Public Opinion Field Effect and Hawkes Process Join Hands for Information Popularity Prediction (Appendix)

Junliang Li[1], Yajun Yang[1*], Yujia Zhang[2], Qinghua Hu[1], Alan Zhao[2], Hong Gao[3]

[1]College of Intelligence and Computing, Tianjin University, China
[2]AI Technology Center of OVB, Tencent, China
[3]School of Computer Science and Technology, Zhejiang Normal University, China
{lijunliang, yjyang, huqinghua}@tju.edu.cn, {yujiazhang, alantzhao}@tencent.com, honggao@zjnu.edu.cn

Datasets Processing Details

In this section, we describe how the dataset is processed to obtain the inputs for the model.

Twitter and Douban

Twitter dataset (Hodas and Lerman 2014) contains tweets from the Twitter platform during October 2010, their propagation paths among users, and the friendships between users. Douban (Zhong et al. 2012) is collected from a social platform where users post updates about the books they read or the movies they watch. We use whether users read the same book to construct a friendship network.

Both datasets belong to the social network type. The format of the dataset is: "user_1, timestamp_1, user_2, timestamp_2...", where each line represents a topic. We filter out cascades with a length greater than 5, and take the median of all times as the observation time. Then, we count the labels of each cascade based on the observation time, and further delete the cascades with zero labels.

Finally, we use node2vec (Grover and Leskovec 2016) and friendship networks to extract the initial embeddings of all users, and aggregate the initial embeddings of all users connected to the same topic to get the initial embedding of the topic node.

Android and Christianity

Both Android and Christianity (Sankar et al. 2020) are collected from the community question-and-answer platform StackExchange. The cascade corresponds to a series of posts associated with the same tag arranged in chronological order, and users' interactions on multiple channels such as asking questions and commenting constitute their friendship network.

The format of the dataset is "root1 user_1 timestamp_1 user_2 timestamp_2...". Similar to the previous social network dataset, we filter out sequences with a length greater than 5 and classify sequences with the same root as the same topic. Finally, node2vec is used to obtain the initial embeddings of the user and topic nodes.

Evaluation Metrics

To assess the performance of comparative methods, we employed three widely-used metrics: Mean Squared Logarithmic Error (MSLE), Mean Absolute Logarithmic Error (MALE), and Symmetric Mean Absolute Percentage Error (SMAPE). The formulation of all evaluation metrics is defined as:

$$\mathrm{MSLE} = \frac{1}{Y}\sum_{i=1}^{Y}(\log(1+\Delta R_{C_i}) - \log(1+\widehat{\Delta R_{C_i}}))^2 \quad (1)$$

$$\mathrm{MALE} = \frac{1}{Y}\sum_{i=1}^{Y}|\log(1+\Delta R_{C_i}) - \log(1+\widehat{\Delta R_{C_i}})| \quad (2)$$

$$\mathrm{SMAPE} = \frac{1}{Y}\sum_{i=1}^{Y}\frac{|\log(1+\Delta R_{C_i}) - \log(1+\widehat{\Delta R_{C_i}})|}{(|\log(1+\Delta R_{C_i})| + |\log(1+\widehat{\Delta R_{C_i}})|)/2} \quad (3)$$

where Y represents the number of samples in the test set, and $\log$ represents the logarithm with the natural exponential e as the base.

Parameter Settings

The feature dimensions of user and topic are both 64, all models are 2-layer, with hidden and output layer dimensions of 32 and 64 respectively, and the number of attention heads (if applicable) is 4. In POFHP, we set $\beta = 2.0$, $\rho = 0.01$, and use a single-layer GRU. For all neural network models, the learning rate is 0.001 and the number of training epochs is 200. We run 10 times with the same partition and report the average results.

When using node2vec to get the initial embedding of the node, we set the embedding dimension to 64, the return parameter to 1, the in-out parameter to 1, the path length to 20, and the window size to 10. The parameters of all compared models follow the default settings in their original papers.

We use NVIDIA A100 (40GB) GPU and implement all models with PyTorch and e PyTorch Geometric (PyG) package (Fey and Lenssen 2019).

*Corresponding author.

References

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864.

Hodas, N. O.; and Lerman, K. 2014. The simple rules of social contagion. *Scientific reports*, 4(1): 4343.

Sankar, A.; Zhang, X.; Krishnan, A.; and Han, J. 2020. Inf-VAE: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. In *Proceedings of the 13th international conference on web search and data mining*, 510–518.

Zhong, E.; Fan, W.; Wang, J.; Xiao, L.; and Li, Y. 2012. Comsoc: adaptive transfer of user behaviors over composite social network. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 696–704.