



CollaborateCas: Popularity Prediction of Information Cascades Based on Collaborative Graph Attention Networks

Xianren Zhang^{1,3}, Jiaying Shang^{2,3}(✉), Xueqi Jia^{2,3}, Dajiang Liu^{2,3}, Fei Hao⁴, and Zhiqing Zhang¹

¹ CQU-UC Joint Co-op Institute, Chongqing University, Chongqing, China
zhangxr2000@foxmail.com, zqzhang@cqu.edu.cn

² College of Computer Science, Chongqing University, Chongqing, China
{shangjx,liudj}@cqu.edu.cn

³ Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing, China

⁴ School of Computer Science, Shaanxi Normal University, Xian, China
fhao@snnu.edu.cn

Abstract. In recent years, with the prosperity of online social media platforms, cascade popularity prediction has attracted much attention from both academia and industry. Due to the recent advance in graph representation learning technologies, many state-of-the-art prediction methods utilize graph neural network to predict the cascade popularity. However, a significant disadvantage shared by these methods is that they treat each cascade independently, while the collaborations among different cascades are ignored. Therefore, in this paper we propose a novel deep learning model **CollaborateCas** which utilizes collaborations among different cascades to learn node and cascade embeddings directly and simultaneously. To this end, we first construct a heterogeneous user-message bipartite graph where different cascades are indirectly connected by common participants. To further capture temporal interdependence among users within each cascade, we construct homogeneous cascade graphs where temporal information is modeled as edge features. Experimental results on two real-world datasets show that our approach achieves significantly higher prediction accuracy compared with state-of-the-art approaches.

Keywords: Information diffusion · Cascade popularity prediction · Graph neural network · Heterogeneous graph · Deep learning

1 Introduction

Recent years have witnessed the prosperity of various online social media platforms which allow users to generate and share various online contents through comments, likes, or retweets. Consequently, the investigation of information diffusion over online social media has attracted much attention [18]. It finds application in a lot of important scenarios such as viral marketing [9], rumor detection

[16], etc. Among many of the research topics related to information diffusion, cascade popularity prediction [3], which aims to predict the future popularity of online contents based on their early diffusion patterns, is a key issue.

To address the cascade popularity prediction problem, a lot of research efforts have been devoted. Recently, deep learning techniques have shown their superiority in automatically capturing valuable information from cascades and predicting cascade popularity in an end-to-end manner [12]. Some approaches [2, 12] represented cascades as multiple node sequences and then fed them into Recurrent Neural Network (RNN) models [5, 10]. To extract underlying diffusion patterns, some researches applied Graph Neural Network (GNN) models [1, 6] on cascade graphs [4] or social networks [3, 11, 14].

Motivation. Although GNN-based approaches have shown high prediction accuracy, a significant disadvantage shared by them is that they treat each cascade independently, while the collaborations among different cascades are ignored. In fact, according to the research of Myers et al. [13], when there are multiple messages spreading over the online social media, these messages will implicitly interact with each other, including both competition and cooperation effects among different cascades. On the one hand, messages with similar content and topics would have a higher chance to be shared by users if they are exposed multiple times to the same user. On the other hand, each user has limited attention with respect to tremendous online contents, thus different messages would implicitly compete with each other [17]. Therefore, it is worthwhile to consider the implicit interactions among different cascades.

Challenges. There are two key challenges in predicting the popularity of cascades when considering the aforementioned factors. The first challenge is how to capture collaborations among different cascades. To this end, instead of treating each cascade independently, multiple cascades should be considered comprehensively and fine-grained user-message interactions should be included into the learning model to get informative cascade embeddings. The second challenge is how to effectively merge temporal and structural information within each cascade. Temporal information can describe the influence of message and predecessors on users' diffusion behavior. Most current methods model temporal information as a chain and use RNN to capture the memory effects. However, modeling temporal information as a chain cannot capture the inter-dependence in tree-like cascade graphs.

To address the above challenges, we propose a novel deep learning model named **CollaborateCas**, which utilizes collaborations among different cascades to learn node and cascade embeddings directly. Specifically, for the first challenge, a heterogeneous user-message bipartite graph is built where users and cascades are represented as two types of nodes and the interactions between users and cascades are taken as edges. Then a type-aware Graph Attention Network (GAT) [15] model is designed to learn representations for the two types of nodes. To deal with the second challenge and based on the observation that users would have different reaction time for different early adopters, we take the difference of infection time as users' edge features in the homogeneous cascade

graphs. The proposed approach is tested on two real world datasets and results show that our model significantly outperforms state-of-the-art baselines in terms of prediction accuracy.

In general, the main contributions of our work are as follows:

- For the cascade popularity prediction problem, we make the first attempt to model user-message interactions as a heterogeneous bipartite graph and design a type-aware GAT model to learn user and cascade embeddings simultaneously. Our model is able to capture collaborations among different cascades by learning from the fine-grained user-message interactions.
- Time differences of early adopters and later users are taken as temporal information and encoded into edge features in homogeneous cascade graphs. The temporal and structural information within each cascade graph are used to capture the inter-dependence and attractiveness among different users.
- The proposed approach is evaluated on two real-world datasets. Experimental results indicate that CollaborateCas significantly outperforms state-of-the-art baselines and the average prediction error is reduced by 9.01% and 5.68% respectively on the two datasets.

2 Problem Formulation

We first introduce some preliminaries and basic definitions to formulate the investigated problem.

Definition 1 (Cascade Set). *The data can be represented as a cascade set $\mathcal{C}^T = \{C_c^T | c \in \mathcal{M}\}$ which contains cascades with respect to the set of messages \mathcal{M} within the observation time window T . Each cascade C_c^T can be represented as a set of tuples $\{(u, v, t) | t \leq T\}$, where (u, v, t) indicates that user v retweeted the message c from user u at time t within the observation time T .*

The purpose of our model is to predict the incremental size of cascade based on observations within a specific time window. Therefore, we define incremental size as follows:

Definition 2 (Incremental Size). *The incremental size of a cascade C_c^T with observation time T after a given time interval Δt is defined as $\Delta S_c = |C_c^{T+\Delta t}| - |C_c^T|$, where $|C_c^T|$ indicates the total number of retweeting behaviors with respect to this cascade by time T .*

Based on the aforementioned definitions, we define the cascade popularity prediction problem as follows:

Definition 3 (Cascade Prediction Problem). *Give a cascade $C_c^T \in \mathcal{C}^T$ within the observation time window T , the cascade popularity prediction problem aims to learn a function $f(\cdot)$ that maps the homogeneous cascade graph $G_c(V, E)$ and heterogeneous bipartite graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ to $\Delta S_c = |C_c^{T+\Delta t}| - |C_c^T|$.*

3 Methodology

This section will give detailed illustration about our CollaborateCas model. The overall architecture of our deep learning model is shown in Fig. 1.

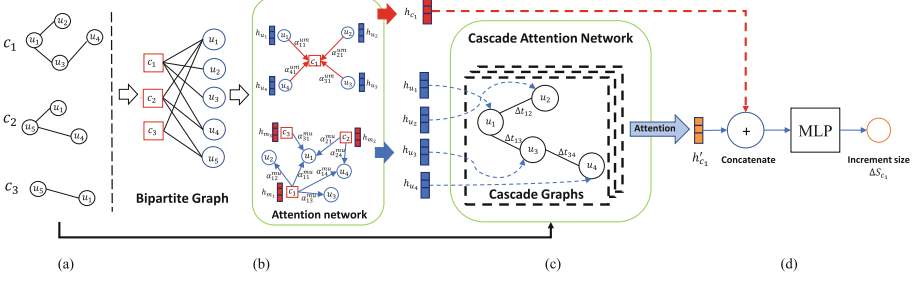


Fig. 1. Overview of CollaborateCas: (a) Input: a cascades set \mathcal{C}^T within observation time T ; (b) A heterogeneous bipartite graph is built based on observed cascades and embeddings are learned with a type-aware attention mechanism; (c) User embeddings learned in the previous step are fed into local cascade graph and temporal information is taken as edge features; (d) Both embeddings are concatenated together and then fed into MLP for final prediction.

3.1 Heterogeneous Bipartite Graph Learning

Based on observed cascades, we construct a global user-message graph to explicitly show relationships between messages and users. Since our model involves two different types of nodes, we design a type-aware attention mechanism and use different weights, i.e., W_{um} and W_{mu} to make a distinction between two different information gathering directions. Let

$$\theta_{ij}^{um} = \vec{a}_{um}^T [W_{um} \vec{h}_{c_i} || W_{um} \vec{h}_{u_j}] \quad (1)$$

$$\theta_{ij}^{mu} = \vec{a}_{mu}^T [W_{mu} \vec{h}_{u_i} || W_{mu} \vec{h}_{c_j}] \quad (2)$$

Where \vec{a}_{um} and W_{um} are weights from user to message. \vec{a}_{mu} and W_{mu} are weights from message to user. Then, θ^{um} and θ^{mu} are used to generate attention coefficients by softmax function. The embeddings are updated as follows:

$$\vec{h}_{c_i} = f\left(\sum_{j \in N_i} \alpha_{ij}^{um} W_u \vec{h}_{u_j}\right) \quad (3)$$

$$\vec{h}_{u_i} = f\left(\sum_{j \in N_i} \alpha_{ij}^{mu} W_m \vec{h}_{c_j}\right) \quad (4)$$

Where N_i is the set of neighbors in bipartite graph. \vec{h}_{c_i} and \vec{h}_{u_i} are embeddings after updating.

3.2 Homogeneous Cascade Graph Learning

In our work, a modified attention mechanism is designed to incorporate temporal information into the graph attention network model. Specifically, we have:

$$\theta_{ij} = f_{mlp}(\Delta t_{ij}) \cdot [W\vec{h}_{u_i} || W\vec{h}_{u_j}] \quad (5)$$

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\theta_{ij}))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(\theta_{ik}))} \quad (6)$$

where Δt_{ij} is the time difference between user i and user j , c is the corresponding cascade, $f_{mlp}()$ is a MLP which is used to project time difference scalar to higher dimensional embedding. Then the cascade embedding is obtained through an attention-based pooling:

$$\vec{h}'_c = \sum_{i \in c} \alpha_i \vec{h}_{u_i} \quad (7)$$

where α_i is the output attention coefficient.

3.3 Cascade Prediction and Loss Function

After embeddings from both heterogeneous bipartite and homogeneous cascade graphs are obtained, they are concatenated and fed into a MLP:

$$\hat{y}_i = MLP([\vec{h}_{c_i} || \vec{h}'_{c_i}]) \quad (8)$$

To optimize parameters of this deep learning model, the loss function is defined as the mean squared error:

$$L = \frac{\sum_i (y_i - \hat{y}_i)^2}{n} \quad (9)$$

Similar to [7], the label is defined as logarithm of incremental size, i.e., $y_i = \log(\Delta S_i + 1)$, where ΔS_i is the incremental size.

4 Evaluation

In this section, we evaluate the performance of our proposed model CollaborateCas by comparing it with several state-of-the-art approaches. Some variants of CollaborateCas are also considered for experimental study. We evaluate our model on two real-world datasets including Sina Weibo dataset [2] and HEP-PH dataset [8]. We adopt two commonly used metrics, i.e., MSE [4] (Mean Square Error) and RMSPE [7] (Root Mean Square Percentage Error).

4.1 Baselines

To show the superiority of our approach, we select 5 state-of-the-art approaches and 3 variants as baselines.

- **Feature-linear & Feature-Deep:** We feed some selected features into a linear regression model (Feature-linear) and a MLP (Feature-deep).
- **Node2Vec:** Node2Vec [10] learns node embeddings from cascade graphs.
- **DeepCas:** DeepCas [12] applies GRU neural network to sequences generated from cascade graph.
- **CasCN:** CasCN [4] combines graph convolutional network with LSTM.
- **Deepcon_str:** Deepcon_str [7] regards each cascade as a node and builds two cascade-level graphs.
- **CollaborateCas-bipartite:** CollaborateCas-bipartite removes the part of homogeneous cascade graphs.
- **CollaborateCas-cascade:** CollaborateCas-cascade removes bipartite graph.
- **CollaborateCas-mean:** The attention mechanism at the output of cascade graph is replaced with mean operation.

Table 1. Overall performance between different approaches on the Sina Weibo dataset.

T	1 h		2 h		3 h	
Metric	MSE	RMSPE	MSE	RMSPE	MSE	RMSPE
Deeplinear	1.5100	0.3112	1.6455	0.3960	1.7128	0.4979
Deepfeature	1.3116	0.3428	1.5293	0.4485	1.4847	0.5659
Node2Vec	2.1966	0.2500	2.2902	0.4625	2.2107	0.4891
DeepCas	1.0759	0.2229	1.3887	0.3924	1.3003	0.3868
CasCN	1.3336	0.2147	1.4956	0.4131	1.2786	0.4527
Deepcon_str	1.0709	0.2087	1.5049	0.3949	1.4794	0.3776
CollaborateCas	0.9149	0.2019	1.2603	0.3446	1.2374	0.3487

Table 2. Overall performance between different approaches on the HEP-PH dataset.

T	3 years		5 years		7 years	
Metric	MSE	RMSPE	MSE	RMSPE	MSE	RMSPE
Deeplinear	2.3738	0.5465	2.6249	0.6064	2.9796	0.6908
Deepfeature	2.3973	0.6134	2.2252	0.7486	2.6035	0.7773
Node2Vec	3.4308	0.6675	3.7664	0.8605	3.4933	0.8380
DeepCas	3.0613	0.6102	3.3759	0.9842	3.4412	1.1956
CasCN	2.5551	0.6544	2.1644	0.7142	2.3033	0.7311
Deepcon_str	2.7794	0.6993	2.5188	0.6890	2.7712	0.7880
CollaborateCas	2.3197	0.5351	2.1560	0.4953	1.9723	0.6849

4.2 Performance Comparison

The experimental results of our proposed model and various baselines are shown in Table 1 and Table 2. CollaborateCas achieves significantly lower MSE and RMSPE than all the baselines. For feature engineering-based methods, feature-linear and feature-deep show similar predictability on this task. Node2Vec and DeepCas have relative lower accuracy than other deep learning models.

CasCN performs worse than Deepcon_str and our model because it treats each cascade independently. Deepcon_str has overall better performance than other deep learning-based baselines. However, this method ignores detailed interactions between users and cascades. CollaborateCas has achieved better results than baselines in all three observation time windows, indicating that our unified modeling of heterogeneous bipartite graph and homogeneous cascade graphs can significantly improve the performance of cascade popularity prediction.

We also compare the performance of different variants of our model, as shown in Table 3. In general, CollaborateCas still performs better than other variants. The most competitive variant is CollaborateCas-bipartite, which means that the heterogeneous bipartite graph is an essential part for cascade prediction.

Table 3. Overall performance between variants of CollaborateCas.

T	1 h		2 h		3 h	
Metric	MSE	RMSPE	MSE	RMSPE	MSE	RMSPE
CollaborateCas-bipartite	0.9809	0.2083	1.2796	0.3851	1.2281	0.3601
CollaborateCas-cascade	1.7285	0.2634	2.2310	0.4485	2.2790	0.4649
CollaborateCas-mean	0.9982	0.1957	1.3773	0.3850	1.2532	0.3644
CollaborateCas	0.9149	0.2019	1.2603	0.3446	1.2374	0.3487

5 Conclusion

To address the cascade popularity problem, we proposed a novel deep learning model called CollaborateCas, which can capture collaborations among different cascades. To this end, we constructed a heterogeneous bipartite graph based on fine-grained user-message interactions and homogeneous cascade graphs incorporating temporal information as edge features. Experiments results demonstrate that CollaborateCas can achieve higher accuracy than state-of-the-art baselines.

Acknowledgements. This work was supported in part by: National Natural Science Foundation of China (Nos. 61966008, U2033213, 61804017).

References

1. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint [arXiv:1312.6203](https://arxiv.org/abs/1312.6203) (2013)

2. Cao, Q., Shen, H., Cen, K., Ouyang, W., Cheng, X.: Deephawkes: bridging the gap between prediction and understanding of information cascades. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1149–1158 (2017)
3. Cao, Q., Shen, H., Gao, J., Wei, B., Cheng, X.: Popularity prediction on social platforms with coupled graph neural networks. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 70–78 (2020)
4. Chen, X., Zhou, F., Zhang, K., Trajcevski, G., Zhong, T., Zhang, F.: Information diffusion prediction via recurrent cascades convolution. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 770–781. IEEE (2019)
5. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint [arXiv:1406.1078](https://arxiv.org/abs/1406.1078) (2014)
6. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Process. Syst.* **29**, 3844–3852 (2016)
7. Feng, X., Zhao, Q., Liu, Z.: Prediction of information cascades via content and structure proximity preserved graph level embedding. *Inf. Sci.* **560**, 424–440 (2021)
8. Gehrke J, Ginsparg P, K.J.: Overview of the 2003 KDD cup. In: *Acm Sigkdd Explor. Newslett.* **5**(2), 149–151 (2003)
9. Gong, Q., et al.: Cross-site prediction on social influence for cold-start users in online social networks. *ACM Trans. Web (TWEB)* **15**(2), 1–23 (2021)
10. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
11. Jiang, B., Lu, Z., Li, N., Wu, J., Yi, F., Han, D.: Retweeting prediction using matrix factorization with binomial distribution and contextual information. In: Li, G., Yang, J., Gama, J., Natwichai, J., Tong, Y. (eds.) DASFAA 2019. LNCS, vol. 11447, pp. 121–138. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-18579-4_8
12. Li, C., Ma, J., Guo, X., Mei, Q.: Deepcas: an end-to-end predictor of information cascades. In: Proceedings of the 26th International Conference on World Wide Web, pp. 577–586 (2017)
13. Myers, S.A., Leskovec, J.: Clash of the contagions: cooperation and competition in information diffusion. In: 2012 IEEE 12th International Conference on Data Mining, pp. 539–548. IEEE (2012)
14. Su, Y., Zhang, X., Wang, S., Fang, B., Zhang, T., Yu, P.S.: Understanding information diffusion via heterogeneous information network embeddings. In: Li, G., Yang, J., Gama, J., Natwichai, J., Tong, Y. (eds.) DASFAA 2019. LNCS, vol. 11446, pp. 501–516. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-18576-3_30
15. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
16. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. *Science* **359**(6380), 1146–1151 (2018)
17. Weng, L., Flammini, A., Vespignani, A., Menczer, F.: Competition among memes in a world with limited attention. *Sci. Rep.* **2**(1), 1–9 (2012)
18. Zhou, F., Xu, X., Trajcevski, G., Zhang, K.: A survey of information cascade analysis: models, predictions, and recent advances. *ACM Comput. Surv. (CSUR)* **54**(2), 1–36 (2021)