Social Influence Prediction Report for Project of EE447

Li Ziyi 517030910319; Yue Ye 517030910357; Wu Shushu 517030910338; Liu Du 517030910346

June 19, 2020

This paper acts as the report for the final project of EE447. In this project, we study the influence prediction process in social network by using deep learning methods. We also compare the model with state-of-the-art model as baseline. The result is excellent, and detailed information is as follows.

influence prediction | social network | network embedding | GCN

1. Introduction

Social influence is ubiquitous around us, not only in our daily life but also on the virtual Web space. Generally, the term social influence refers to the phenomenon that a person's emotions, opinions, or behaviors are affected by others. With the development of network technology, Social and information networking activities such as on Facebook, Twitter, WeChat, and Weibo have become an indispensable part of our everyday life. So a social influence prediction for each user is critical for a variety of applications such as online recommendation and advertising.

According to the different type of the social influence, social influence prediction can be divided into two parts, as shown in Fig.1 and Fig.2. One aims to predict the global or aggregated patterns of social influence such as the cascade size within a time-frame. The other aims to predict the social influence for each individual such as the action status of a user.

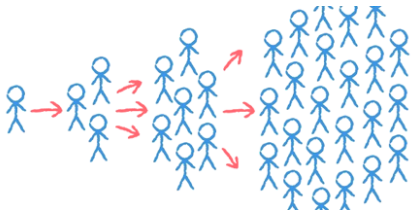

Fig. 1. Aggregated/global social influence

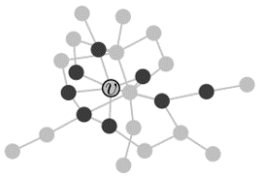

Fig. 2. User-level social influence

In our project, we focus on the prediction of user-level social influence. We aim to predict the action status of the user given his social relationship the action status of his neighbours. First, we formulate this problem as a binary graph classification problem. Then inspired by the recent work in word embedding and graph convolution, we design a deep learning based framework. In specific, we get the sub-network of the user, and take the use of word embedding and graph convolution techniques to learn the hidden features and predict the action status of the user. We show the improvement on four social networks in various domains - OAG, Digg, Twitter and Weibo.

Contribution of each member In our group, Liu Du and Wu Shushu are responsible for the algorithm design: Liu Du designs the embedding layer and Wu Shushu designs the GCN layer. Li Ziyi is responsible for the problem formulation and the overall design of the framework. Yue Ye is responsible for the dataset-preprocessing and the experiment setup. At last, Li Ziyi and Yue Ye complete the codes and experiment.

2. Problem Formulation

Let $G = (V, E)$ be a common social network, where V denotes the set of users and E denotes the set of edges. In specific, $V = (v, s)$, where v denotes the node and s denotes the action state of node. $E = (u, v, w, r)$, where u and v denote the two nodes of the edge and w denote the weight of the edge, r denotes the relationship of the two users, such as they are follower of each other or only one is follower of the other one.

Then for each user v, we find its neighbours $\Gamma_v^r = \{u : d(u, v) \leq r\}$, where $d(u, v)$ is the distance between u and v in network G. And this sub-network is denoted as G_v^r as Fig.3. For each user, we define a binary action status: $s_u \in \{0, 1\}$, where $s_u = 1$ means u performs this action, otherwise he doesn't perform this action. Naturally, we have the set of action status of v's neighbors: $S_v = \{s_u : u \in \Gamma_v^r \setminus \{v\}\}$.

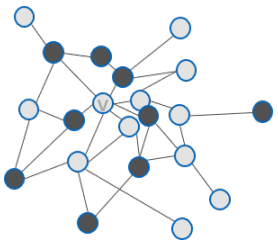

Fig. 3. $G_v^r : \Gamma_v^r = \{u : d(u, v) \leq r\}$

Further, given G_v^r and S_v, we can express the activation probability of v:

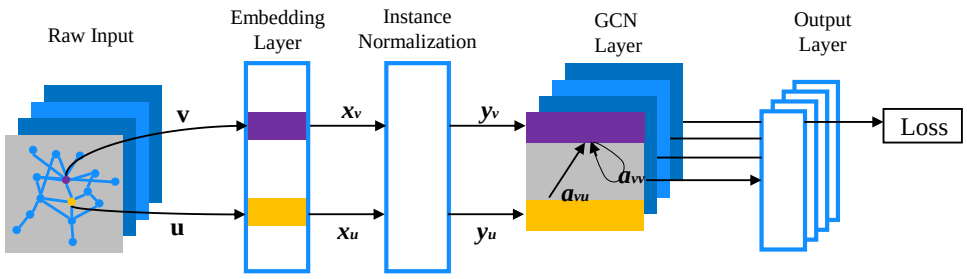

Fig. 4. The overview framework of our proposed model

$$P(s_v|G_v^r, S_v)$$

So this social influence prediction problem can be reformulate as a binary graph classification problem which can be solved by minimizing the following negative log likelihood objective:

$$L(\Theta) = -\sum_{i=1}^{N} \log(P_\theta(s_{v_i}|G_{v_i}^r, S_{v_i}))$$

where Θ are the parameters of this machine learning model. Now what we should do is to train a learning model using machine learning knowledge.

3. Model Framework

We design an neural network model to solve this problem. The input is G_v^r and S_v and the output is a hidden representation for v to predict his action status. As shown in Fig.4, our proposed neural network consists of an embedding layer and several graph convolution layers. Then we will introduce these layers in detail.

A. Embedding layer. In order to learn latent representations for the network, network embedding is required to be done on the dataset. DeepWalk has been commonly considered as powerful benchmark solutions for evaluating network embedding research and a study points out that DeepWalk can be viewed as implicit factorization of a closed-form matrix. Based on this idea, NetMF was proposed to factorize this matrix which achieved more effective embeddings than DeepWalk. The matrix is as follow.

$$\text{tran\_log}\left(\frac{\text{vol}(G)}{b}\boldsymbol{M}\right),$$

where $\text{vol}(G) = \sum_i \sum_j \boldsymbol{A}_{ij}$ ($\boldsymbol{A}$ is the adjacent matrix of G) denotes the volume of the graph, b is the number of negative samples, $\text{tran\_log} = \max(0, \log(x))$, and,

$$\boldsymbol{M} = \frac{1}{T}\sum_{r=1}^{T}(D^{-1}A)^r D^{-1}$$

where $D = \text{diag}(d_1, ..., d_n)$ is the degree matrix with $d_i = \sum_j \boldsymbol{A}_{ij}$ and T is the context window size.

But when it comes to the factorization of a dense matrix, NetMF turns really time-consuming. In our project, we use NetSMF instead of DeepWalk or NetMF to achieve network embeddings. The basic idea of NetSMF is the same as NetMF, while it turns to find a sparse matrix which is spectrally similar to the original one and factorize it with relatively low cost. We define $\widetilde{\boldsymbol{M}}$ as

$$\widetilde{M} = D^{-1}(D - \widetilde{L})D^{-1}$$

where $\widetilde{L}$ is a sparsifier of random-walk polynomial of G. Then we get a sparse alternative as below.

$$\text{tran\_log}\left(\frac{\text{vol}(G)}{b}\widetilde{\boldsymbol{M}}\right),$$

B. GCN Layer. Graphs are a kind of data structure which models a set of objects (nodes) and their relationships (edges). In this project we use graphs to model social networks and to predict social influence. Having done embedding, we can get a vertex feature matrix $\boldsymbol{H}^{(0)} \in \mathbb{R}^{N\times D}$, where N is the number of nodes and D is the number of input features.

Recently, researches of analyzing graphs with machine learning have been receiving more and more attention. Graph neural networks (GNNs) are deep learning based methods that operate on graph domain. The propagation step and output step are of vital importance in the model to obtain the hidden states of nodes (or edges). Different propagation methods created different GNN variants. There is an increasing interest in generalizing convolutions to the graph domain. Advances in this direction are often categorized as spectral approaches and non-spectral (spatial) approaches.

Kipf and Welling (2016) proposed a semi-supervised Graph Convolution Network(GCN) based on a a localized first-order approximation of spectral graph convolutions.

For GCN models, the goal is to learn a function of features on a graph $G = (V, E)$ which takes as input:

- A feature description xi for every node i; summarized in a $N \times D$ feature matrix $\boldsymbol{H}$ (N: number of nodes, D: number of input features)
- A representative description of the graph structure in matrix form; typically in the form of an adjacency matrix $\boldsymbol{A}$ (or some function thereof)

Every neural network layer can then be written as a nonlinear function

$$\boldsymbol{H}^{(l+1)} = f(\boldsymbol{H}^{(l)}, \boldsymbol{A})$$

with l being the number of layers.

For GCN, the layer-wise propagation rule is as follows :

$$f(\boldsymbol{H}^{(l)}, \boldsymbol{A}) = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}W^{(l)})$$

Here, $\hat{A} = \boldsymbol{A} + I_N$ is the adjacency matrix of the undirected graph G with added self-connections. I_N is the identity matrix. This is because if we don't add the identity matrix, multiplication with $\boldsymbol{A}$ means that, for every node, we sum up all the feature vectors of all neighboring nodes but not the node itself.

$\hat{D}$ is the diagonal node degree matrix of $\hat{A}$. And we used a symmetric normalization. Because $\boldsymbol{A}$ is typically not normalized and therefore the multiplication with $\boldsymbol{A}$ will completely change the scale of the feature vectors.

For a spectral network, the convolution operation is defined in the Fourier domain by computing the eigendecomposition of the graph Laplacian. The operation can be defined as the multiplication of a signal $\boldsymbol{x} \in \mathbb{R}^N$ (a scalar for each node) with a filter $\mathbf{g}_\theta = (\theta)$ parameterized by $\theta \in \mathbb{R}$:

$$\mathbf{g}_\theta \star \mathbf{x} = \mathbf{U}\mathbf{g}_\theta(\mathbf{\Lambda})\mathbf{U}^T\mathbf{x}$$

where $\mathbf{U}$ is the matrix of eigenvectors of the normalized graph Laplacian $\mathbf{L} = I_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, with a diagonal matrix of its eigenvalues $\mathbf{\Lambda}$ and $\mathbf{U}^T\mathbf{x}$ being the graph Fourier transform of $\boldsymbol{x}$. We can understand $\mathbf{g}_\theta$ as a function of the eigenvalues of $\mathbf{L}$, i.e. $\mathbf{g}_\theta(\mathbf{\Lambda})$.

4. Experiment Setup

To quantitatively evaluate the proposed deep-learning framework, we set up our experiments with large-scale real-world datasets which are described as follows.

A. Datasets. Our experiments are implemented on four social networks in various domains – OAG, Digg, Twitter, and Weibo.

OAG Open Academic Graph dataset is generated by linking two large academic graphs: Microsoft Academic Graph Dong et al. (2017b) and Aminer Tang et al. (2008). The social network is defined to be the co-author network, and the social action is defined to be citation behaviors – a researcher cites a paper from the above conferences. We are interested in how ones citation behaviors are influenced by her collaborators.

Digg Digg Hogg and Lerman (2012) is a news aggregator which allows people to vote web content, up or down. The dataset contains data about stories promoted to Diggs front page over a period of a month in 2009. For each story, it contains the list of all Digg users who have voted for the story up to the time of data collection and the time stamp of each vote. The voters friendship links are also retrieved.

Twitter The Twitter dataset De Domenico et al. (2013) was built after monitoring the spreading processes on Twitter before, during and after the announcement of the discovery of a new particle with the features of the elusive Higgs boson on Jul. 4th, 2012. The social network is defined to be the Twitter friendship network, and the social action is defined to be whether a user retweets Higgs related tweets.

Weibo Weibo is the most popular Chinese microblogging service. The complete dataset Zhang et al. (2013) contains the directed following networks and tweets (posting logs) of 1,776,950 users between Sep. 28th, 2012 and Oct. 29th, 2012. The social action is defined as retweeting behaviors in Weibo a user forwards (retweets) a post (tweet).

B. Evaluation Metric. We evaluate the predictive performance of the framework in terms of the following metrics:

Area Under Curve(AUC) When using normalized units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative'). This can be seen as follows: the area under the curve is given by (the integral boundaries are reversed as large T has a lower value on the x-axis)

Precision(Prec.) Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of the total amount of relevant instances that were actually retrieved. Both precision and recall are therefore based on an understanding and measure of relevance.

$$\text{Prec.} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall (Rec.)

$$\text{Rec.} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

F_1-Measure (F_1) In statistical analysis of binary classification, the F_1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

$$\text{F}_1 = 2 \cdot \frac{\text{Prec.} \cdot \text{Rec.}}{\text{Prec.} + \text{Rec.}}$$

C. Comparison Methods. Baselines are implemented in this section.

PSCN As we model social influence locality prediction as a graph classification problem, we compare our framework with the state-of-the-art graph classification models, PSCN Niepert et al. (2016). For each graph, PSCN selects w vertices according to a user-defined ranking function, e.g., degree and betweenness centrality. Then for each selected vertex, it assembles its top k near neighbors according to breadth-first search order. For each graph, The above process constructs a vertex sequence of length $w \times k$ with F channels, where F is the number of features for each vertex. Finally, PSCN applies 1-dimensional convolutional layers on it.

5. Evaluation and Results

As shown in Table 1, GCN achieves significantly better performance over baselines in terms of both AUC and F1,

demonstrating the effectiveness of our proposed framework. In OAG and Digg, GCN discovers the hidden mechanism and dynamics of social influence locality.

For PSCN, it selects a subset of vertices according to a user-defined ranking function. As mentioned before, instead of using betweenness centrality, we propose to use BFS order-based ranking function. Such ranking function can be regarded as a predefined graph attention mechanism where the ego user pays much more attention to her active neighbors.

Data	Model	AUC	Prec.	Rec.	F1
OAG	PSCN	68.69	36.04	64.10	46.14
	GCN	**70.17**	**40.74**	**73.23**	**47.31**
Digg	PSCN	81.73	63.22	62.37	62.79
	GCN	**85.52**	**62.45**	**67.74**	**64.99**
Twitter	PSCN	78.74	48.32	64.91	55.40
	GCN	**80.84**	**51.85**	**66.72**	**58.64**
Weibo	PSCN	81.43	47.65	72.51	57.51
	GCN	**82.99**	**50.19**	**72.59**	**59.34**

Table 1. Prediction performance of different methods on the four datasets (%)

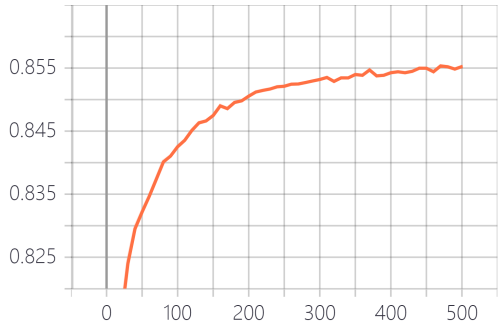

Fig. 5. Test AUC-Epochs in Digg dataset

6. Related Work

The term Social Networks (SNS) was first used by Barnes Barnes (9547) in the Human Relations Journal in 1954. Social networks originated from e-mail and are now the most widely used applications. With the evolution of social networks, there are more and more new platforms, e.g., Facebook and Flickr in 2004, YouTube in 2005, Twitter in 2006, and Sina Weibo in 2009. The ways in which people obtain information have changed. In the past, individuals were passive receivers of information yet now they are its active publishers and communicators.

Most existing work focused on social influence modeled as a global-social process, such as the cascade size. Only a few have explored the user-level social influence. For global-social influence problem, researchers have made great progress since 2003 this problem was put forward by David Kempe Kempe et al. (2003). Edith Cohen and Daniel Delling solve the scalability problem of influence computation Cohen et al. (2014). Xudong Wu and Luoyi Fu proposed a novel framework for group-level location promotion in social network Wu et al. (2018). Lichao Sun1 and Weiran Huang studied the multi-Round influence maximization (MRIM) problem, where influence propagates in multiple rounds independently from possibly different seed sets Sun et al. (2018).

On the other hand, the studies on the user-level in social influence where each user is only influenced by her neighbours. Examples of such work include pairwise influence Goyal et al. (2010), topic-level influence Tang et al. (2009), structural diversity Ma (2013) and group formation Qiu et al. (2016). Such user-level models act as fundamental building blocks of many real-world problems and applications.

Word embedding and GCN both belongs to the area of representation learning. It is a very hot topic in nowadays graph learning. These techniques are then widely used in social network problem. Word(vertex) embedding aims to learn a low-dimensional latent features for each vertex, such as DeepWalk Perozzi et al. (2014), node2vec Grover and Leskovec (2016), metapath2vec Dong et al. (2017a), NetMF Qiu et al. (2018) and so on.

7. Conclusion

In this project, we study the influence prediction process in social network by using word embedding and deep learning methods. We reformulate this problem and design a deep learning based framework. We also combine the word embedding method and using four datasets to test out framework. At last, we compare our model with state-of-the-art method and shows the excellence of our framework.

Writing contribution of each member:
Introduction – Li Ziyi
Problem Formulation – Li Ziyi
Model-Embedding layer – Liu Du
Model-Gcn layer – Wu Shushu
Experiment Setup – Yue Ye
Evaluation and Results – Yue Ye
Related work – Li Ziyi
Conclusion – Li Ziyi

References

Barnes, J. (1954,7). Class and committees in a norwegian island parish, human relations. *Hum. Relat.*, page 3958.

Cohen, E., Delling, D., Pajor, T., and Werneck, R. (2014). Sketch-based influence maximization and computation: Scaling up with guarantees.

De Domenico, M., Lima, A., Mougel, P., and Musolesi, M. (2013). The anatomy of a scientific rumor. *Scientific reports*, 3:2980.

Dong, Y., Chawla, N., and Swami, A. (2017a). metapath2vec: Scalable representation learning for heterogeneous networks. pages 135–144.

Dong, Y., Ma, H., Shen, Z., and Wang, K. (2017b). A century of science: Globalization of scientific collaborations, citations, and innovations. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1437–1446.

Goyal, A., Bonchi, F., and Lakshmanan, L. (2010). Learning influence probabilities in social networks. volume 241-250, pages 241–250.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. volume 2016, pages 855–864.

Hogg, T. and Lerman, K. (2012). Social dynamics of digg. *EPJ Data Science*, 1(1):5.

Kempe, D., Kleinberg, J., and Tardos, E. (2003). Maximizing the spread of influence through a social network. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 137-146.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks.

Ma, H. (2013). An experimental study on implicit social recommendation. pages 73–82.

Niepert, M., Ahmed, M., and Kutzkov, K. (2016). Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., and Tang, J. (2018). Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. pages 459–467.

Qiu, J., Li, Y., Tang, J., Lu, Z., Ye, H., Chen, B., Yang, Q., and Hopcroft, J. (2016). The lifecycle and cascade of wechat social messaging groups. pages 311–320.

Sun, L., Huang, W., Yu, P., and Chen, W. (2018). Multi-round influence maximization. pages 2249–2258.

Tang, J., Sun, J., Wang, C., and Yang, Z. (2009). Social influence analysis in large-scale networks. volume 807-816, pages 807–816.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. (2008). Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998.

Wu, X., Fu, L., Yao, Y., Fu, X., and Chen, G. (2018). Glp: A novel framework for group-level location promotion in geo-social networks. *IEEE/ACM Transactions on Networking*, PP:1–14.

Zhang, J., Liu, B., Tang, J., Chen, T., and Li, J. (2013). Social influence locality for modeling retweeting behaviors. In *Twenty-Third International Joint Conference on Artificial Intelligence*.