z_{\vec{\theta},C}(\mathbf{x}) - \log Z_{\vec{\theta}}(\mathbf{x})) \tag{6}$$

To solve this problem with quasi-Newton numerical optimization methods such as L-BFGS (Liu and Nocedal 1989), we need to compute the gradient of the objective function, which for a given parameter θ_k is given by the following equation:

$$\frac{\partial \log \mathcal{L}_N}{\partial \theta_k} = \sum_{\mathbf{x}} \left(\frac{\partial \log Z_{\vec{\theta},C}(\mathbf{x})}{\partial \theta_k} - \frac{\partial \log Z_{\vec{\theta}}(\mathbf{x})}{\partial \theta_k} \right) \tag{7}$$

For a cascade that contains n nodes, even if we have tractable number of valid directed spanning trees in $\mathcal{Y}_C$, there will be n^{n-2} (Cayley's formula) possible directed spanning trees for the normalization factor $Z_{\vec{\theta}}(\mathbf{x})$, which makes the computation intractable. Fortunately, there exists an efficient algorithm that can compute $Z_{\vec{\theta}}(\mathbf{x})$, or $Z_{\vec{\theta},C}(\mathbf{x})$, in $O(n^3)$ time.

Matrix-Tree Theorem and Laplacian Matrix

Tutte (1984) proves that for a set of nodes $x_0, \ldots, x_n$, the sum of scores of all directed spanning trees $Z_{\vec{\theta}}(\mathbf{x})$ in $\mathcal{Y}$ rooted at x_j is

$$Z_{\vec{\theta}}(\mathbf{x}) = \left| \hat{\mathbf{L}}^j_{\vec{\theta},\mathbf{x}} \right| \tag{8}$$

where $\hat{\mathbf{L}}^j_{\vec{\theta},\mathbf{x}}$ is the matrix produced by deleting the j-th row and column from Laplacian matrix $\mathbf{L}_{\vec{\theta},\mathbf{x}}$.

Before we define Laplacian matrix, we first denote:

$$u_{\vec{\theta},\mathbf{x}}(j,i) = e^{\vec{\theta}\cdot\vec{f}_{\mathbf{x}}(j,i)} = e^{s_{\vec{\theta},\mathbf{x}}(j,i)} \tag{9}$$

where $j = \mathbf{y}(i)$, which is the parent of x_i in $\mathbf{y}$. Recall that we define the unnormalized score of a spanning tree over $\mathbf{x}$ as a log-linear model using edge-factored scores (Eq 1, 2). Therefore, we have:

$$e^{\vec{\theta}\cdot\vec{f}(\mathbf{x},\mathbf{y})} = e^{\sum_i \vec{\theta}\cdot\vec{f}_{\mathbf{x}}(\mathbf{y}(i),i)} = \prod_{i=1}^{n} u_{\vec{\theta},\mathbf{x}}(j,i) \tag{10}$$

where $u_{\vec{\theta},\mathbf{x}}(j,i)$ represents the multiplicative contribution of the edge from parent j to child i to the total score of the tree.

Now we can define the Laplacian matrix $\mathbf{L}_{\vec{\theta},\mathbf{x}} \in \mathbb{R}^{(n+1)\times(n+1)}$ for directed spanning trees by:

$$[\mathbf{L}_{\vec{\theta},\mathbf{x}}]_{j,i} = \begin{cases} -u_{\vec{\theta},\mathbf{x}}(j,i) & \text{if edge } (j,i) \in C \\ \sum_{k\in\{0,\ldots,n\},k\neq j} u_{\vec{\theta},\mathbf{x}}(k,i) & \text{if } j = i \\ 0 & \text{if edge } (j,i) \notin C \end{cases} \tag{11}$$

where j represents a parent node and i represents a child node. As for all possible valid directed spanning trees, we will have 0 for all entries where the edge from parent j to

Table 1: Cascade-level inference of DST with different feature sets, in unsupervised learning setting (Table 1a) in comparison with naive attach-everything-to-earliest baseline, as well as supervised learning setting (Table 1b)

Dataset	Cascade types	Method	Recall	Precision	F1
ICWSM 2011	Separated Cascades	DST Basic	0.348	0.454	0.394
		DST Enhanced	**0.504**	**0.658**	**0.571**
		Naive Baseline	0.450	0.587	0.509
	Merged Cascades	DST Basic	0.027	0.035	0.031
		DST Enhanced	**0.036**	**0.047**	**0.040**
		Naive Baseline	0.015	0.019	0.017
ICWSM 2011 (tree structure enforced)	Separated Cascades	DST Basic	0.622	0.622	0.622
		DST Enhanced	**0.946**	**0.946**	**0.946**
		Naive Baseline	0.941	0.933	0.937
	Merged Cascades	DST Basic	0.042	0.042	0.042
		DST Enhanced	**0.246**	**0.246**	**0.246**
		Naive Baseline	0.043	0.043	0.043

(a) Unsupervised Setting

Merged Cascades	Training		Test	
	Recall	Precision	Recall	Precision
Basic Feature Set	0.171±0.001	0.171±0.001	0.164±0.007	0.164±0.007
Enhanced Feature Set	0.475±0.002	0.475±0.002	0.455±0.011	0.455±0.011
Naive Baseline	0.042±0.001	0.042±0.001	0.046±0.009	0.046±0.009

(b) Supervised Setting

child i doesn't satisfy the specified constraint set. For all possible directed spanning trees, however, the constraint set C is $V \times V$, that is all possible edges.

We can use the LU factorization to compute the matrix inverse, so that the determinant of the Laplacian matrix can be done in $O(n^3)$ times. Meanwhile, the Laplacian matrix is diagonally dominant, in that we use positive edge scores to create the matrix. The matrix therefore is guaranteed to be invertible.

Gradient

Smith and Smith (2007) use a similar inference approach for probabilistic models of nonprojective dependency trees. They derive that for any parameter θ_k,

$$\frac{\partial \log Z_{\vec{\theta}}(\mathbf{x})}{\partial \theta_k} = \frac{1}{\left|\mathbf{L}_{\vec{\theta},\mathbf{x}}\right|} \sum_{i=1}^{n} \sum_{j=0}^{n} u_{\vec{\theta},\mathbf{x}}(j,i) f_{\mathbf{x}}^{k}(j,i) \times \left(\frac{\partial \left|\mathbf{L}_{\vec{\theta},\mathbf{x}}\right|}{\partial [\mathbf{L}_{\vec{\theta},\mathbf{x}}]_{i,i}} - \frac{\partial \left|\mathbf{L}_{\vec{\theta},\mathbf{x}}\right|}{\partial [\mathbf{L}_{\vec{\theta},\mathbf{x}}]_{j,i}} \right) \tag{12}$$

Also, for an arbitrary matrix $\mathbf{A}$, they derive the gradient of $\mathbf{A}$ with respect to any cell $[\mathbf{A}]_{j,i}$ using the determinant and entries in the inverse matrix:

$$\frac{\partial |\mathbf{A}|}{\partial [\mathbf{A}]_{j,i}} = |\mathbf{A}| \, [\mathbf{A}^{-1}]_{i,j} \tag{13}$$

Plugging (13) to (12) gives us the final gradient of $Z_{\vec{\theta}}(\mathbf{x})$ with respect to θ_k:

$$\frac{\partial \log Z_{\vec{\theta}}(\mathbf{x})}{\partial \theta_k} = \sum_{i=1}^{n} \sum_{j=0}^{n} u_{\vec{\theta},\mathbf{x}}(j,i) f_{\mathbf{x}}^{k}(j,i) \times \left([\mathbf{L}_{\vec{\theta},\mathbf{x}}^{-1}]_{i,i} - [\mathbf{L}_{\vec{\theta},\mathbf{x}}^{-1}]_{i,j} \right) \tag{14}$$

Experiments

One of the hardest tasks in network inference problems is gathering information about the true network structure. Most existing work has conducted experiments on both synthetic data with different parameter settings and on real-world networks that match the assumptions of proposed method. Generating synthetic data, however, is less feasible if we want to exploit complex textual features, which negates one of the advantages of the DST model. Generating child text from parent documents is beyond the scope of this paper, although we believe it to be a promising direction for future work. In this paper, therefore, we train and test on documents from the ICWSM 2011 Spinn3r dataset (Burton, Kasch, and Soboroff 2011). This allows us to compare our method with MultiTree (Rodriguez and Schölkopf 2012) and InfoPath (Rodriguez et al. 2014), both of which output a network given a set of cascades. We also analyze the performance of DST at the cascade level, an ability that MultiTree, InfoPath and similar methods lack.

Table 2: Comparison of MultiTree, InfoPath and DST on inferring a static network on the original ICWSM 2011 dataset and on a dataset with enforced tree structure. The DST model is trained and tested unsupervisedly on both separate cascades and merged cascades using different feature sets and the naive attach-everything-to-earliest-node baseline.

Dataset	Cascade types	Method	Recall	Precision	F1	AP
ICWSM 2011	Separated Cascades	MultiTree	0.367	0.242	0.292	N/A
		InfoPath	0.414	0.273	0.329	N/A
		DST Basic	0.557	0.368	0.443	0.279
		DST Enhanced	**0.842**	0.556	**0.670**	**0.599**
		Naive Baseline	0.622	**0.595**	0.608	0.385
	Merged Cascades	DST Basic	0.052	0.034	0.041	**0.003**
		DST Enhanced	**0.057**	**0.038**	**0.045**	**0.003**
		Naive Baseline	0.015	0.019	0.017	0.001
ICWSM 2011 (tree structure enforced)	Separated Cascades	MultiTree	0.249	0.196	0.220	N/A
		InfoPath	0.375	0.294	0.330	N/A
		DST Basic	0.618	0.486	0.544	0.452
		DST Enhanced	**0.950**	0.747	**0.836**	**0.915**
		Naive Baseline	0.941	**0.933**	0.937	0.892
	Merged Cascades	DST Basic	0.083	0.065	0.073	0.012
		DST Enhanced	**0.207**	**0.163**	**0.182**	**0.047**
		Naive Baseline	0.043	0.043	0.043	0.005

Dataset Description

The ICWSM 2011 Spinn3r dataset consists of 386 million different web posts, such as blog posts, news articles, social media content, etc., made between January 13 and February 14, 2011. We first avoid including hyperlinks that connect two posts from the same websites, as they could simply be intra-website navigation links. In addition, we enforce a strict chronological order from source post to destination post to filter out erroneous date fields. Then, by backtracing hyperlinks to the earliest ancestors and computing connected components, we are able to obtain about 75 million clusters, each of which serves as a separate cascade. We only keep the cascades containing between 5 and 100 posts, inclusive. This yields 22,904 cascades containing 205,234 posts from 61,364 distinct websites. We create ground truth for each cascade by using the hyperlinks as a proxy for real information flow. For time-varying network, we include edges only appear in a particular day into the corresponding network, while for the static network we simply include any existing edge regardless of the timestamp.

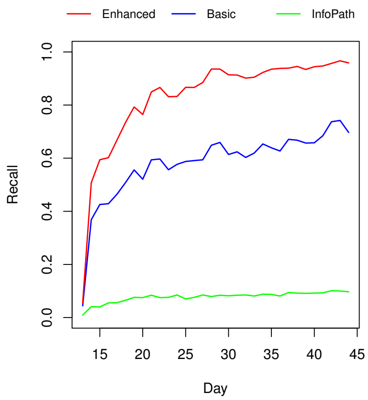
Of these cascades, approximately 61% don't have a tree structure, and among the remaining cascades, 84% have flat structures, meaning for each cascade, all nodes other than the earliest one are in fact attached to the earliest node in the ground truth. In this paper we report the performance of our models on the original dataset and on a dataset where the cascades are guaranteed to be trees. To construct the tree cascades, before finding the connected components using hyperlinks, we remove posts which have hyperlinks coming from more than one source website. Selecting, as above, cascades whose sizes are between 5 and 100 nodes, this yields 20,424 separate cascades containing 201,875 posts from 63,576 distinct websites

We also merge cascades to combine all cascades that start within an hour of each other to make the inference problem more challenging and realistic, since if we don't know links, we are unlikely to know the membership of nodes in cascades exactly. In the original ICWSM 2011 dataset, we obtain 789 merged cascades and for the tree-constrained data, we obtain 938 merged cascades. When merging cascades, we only change the links to the dummy root node, and the underlying network structure remains the same. The DST model would be able to learn different parameters depending on whether we train it on separate cascades or merged cascades. We report on the comparison between both with MultiTree and InfoPath.

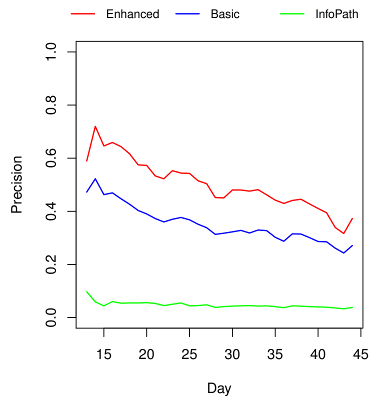
Feature Sets

Most existing work on network structure inference described in Related Work only uses the time difference between two nodes as the feature for learning and inference. Our model has the ability to incorporate different features as pointed out in Eq. 1 and 2. Hence in this paper we experiment different features and report on the following sets:

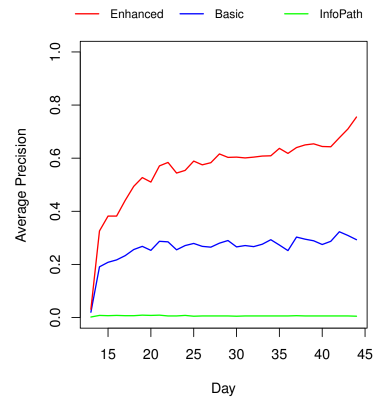
- *basic feature sets*, which include only the node information and timestamp difference, which resembles what the other models do; and
- *enhanced feature sets*, which include the basic feature sets, as well as the languages that both nodes of an edge use, what content types as assigned by Spinn3r (blog, news, etc.), whether a node is the earliest node in the cluster, and the Jaccard distance between the normalized texts in the two nodes.



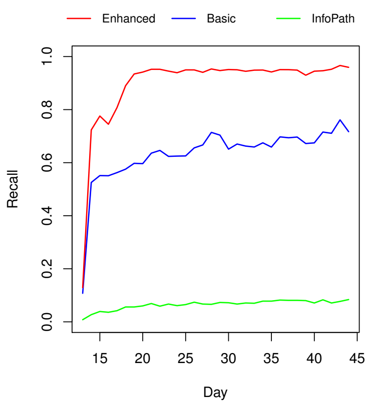
(a) Recall on graph data



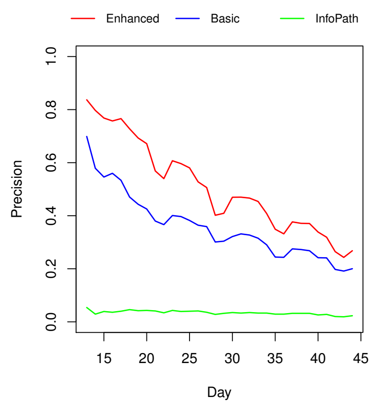
(b) Precision on graph data



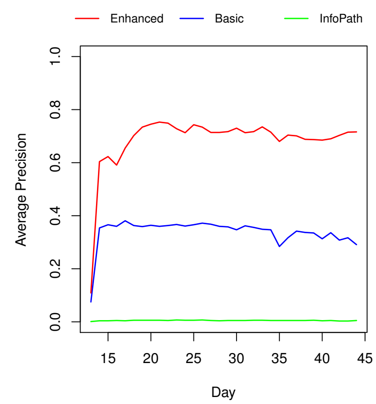
(c) AP on graph data



(d) Recall on tree data



(e) Precision on tree data



(f) AP on tree data

Figure 1: Recall, precision, and average precision of InfoPath and DST on predicting the time-varying networks generated per day. The DST model is trained unsupervisedly on separate cascades using basic and enhanced features. The upper row uses graph-structured cascades from the ICWSM 2011 dataset. The lower row uses the subset of cascades with tree structures.

We use one-hot encoding to represent the feature vectors. In addition, we discretize real-valued features by binning them.

Result of Unsupervised Learning at Cascade Level

In practice, we use Apache Spark for parallelizing the computation to speed up the optimization process. We choose batch gradient descent with a fixed learning rate 5×10^{-3} and report the result after $1,500$ iterations. Inspecting the results of the last two iterations confirms that all training runs converge. The constraint set C contains edges that satisfy: (1) time constraints, and (2) only nodes within the first hour of a specific cascade can be attached to root.

The DST model outputs finer-grained structure than existing approaches and predicts a tree for each cascade, with edges equal to the number of nodes. We report the micro-averaged recall, precision, and F1 for the whole dataset.

The top half of Table 1a shows the results of training the DST model in an unsupervised setting with different feature sets on both separate cascades dataset and merged cascades dataset. We also include a naive baseline that simply attaches all other nodes to the earliest node in a cascade.

From Table 1a we can see the flatness problem. The naive baseline can already achieves 45% recall and 58.7% precision, while knowing the websites and time lags only yields 34.8% recall and 45.4% precision, which partly attributes to the time constraints we apply on creating the Laplacian matrix so that the model can at least gets the earliest node and one of the edges leaving from that node right. On the other hand, the enhanced feature set utilizes the features from the textual content of posts such as the Jaccard distance. Having this information helps the DST model outperform the naive baseline. In the merged clusters setting, instead of only one seed per cascade being attached to the implicit root, we have multiple seeds occurring within the same hour attached to the root. Hence, the naive baseline strategy can at most get the original cascade to which the earliest node belongs right. DST with both feature sets can achieve a better result. We believe in the future, adding more content based features will further boost the performance. We expect, however, that disentangling multiple information flow paths will remain a challenging problem in many domains.

Result of Unsupervised Learning at Network Level

In this section, we evaluate effectiveness on inferring the network structure, comparing to MultiTree and InfoPath. The DST model outputs a tree for each cascade with posterior probabilities for each edge. To convert to a network, we sum all posteriors for a certain edge to get a combined score, from which we obtain a ranked list of edges between websites. We report on two different sets of quantitative measurements: recall/precision/F1 and average precision.

When using InfoPath, we assume an exponential edge transmission model, and sample 20,000 cascades for the stochastic gradient descent. The output has the activation rates for a certain edge in the network per day. We keep those edges which have non-zero activation rate and actually present on that day to counteract the decaying assumption in InfoPath. We then compute the recall/precision/average-precision for each day. To compare the DST model with InfoPath on the time-varying network, we pick edges from the ranked list of the DST model on each day, the number of which matches InfoPath's choice. We exclude MultiTree for the lack of ability to model a dynamic network.

Figure 1 shows the comparison between InfoPath and the DST model with different feature sets. We can see that the DST model outperforms InfoPath by a large margin on every metric with the enhanced feature set being the best.

Now we can compare the DST model with MultiTree and InfoPath on the static network. We include every edge in the output of InfoPath. The top part of Table 2 shows a comparison between the two models in a similar way to the comparisons mentioned before, where the number of edges from the DST model equals to the number of total edges selected by InfoPath. As for MultiTree, we keep all the parameters default while setting the number of edges to match InfoPath's as well. Since MultiTree assumes a fixed activation rate, while InfoPath gives activation rate based on the timestep, there is no way to rank the edges in the static network both methods inferred; therefore, we don't report average-precision for them.

The DST model also outperforms MultiTree and InfoPath in inferring static network structure. Notably, the recall/precision of InfoPath is much higher than the recall/precision per day (Figure 1). This is due to the fact that edges InfoPath correctly selects in the static network might not be correct on that specific day.

Enforcing Tree Structure on the Data

In the ICWSM 2011 dataset, 61% of the cascades are DAGs. Since DST, MultiTree, and InfoPath all assume that they are trees, we evaluate performance on data where this constraint is satisfied—i.e., the tree-constrained dataset described above. The bottom part of Table 1a shows that the naive baseline for separate cascades achieves 94.1% recall/precision because of flatness. DST with enhanced features beats it by a mere 0.5%. This leaves very little room for DST to improve in cascade structure inference problem for separate cascades. For merged cascades, the naive baseline can at most get the original cascade to which the earliest node belongs right. DST with basic feature set did adequately on finding the earliest nodes but found very few correct edges inside the cascades, while enhanced feature set is better at reconstructing the cascade structures thanks to the knowledge of textual features, which leads about a 600% margin. With only 24.6% recall/precision, there is still room for improvement on this very hard inference problem. On network inference, DST with the enhanced feature set also performs the best for recall and average precision but lags on precision. Table 2, Figure 1d, 1e and 1f show similar performance when comparing with MultiTree and InfoPath on inferring different types of network structure.

Result of Supervised Learning at Cascade Level

Our proposed model has the ability to perform both supervised and unsupervised learning, with different objective functions. One of the main contributions of the DST model is to be able to learn the cascade level structure in a feature-enhanced and unsupervised way. However, supervised learning can establish on upper bound for unsupervised performance when trained with the same features.

Table 1b shows the result of supervise learning using DST on the merged cascades with tree structure enforced. Since there are only 938 merged cascades, we perform a 10-fold cross validation on both dataset and we report the result of 5 folds. We split the training and test set by interleaved round-robin sampling from the merged cascades dataset. Although not precisely comparable to DST in the unsupervised setting due to this jackknifing, Table 1b still shows results about twice as large as for unsupervised training.

Conclusion

We have proposed a method to uncover the network structure of information cascades using an edge-factored, conditional log-linear model, which can incorporate more features than most comparable models. This directed spanning tree (DST) model can also infer finer grained structure on the cascade level, besides inferring global network structure. We utilize the matrix-tree theorem to prove that the likelihood function of the conditional model can be solved in cubic time and to derive a contrastive, unsupervised training procedure. We show that for ICWSM 2011 Spinn3r dataset, our proposed method outperforms the baseline MultiTree and InfoPath methods in terms of recall, precision, and average precision. In the future, we expect that applications of this technique could benefit from richer textual features—including full generative models of child document text—and different model structures trained with the contrastive approach.

Acknowledgements

This research was supported by the National Institute on Deafness and Other Communication Disorders of the National Institutes of Health under award number R01DC009834 and by the Andrew W. Mellon Foundation. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH or the Mellon.

References

Abrahao, B.; Chierichetti, F.; Kleinberg, R.; and Panconesi, A. 2013. Trace complexity of network inference. In *KDD*, 491–499.

Amin, K.; Heidari, H.; and Kearns, M. 2014. Learning from contagion (without timestamps). In *ICML*, 1845–1853.

Bansal, M.; Burkett, D.; de Melo, G.; and Klein, D. 2014. Structured learning for taxonomy induction with belief propagation. In *ACL*, 1041–1051.

Brugere, I.; Gallagher, B.; and Berger-Wolf, T. Y. 2016. Network Structure Inference, A Survey: Motivations, Methods, and Applications. *ArXiv e-prints*.

Burton, K.; Kasch, N.; and Soboroff, I. 2011. The ICWSM 2011 Spinn3r dataset. In *ICWSM*.

Chu, Y., and Liu, T. 1965. On the shortest arborescence of a directed graph. *Science Sinica* 14:1396–1400.

Daneshmand, H.; Gomez-Rodriguez, M.; Song, L.; and Schoelkopf, B. 2014. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *ICML*, 793–801.

Edmonds, J. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards* 71B:233–240.

Gomez Rodriguez, M.; Leskovec, J.; and Krause, A. 2010. Inferring networks of diffusion and influence. In *KDD*, 1019–1028.

Gomez Rodriguez, M.; Leskovec, J.; and Schölkopf, B. 2013. Structure and dynamics of information pathways in online media. In *WSDM*, 23–32.

Grimmer, J., and Stewart, B. M. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis* 21(3):267–297.

Gui, H.; Sun, Y.; Han, J.; and Brova, G. 2014. Modeling topic diffusion in multi-relational bibliographic information networks. In *CIKM*, 649–658.

Koo, T.; Globerson, A.; Carreras Pérez, X.; and Collins, M. 2007. Structured prediction models via the matrix-tree theorem. In *EMNLP-CoNLL*, 141–150.

Linderman, S. W., and Adams, R. P. 2014. Discovering latent network structure in point process data. In *ICML*, 1413–1421.

Liu, D. C., and Nocedal, J. 1989. On the limited memory BFGS method for large scale optimization. In *Mathematical Programming*, volume 45. Springer. 503–528.

Mastrandrea, R.; Fournet, J.; and Barrat, A. 2015. Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLoS ONE* 10(9):e0136497.

McDonald, R., and Satta, G. 2007. On the complexity of non-projective data-driven dependency parsing. In *IWPT*, 121–132.

McDonald, R.; Crammer, K.; and Pereira, F. 2005. Online large-margin training of dependency parsers. In *ACL*, 91–98.

Myers, S., and Leskovec, J. 2010. On the convexity of latent social network inference. In *NIPS*, 1741–1749.

Rodriguez, M. G., and Schölkopf, B. 2012. Submodular inference of diffusion networks from multiple trees. In *ICML*, 1–8.

Rodriguez, M. G.; Balduzzi, D.; and Schölkopf, B. 2011. Uncovering the temporal dynamics of diffusion networks. In *ICML*, 561–568.

Rodriguez, M. G.; Leskovec, J.; Balduzzi, D.; and Schölkopf, B. 2014. Uncovering the structure and temporal dynamics of information propagation. *Network Science* 2(01):26–65.

Rodriguez, M. G.; Leskovec, J.; and Schölkopf, B. 2013. Modeling information propagation with survival theory. In *ICML*, 666–674.

Rong, Y.; Zhu, Q.; and Cheng, H. 2016. A model-free approach to infer the diffusion network from event cascade. In *CIKM*, 1653–1662.

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

Smith, N. A., and Eisner, J. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *ACL*.

Smith, D. A., and Smith, N. A. 2007. Probabilistic models of nonprojective dependency trees. In *EMNLP-CoNLL*.

Snowsill, T. M.; Fyson, N.; De Bie, T.; and Cristianini, N. 2011. Refining causality: Who copied from whom? In *KDD*, 466–474.

Stack, J. C.; Bansal, S.; Kumar, V. A.; and Grenfell, B. 2012. Inferring population-level contact heterogeneity from common epidemic data. *Journal of the Royal Society Interface* rsif20120578.

Tutte, W. T. 1984. *Graph Theory*, volume 21 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley.

Wang, L.; Ermon, S.; and Hopcroft, J. E. 2012. Feature-enhanced probabilistic models for diffusion network inference. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 499–514.

Zhai, X.; Wu, W.; and Xu, W. 2015. Cascade source inference in networks: A Markov chain Monte Carlo approach. *Computational Social Networks* 2(1).