Proceedings of the Twelfth International AAAI Conference on Web and Social Media (ICWSM 2018)

Joint Modeling of Text and Networks for Cascade Prediction

Cheng Li
School of Information
University of Michigan, Ann Arbor
lichengz@umich.edu

Xiaoxiao Guo
Computer Science and Eng.
University of Michigan, Ann Arbor
guoxiao@umich.edu

Qiaozhu Mei
School of Information
University of Michigan, Ann Arbor
qmei@umich.edu

Abstract

A critical research problem about information cascades, which is a central topic of social network analysis, is to predict the potential influence or the future growth of cascades. Recent developments of deep learning have provided promising alternatives, which no longer rely on heavy feature engineering efforts and instead learn the representation of cascade graphs in an end-to-end manner. In reality, however, the influence of a cascade not only depends on the cascade graph and the global network structure, but also largely relies on the content of the cascade and the preferences of users. In this work, we extend the deep learning approaches to cascade prediction by jointly modeling the content and the structure of cascades. We find that text information provides a valuable addition for the learning of cascade graphs, especially when some users (nodes) have rarely participated in the past cascades. To this end, a gating mechanism is introduced to dynamically fuse the structural and textual representations of nodes based on their respective properties. Attentions are employed to incorporate the text information associated with both cascade items and nodes. Empirical experiments demonstrate that incorporating text information brings a significant improvement to cascade prediction, and that the proposed model outperforms alternative ways to combine text and networks.

Introduction

Online social network platforms have greatly expedited the diffusion of information, which plays a key role in many social network phenomena such as the adoption of innovations, viral marketing, rumor spreads, crowdsourcing, and persuasion campaigns, just to name a few.

Researchers have explored various features that are predictive of cascade growth, such as node centrality, network density, structural patterns, and textual content (Weng, Menczer, and Ahn 2014; Cheng et al. 2014; Cui et al. 2013; Jenders, Kasneci, and Naumann 2013). These studies are conducted over a wide range of information types being diffused, such as Tweets (Yu et al. 2015; Weng, Menczer, and Ahn 2014; Zhao et al. 2015; Jenders, Kasneci, and Naumann 2013; Cui et al. 2013; Guille and Hacid 2012), videos (Bauckhage, Hadiji, and Kersting 2015), photos (Cheng et al. 2014) and academic papers (Shen et al. 2014). One recently proposed model is called the DeepCas (Li et al. 2017), which learns the representation of cascade graphs and predicts cascade growth automatically, without arbitrary designs of features.

While one deep architecture works well for cascade graphs, it is not obvious how it could be extended to handle a different type of information, which as the focus of this paper, is the content of an information cascade. Indeed, the items being diffused in online social networks are usually in the form of a text message or can be described by text – Tweets, memes, and scientific papers. How far it will be passed along largely depends on its content (e.g., a political rumor travels faster than a movie review tweeted by the same user). On the other hand, who will pass these items along also highly depends on the preference or interest of the users who are exposed to the diffusion, which are often encoded in the rich text information associated to these users.

Researchers have explored various approaches to incorporating text content into the learning of node representations (Yang et al. 2015; Pan et al. 2016). How to go beyond node to learn representation for the entire graph (like what DeepCas does) with the presence of text content is not studied. In our context, however, simply treating graph structure as one modality and the aggregated text content from graph nodes as another leads to significant loss of information. This is due to the complexity of information networks – text is not present as a single document for a cascade, but resides in individual nodes, and a cascade is a complex system of nodes and edges. Moreover, independent from the graph structure is the item being diffused, which is also associated with textual content that is critical for the prediction. To utilize the structural and textual information of nodes in a better way, we introduce a gating mechanism to dynamically fuse the node representations from two sources, based on how well each representation is learned. To incorporate the text information from both the item being diffused and the nodes involved in the cascade, an attention mechanism is employed over the content of nodes. Empirical evaluations demonstrate that incorporating text information benefits the cascade prediction task significantly, and that our proposed model outperforms alternative methods to combine network and text.

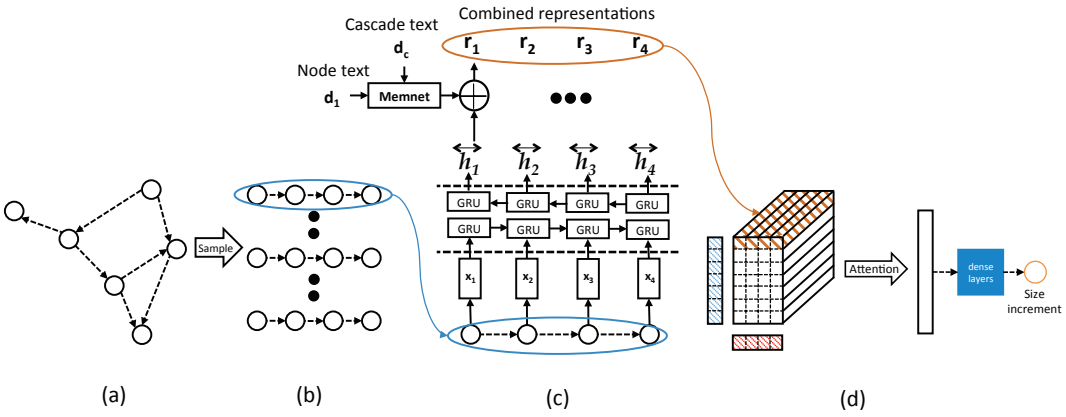

Figure 1: The end-to-end pipeline to jointly model structure and text.

Method

Basic DeepCas model. DeepCas works by making an analogy that cascade graphs are represented as documents, with nodes as words and paths as sentences. Given a cascade graph g_c as input, it first samples node sequences from it and feeds the sequences into a Gated Recurrent Unit (GRU), where attention mechanisms are specifically designed to learn how to assemble sequences into a "document", so that the future cascade size could be predicted.

Like DeepWalk and node2vec (Perozzi, Al-Rfou, and Skiena 2014; Grover and Leskovec 2016), node sequences are sampled from a cascade graph through random walks, in which a walker either transits to a neighboring node, performs a jump to a random node, or terminates the entire walking process with certain probability. These sequences are fed to GRUs to produce sequence representations.

To assemble the graph representation from the produced sequence representations, the number of required sequences and length of sequences per graph are learned through an attention mechanism. All the parameters produced in the pipeline can then be optimized towards predicting the future cascade size.

Joint modeling of structure and text. The proposed framework to model structure and text jointly is shown in Figure 1. It takes as input the cascade graph g_c and predicts the increment of cascade size Δs_c. Similar to DeepCas, it first samples node sequences from cascade graphs and then feeds the sequences into GRUs. Figure 1 (c) shows the major difference, where a gating mechanism is employed to fuse the structural and textural representation of nodes.

Modeling structure. In Figure 1 (c), the representation of node v output by GRU is $\overleftrightarrow{h}_v \in \mathbf{R}^{2H}$, where H is the size of embedding vectors. This representation, calculated based on its relationships to neighboring nodes, captures the structural information of each node. Therefore, $\overleftrightarrow{h}_v$ could be treated as the structural representation of node v.

Modeling text. The text representation of nodes is not a simple function of node text d_v, because it could change dynamically with respect to the cascade text d_c. Thus it needs to be calculated as $m_v = \phi'(d_v, d_c)$.

To account for this, we apply an attention mechanism over the node text. Specifically, denote the embedding of the i-th word as $d_{v,i} \in \mathbb{R}^H$ and the embedding of the query text as $d_c \in \mathbb{R}^H$. The output is then defined as

$$m_v = \sum_i \alpha_i d_{v,i} \tag{1}$$

$$\alpha_i = \text{SoftMax}(d_{v,i}^\intercal d_c) \tag{2}$$

where α_i is the attention between the i-th word and the cascade text.

Fusing structure and text representation. Simple concatenation learns to assign a global weight of importance to each source, failing to consider the uniqueness of nodes. Some nodes appear so frequently in cascades that structural embedding is already sufficient to represent the node. Some nodes have rich text information to be utilized. Finally, there are nodes with scarce information from both sources.

We design a gating mechanism to dynamically fuse the structural and textual representations of nodes. The gates control how information flows by measuring the informativeness of each source. The informativeness of structural information, $i_v^{(s)}$, can be simply measured by $\text{fq}(v)$, the frequency of node v occurred in the training set, scaled by logarithm:

$$i_v^{(s)} = \log(\text{fq}(v+1)). \tag{3}$$

To measure the informativeness of text, we compute the match between the node text and the cascade text, as users who are constantly promoting certain topics are likely to have larger influence in those topics. One thing to be noted is that there are also some general topics that match well with a large population of users. To account for this factor, we are actually measuring how much better the node text matches the cascade text than average. Similar to negative sampling (Mikolov et al. 2013), the match of average nodes can be approximately computed by randomly sampling a node set $V_s \subseteq V$. In this way, the informativeness of text $i_v^{(t)}$ is calculated as:

$$i_v^{(t)} = \psi(d_v)^T \psi(d_c) - \frac{1}{|V_s|} \sum_{v' \in V_s} \psi(d_{v'})^T \psi(d_c), \quad (4)$$

where $\psi(\cdot)$ computes the document representation simply by taking the average of word embeddings.

Given the two measures $i_v^{(s)}$ and $i_v^{(t)}$, we can now use two gates $g_v^{(1)}$ and $g_v^{(2)}$ to distinguish the three cases mentioned above:

$$g_v^{(1)} = \sigma(W^{(1)}[i_v^{(s)}; i_v^{(t)}]) \quad (5)$$

$$g_v^{(2)} = \sigma(W^{(2)}[i_v^{(s)}; i_v^{(t)}]) \quad (6)$$

$$r_v = (1 - g_v^{(1)}) \left((1 - g_v^{(2)}) \overleftrightarrow{h}_v + g_v^{(2)} m_v \right) + g_v^{(1)} e \quad (7)$$

where $\sigma(x)$ is the sigmoid function and e is an embedding vector learned globally that represents nodes with neither rich structural nor textual information. This global embedding e learns aggregated information for nodes with scarce information. If gate $g_v^{(1)}$ is close to one, it chooses information more from the global embedding e. This means that it is hard to extract any information from node v, and therefore we back off to the aggregated information. If $g_v^{(2)}$ is close to one, it allows more information from text, rather than structure, to flow through.

From sequence to graph representation The rest of the framework basically follows DeepCas, except that the formation of the final graph representation is based on the fused representation r_v, rather than the structural representation $\overleftrightarrow{h}_v$.

Experiment setup

Data Sets. We use TWITTER and AMINER as our data sets. In Twitter, a cascade is composed of the author of the original Tweet and other users who have retweeted it. The AMINER data set uses the DBLP citation network released by ArnetMiner [1]. A cascade of a particular paper involves all authors who have written or cited that paper.

To avoid using future information, we only use a piece of text of a node if the text is generated before training time. Specifically for TWITTER, we collect user tweets and retweets in April and May of 2016. The concatenation of all tweets of a user in this period is considered as the node text d_v, while the original tweet that starts the cascade is used as the cascade text d_c. For AMINER, we gather all titles of each author's publications between 1992 and 2002. Their concatenation is used as the node text d_v. The title of the paper that starts the citation cascade is used as the cascade text d_c.

Evaluation Metric We use the mean squared error (MSE) to evaluate the accuracy of predictions, which is a common choice for previous work of cascade prediction (Tsur and Rappoport 2012; Yu et al. 2015; Kupavskii et al. 2012; Li et al. 2017). Denote $\hat{y}$ a prediction value, and y the ground truth value, the MSE is: $\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$. Following existing literature(Kupavskii et al. 2012; Tsur and Rappoport 2012; Li et al. 2017), we predict a scaled version of the actual increment of the cascade size, i.e., $y_i = \log_2(\Delta s_i + 1)$. This is because the growth of all cascades follows a power-law distribution, where a large number of cascades did not grow at all after t.

Baseline methods. Apart from strong baselines compared with DeepCas (Li et al. 2017), we additionally compare with node embedding methods that model text content of nodes. (1) **Features-deep**. Structural features used in (Li et al. 2017) are all included, , which are features related to centrality, density, node identity, communities, and network substructures. In addition, we include text based features, including ngrams ($n = 1, 2, 3$), the average of word embeddings, and topic distribution of text, which is found by Latent Semantic Analysis (LSA) (Deerwester et al. 1990), from both nodes and cascades. The features are fed to a deep MLP network. (2) **Node2vec** (Grover and Leskovec 2016). We concatenate the average of node embeddings and word embeddings of both node and cascade text, which is fed through MLP to make the prediction. (3) **TriDNR** (Pan et al. 2016). Since we do not have node labels, we only optimize for the network structure and text objectives. As node text is already incorporated into the node embedding, we only use the average of node embedding and embedding of cascade text for prediction.

Experiment results

The overall performance of all competing methods across data sets are displayed in Table 1. The last row of each table shows the performance of the complete version of our methods, which outperform all baseline methods, including DeepCas, with a statistically significant drop of MSE.

Features-Deep, including all kinds of well-designed features, is not as strong as in the setting of (Li et al. 2017) – both node2vec and TriDNR are able to outperform feature-based method in some of the configurations. Though a large number of designed features, including the newly incorporated text-based features, are used as input, significant structure and/or text information and their interactions are lost. The loss of structural and textual information in Features-Deep harms its predictive ability.

Node2vec and TriDNR underperform DeepCas. This result is surprising because DeepCas does not consider any text input. Recall that both Node2vec and TriDNR eventually make predictions based on the average of node embeddings and the average of word embeddings. The poor performance of Node2vec and TriDNR confirms that taking the average of embeddings, both structural and textual, as the graph representation is not as informative as representing the graph as a set of paths. The simple average treatments lose crucial information in making accurate prediction on information cascades.

Conclusion

In this work, we explored how to model structure and text jointly in a deep learning architecture designed for cascade

[1] https://aminer.org/citation, DBLP-Citation-network V8.

Table 1: Performance measured by MSE (the lower the better), where original label Δs is scaled to $y = \log_2(\Delta s + 1)$.

(a) TWITTER

t	1 day			3 days			5 days		
Δt	1 day	3 days	5 days	1 day	3 days	5 days	1 day	3 days	5 days
Features-Deep	2.894^{***}	3.514^{***}	3.501^{***}	2.113^{***}	3.026^{***}	3.109^{***}	0.956^{***}	1.542^{***}	1.723^{***}
node2vec	2.387^{***}	2.936^{***}	2.930^{***}	1.980^{***}	2.731^{***}	2.706^{***}	1.053^{***}	1.687^{***}	1.855^{***}
TriDNR	2.678^{***}	3.439^{***}	3.510^{***}	2.131^{***}	3.062^{***}	3.143^{***}	1.109^{***}	1.887^{***}	2.161^{***}
DeepCas	2.102	2.758	2.772	1.465	2.004	2.020	0.907	1.410	1.494
DCGT	$1.945^{***}_{\triangledown\triangledown\triangledown}$	$2.144^{***}_{\triangledown\triangledown\triangledown}$	$2.397^{***}_{\triangledown\triangledown\triangledown}$	$1.371^{***}_{\triangledown\triangledown\triangledown}$	$1.862^{***}_{\triangledown\triangledown\triangledown}$	$1.871^{***}_{\triangledown\triangledown\triangledown}$	$0.890^{***}_{\triangledown\triangledown\triangledown}$	$1.312^{***}_{\triangledown\triangledown\triangledown}$	$1.392^{***}_{\triangledown\triangledown\triangledown}$

(b) AMINER

t	1 year			2 years			3 years		
Δt	1 year	2 years	3 years	1 year	2 years	3 years	1 year	2 years	3 years
Features-Deep	2.678^{*}	2.902^{**}	2.922^{***}	1.908^{***}	1.990^{***}	2.032^{***}	1.608^{**}	1.683^{***}	1.748^{***}
node2vec	2.466	2.663^{*}	2.706^{**}	1.902^{***}	2.046^{***}	2.073^{***}	1.697^{***}	1.786^{***}	1.832^{***}
TriDNR	2.586^{*}	2.821^{**}	2.866^{**}	1.971^{***}	2.110^{***}	2.130^{***}	1.678^{***}	1.763^{***}	1.806^{***}
DeepCas	2.425	2.556	2.576	1.826	1.898	1.914	1.575	1.607	1.643
DCGT	$2.301^{**}_{\triangledown}$	2.412^{**}	2.494^{*}	$1.742^{***}_{\triangledown\triangledown\triangledown}$	$1.818^{***}_{\triangledown\triangledown\triangledown}$	$1.820^{***}_{\triangledown\triangledown\triangledown}$	$1.482^{***}_{\triangledown\triangledown\triangledown}$	$1.529^{***}_{\triangledown\triangledown\triangledown}$	$1.502^{***}_{\triangledown\triangledown\triangledown}$

"***(**, *)" means the result is significantly better or worse over *DeepCas* according to paired t-test test at level 0.01(0.05, 0.1).

prediction. Text is present at different levels of the cascade graphs, where both the diffusion items and the nodes in the social network are associated with textual content. The proposed model, built upon recent work that predicts the future growth of a cascade from the structure of cascade graphs, successfully utilizes text to provide valuable complementary information for the learning of graphs. Such information is especially useful when their node members rarely participate in diffusions. A gating mechanism is introduced to dynamically fuse the structural and textual representations of nodes based on their respective properties. To incorporate the text information associated with both diffusion items and nodes, attentions are employed over node text based on their interactions with item text. Empirical evaluations demonstrate that incorporating text information significantly improves cascade prediction, and that our proposed model outperforms alternative methods that combines text and network information. The new gating and attention mechanisms provide a general approach to incorporating multiple types of information in a deep learning architecture and can be extended to incorporate other signals for cascade prediction, such as time series and user profiles.

Acknowledgments This work is partially supported by the National Science Foundation under grant numbers IIS-1054199, IIS-1633370 and SES-1131500.

References

Bauckhage, C.; Hadiji, F.; and Kersting, K. 2015. How viral are viral videos. In *Proc. of ICWSM*.

Cheng, J.; Adamic, L.; Dow, P. A.; Kleinberg, J. M.; and Leskovec, J. 2014. Can cascades be predicted? In *Proc. of WWW*.

Cui, P.; Jin, S.; Yu, L.; Wang, F.; Zhu, W.; and Yang, S. 2013. Cascading outbreak prediction in networks: a data-driven approach. In *Proc. of SIGKDD*.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proc. of SIGKDD*.

Guille, A., and Hacid, H. 2012. A predictive model for the temporal dynamics of information diffusion in online social networks. In *Proc. of WWW*.

Jenders, M.; Kasneci, G.; and Naumann, F. 2013. Analyzing and predicting viral tweets. In *Proc. of WWW*.

Kupavskii, A.; Ostroumova, L.; Umnov, A.; Usachev, S.; Serdyukov, P.; Gusev, G.; and Kustarev, A. 2012. Prediction of retweet cascade size over time. In *Proc. of CIKM*.

Li, C.; Ma, J.; Guo, X.; and Mei, Q. 2017. Deepcas: an end-to-end predictor of information cascades. In *Proc. of WWW*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Pan, S.; Wu, J.; Zhu, X.; Zhang, C.; and Wang, Y. 2016. Tri-party deep network representation. *Network*.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proc. of SIGKDD*.

Shen, H.-W.; Wang, D.; Song, C.; and Barabási, A.-L. 2014. Modeling and predicting popularity dynamics via reinforced poisson processes. *arXiv preprint arXiv:1401.0778*.

Tsur, O., and Rappoport, A. 2012. What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In *Proc. of WSDM*.

Weng, L.; Menczer, F.; and Ahn, Y.-Y. 2014. Predicting successful memes using network and community structure. *arXiv preprint arXiv:1403.6199*.

Yang, C.; Liu, Z.; Zhao, D.; Sun, M.; and Chang, E. Y. 2015. Network representation learning with rich text information. In *Proc. of IJCAI*.

Yu, L.; Cui, P.; Wang, F.; Song, C.; and Yang, S. 2015. From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics. In *Proc. of ICDM*.

Zhao, Q.; Erdogdu, M. A.; He, H. Y.; Rajaraman, A.; and Leskovec, J. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proc. of SIGKDD*.