UserRBPM: User Retweet Behavior Prediction with Graph Representation Learning

Huihui Guo, Li Yang(✉), and Zeyu Liu

Xidian University, Xi'an 710071, China
guohuihui@stu.xidian.edu.cn, yangli@xidian.edu.cn

Abstract. Social and information networks such as Facebook, Twitter, and Weibo have become the main social platforms for the public to share and exchange information, where we can easily access friends' activities and are in turn be influenced by them. Consequently, the analysis and modeling of user retweet behavior prediction have important application value in such aspects as information dissemination, public opinion monitoring, and product recommendation. Most of the existing solutions for user retweeting behavior prediction are usually based on network topology maps of information dissemination or design various hand-crafted rules to extract user-specific and network-specific features. However, these methods are very complex or heavily dependent on the knowledge of domain experts. Inspired by the successful use of neural networks in representation learning, we design a framework UserRBPM to explore potential driving factors and predictable signals in user retweet behavior. We use the graph embedding technology to extract the structural attributes of the ego-network, consider the drivers of social influence from the spatial and temporal levels, and use the graph convolutional network and the graph attention mechanism to learn its potential social representation and predictive signals. Experimental results show that our proposed UserRBPM framework can significantly improve prediction performance and express social influence better than traditional feature engineering-based approaches.

Keywords: Social networks · Retweet behavior prediction · Graph convolution · Graph attention · Representation learning

1 Introduction

Due to their convenient capability to share real-time information, social media sites (e.g., Weibo, Facebook, and Twitter) have grown rapidly in recent years. They have become the main platforms for the public to share and exchange information, and to a great extent meet the social needs of users. Under normal

Supported by the National Natural Science Foundation of China (62072359, 62072352), and the National Key Research and Development Project (2017YFB0801805).

J. Xiong et al. (Eds.): MobiMedia 2021, LNICST 394, pp. 613–632, 2021.
https://doi.org/10.1007/978-3-030-89814-4_45

circumstances, online social networks will record a large amount of information generated by people through interactive activities, including various user behavior data. User behaviors (also called actions) in online social networks contain posting messages, purchasing products, retweeting information, and establishing friendships, etc. By analyzing the distribution and causality of these behaviors, we can evaluate the influence between the initiator and the communicator of the behavior, as well, we can predict people's behaviors on social networks and deepen our understanding and understanding of human social behavior [18]. Till now, there is little doubt that the large amount of data generated by users' interaction provides an opportunity to study user behavior patterns, and the analysis and modeling of retweet behavior prediction have become a research hotspot. In addition to analyzing the retweeting behavior itself, retweeting can also help with a variety of tasks such as information spreading prediction [30], popularity prediction [38], and precision marketing [2].

Previous researches investigated the problem of user retweet behavior prediction from different points of view. On the first approach, some researchers build retweet behavior prediction models through network topology maps of information dissemination. Matsubara et al. [15] studied the dynamics of information diffusion in social media by extending an analysis model for information dissemination from the classic 'Susceptible-Infected' (SI) model. Wang et al. [26] proposed an improved SIR model, which used the mean field theory to study the dynamic behavior in uniform and heterogeneous network models. Their experiment showed that the existence of the network would influence information communication. This kind of research method studied retweeting behavior by modeling the propagation path of the message from a global perspective. The other approach is the machine learning method based on feature engineering. Liu et al. [13] proposed a retweeting behavior prediction model based on fuzzy theory and neural network algorithm, which can effectively predict the user's retweeting behavior and dynamically perceive the changes in hotspot topics. This research method relies on the knowledge of domain experts, and the process of feature selection may take a long time. However, in many online applications such as personalized recommendation [29,31] and advertising [2], it is critical to effectively analyze the social influence of each individual and further predict the retweeting behavior of users.

In this paper, we focus on user-level social influence. We aim to predict the action statuses of the target user according to the action statuses of her near neighbors and her local structural information. For example, in social networks, a person's behavior will be affected by her neighbors. As shown in Fig. 1, for the central user u, if some friends (red node) around her posted a microblog and other friends (white node) did not post it, whether the action statuses of user u will be affected by the surrounding friends and forward this tweet can be regarded as a user retweeting behavior prediction problem. The social influence hidden behind the retweeting behavior not only depends on the number of active users, but may also be related to the local network structure formed by "active" users. The problem mentioned above are common in practical applications, such as presidential elections [3], innovation adoption [21], and e-commerce [11]. Therefore, it has inspired many research work on user-level influence models, most of which [9,34] consider complicated handcrafted features, which require extensive knowledge of specific domains.

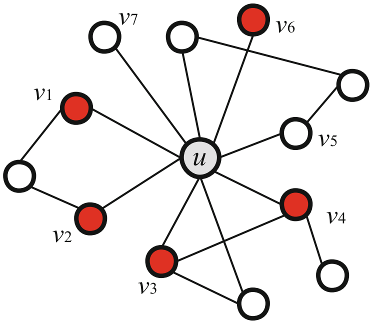

Fig. 1. A motivating example of user retweet behavior prediction. (Color figure online)

The recently proposed graph convolution networks(GCN) [4,12] is currently the best choice for graph data learning tasks. Inspired by the successful application of neural networks in representation learning [20,33], we designed an end-to-end framework UserRBPM to explore potential driving factors and predictive signals in user retweeting behaviors. We expect deep learning frameworks to have better expressive capability and prediction performance. The designed solution is to represent both influence driving factors and network structures into a latent space, and then use graph neural networks to effectively extract spatial features for learning, and further construct a user retweet behavior prediction model. We demonstrate the effectiveness and efficiency of our proposed framework on Weibo social networks. We compare UserRBPM with several conventional methods, and experiment results show that the UserRBPM framework can significantly improve the prediction performance.

The main contributions of this work can be summarized as follows:

- We designed an end-to-end learning framework UserRBPM to explore potential driving factors and predictive signals in user retweeting behaviors.
- We convert the retweeting behavior prediction into a binary graph classification, which is more operable and comprehensible.
- Experiment results demonstrate that the UserRBPM framework can achieve better prediction performance than existing methods.

Organization. The rest of this paper is organized as follows. Section 2 summarizes related work. Section 3 formulates the user retweet behavior prediction problem. We detail the proposed framework in Sect. 4. In Sect. 5 and Sect. 6, we conduct extensive experiments and analyze the results. Finally, we conclude our work in Sect. 7.

2 Related Work

In this section, we categorize and summarize prior work on user retweet behavior prediction and graph representation learning.

2.1 User Retweet Behavior Prediction

Many studies on user retweet behavior in social networks are based on the analysis and modeling of the dynamics in the process of information dissemination. Currently, researches on user behavior prediction in social networks take primarily two approaches. On the first approach, Yuan et al. [32] investigated the dynamics of friend relationships through online social interaction, and thus proposed a model to predict repliers or retweeters according to a particular tweet posted at a certain time in online social networks. Tang et al. [22] studied the conformity phenomenon of user behavior in social networks, and proposed a probabilistic model called Confluence to predict user behavior. This model can distinguish and quantify the effects of the different types of conformity. Zhang et al. [37] proposed three metrics: user enthusiasm, user engine, and user duration, to describe the user retweet behavior in the message spreading process, and studied the relationship between these three metrics and the influence obtained by the user retweet behavior.

The other approach is the machine learning method based on feature engineering, which solved the problem of user behavior analysis and prediction by manually formulating rules to extract the basic features of users and network structural features. Luo et al. [14] explored features: followers status, retweet history, followers interests, and followers active time with a learning-to-rank framework to discover who would retweet a tweet poster on Twitter. Zhang et al. [34] analyzed the influence of the number of active neighbors of a user on retweeting behavior, proposed two instantiation functions based on structural diversity and pairwise influence, and applied a classifier based on logistic regression to predict users' retweet behaviors. Jiang et al. [9] pointed out that the retweeting prediction is a sing-type setting problem. By analyzing the basic influence factors of retweet behavior in Weibo, the sing-type collaborative filtering method is used to measure users' personal preference and social influence for the purpose of predicting retweet behavior.

2.2 Graph Representation Learning

Graph representation learning has emerged as a powerful technique for solving real-world problems. Various downstream graph learning tasks have benefit from its recent developments, such as node classification [7], similarity search [35], and graph classification [19,36]. The primary challenge in this field is to find a way to represent or encode the structure of graphs so that it can be easily exploited by machine learning models. Traditional machine learning approaches relied on user-defined heuristics to extract features encoding structural information about a graph (e.g., degree statistics or kernel functions). However, recent years, have seen a surge in approaches for automatically learning to encode graph structure into low-dimensional embedding using techniques based on deep learning and nonlinear dimension reduction. Chen et al. [5] exploited graph attention networks(GAT) to learn user node representation by spreading information in heterogeneous graphs, and then leveraged limited labels of users to build end-to-end semi-supervised user profiling predictor. Zhang et al. [33] introduced the problem of heterogeneous graph representation learning and proposed a heterogeneous graph neural networks model HetGNN. Extensive experiments on various graph mining tasks, i.e., link prediction, recommendation, and node classification, demonstrated that HetGNN can outperform state-of-the-art methods.

3 Problem Formulation

In this section, we introduce necessary definitions and then formulate the problem of user retweet behavior prediction.

Definition 1. Ego network
The ego network model is one of the important tools for studying human social behavior and social networks. Compared with the global network version, the research version of the ego network pays more attention to individual users, and it is in line with the needs of personalized services in actual application systems. The research version of this paper can also be extended to other scenarios that include network relationships.

***r*-neighbors** Let $G = (V, E)$ denote a social network, where V is a set of users nodes and $E \subseteq V \times V$ is a set of relationships between users. We use $v_i \in V$ to represent a user and $e_{ij} \in E$ to represent a relationship between v_i and v_j. In this work, we consider undirected relationships. For a user u, its r-neighbors nodes are defined as $\Gamma_u^r = \{v : d(u,v) \le r\}$, where $d(u,v)$ is the shortest path distance (in terms of the number of hops) between u and v in the network G, $r \ge 1$ is a tunable integer parameter to control the scale of the ego network.

***r*-ego network** The r-ego network of user u is the subnetwork induced by Γ_u^r, denoted by G_u^r.

Definition 2. Social action
In sociology, social action is an act which takes into account the actions and reactions of individuals. Users in social networks perform social actions, such as retweeting behaviors, citation behaviors. At each time stamp t, we observe a binary action status of user u, $s_u^t \in \{0, 1\}$, where $s_u^t = 1$ indicates user u has performed this action before or on the timestamp t, and $s_u^t = 0$ indicates that the user has not performed this action yet.

In this paper, our research motivation of user retweeting behavior prediction problem can be vividly illustrated by Fig. 1. For a user u in her 2-ego network (i.e., r = 2), if some users retweet a microblog m before or on the timestamp t, they are considered to be active. We can observe the action statuses of u's neighbors, such as $s_{v_1}^t = 1$, $s_{v_2}^t = 1$, and $s_{v_5}^t = 0$. Moreover, the set of active neighbors of user u is represented by $\psi_u^t = \{v_1, v_2, v_3, v_4, v_6\}$. As shown in Fig. 1, we study whether the action statuses of user u will be influenced by the surrounding friends and forward this microblog. Next, we will formalize the problem of user retweet behavior prediction.

Problem 1. User retweet behavior prediction [34]
User retweet behavior prediction models the probability of $u's$ action states conditioned on her r-ego network and the action states of her r-neighbors. More formally, given G_u^r and $S_u^t = \{s_v^t : v \in \Gamma_u^r \setminus \{u\}\}$, it can be concluded that the user retweet behavior prediction formula of user u after a given time interval Δt is as follows:

$$A_v = P\left(s_u^{t+\Delta t} \mid G_u^r, S_u^t\right) \tag{1}$$

Practically, A_v denotes the predicted social action status of user u. Suppose we have N instances, and each instance is a 3-tuple (u, a, t), where u is a user, a is a social action and t is a timestamp. For such a 3-tuple (u, a, t), we also know $u's$ r-ego network G_u^r, the action states of $u's$ r-neighbors S_u^t, and $u's$ future action states at $t + \Delta t$, i.e., $s_u^{t+\Delta t}$. We then formulate user retweet behavior prediction as a binary graph classification problem which can be solved by minimizing the following negative log likelihood objective w.r.t model parameter θ:

$$L(\theta) = -\sum_{i=1}^{n} \log\left(P_\theta\left(s_u^{t+\Delta t} \mid G_u^r, S_u^t\right)\right) \tag{2}$$

4 Model Framework

In this paper, we formally propose the UserRBPM to address the user retweet behavior prediction problem. The framework is based on graph neural networks to parameterize the probability in Eq. (2) and automatically detect the potential driving factors and predictive signals of user retweet behavior prediction. As shown in Fig. 2, UserRBPM is consisted of pre-trained network embedding layer, input layer, GCN/GAT layer and output layer.

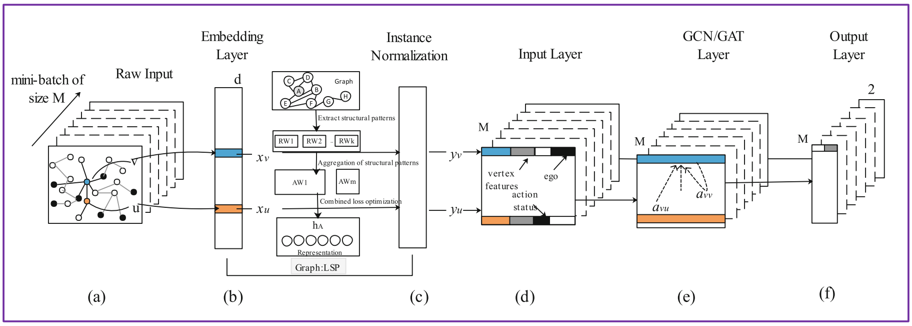

Fig. 2. Our proposed framework of UserRBPM (User Retweet Behavior Prediction Model).

4.1 Sampling Near Neighbors

Given a user u, the most straightforward way to extract its r-ego network is to perform a Breadth-First-Search (BFS) starting from user u. However, for different users, r-ego network scale (regarding the number of nodes) may vary greatly. Meanwhile, the size of user $u's$ r-ego network can be very large due to small-world property in social networks [27]. In real-world application scenarios, when sampling neighbor nodes of an ego user node, the problem that may arise is that each node has a different number of neighbors. Specifically, due to the small-world phenomenon in social networks, the size of user $u's$ r-ego network may be relatively very large or small. In addition, these different sizes of data are not suitable for most deep learning models.

In order to address the above problem, we select to perform random walk with restart (RWR) [23] from the original r-ego network to fix the sample size. Inspired by [1,24] which suggest that people are more susceptible to be influenced by active neighbors than inactive ones, we start a random walk on G_u^r from user u or its active neighbors. The walk iteratively travels to its neighborhood with a probability proportional to the weight of each edge. In addition, the walk returns back to the starting vertex u with a positive probability at each step. In this way, a fixed size of number of vertices can be collected, denoted by $\vec{\Gamma}_u^r$ with $\left|\vec{\Gamma}_u^r\right| = n$. We then regard the sub-network $\vec{G}_u^r$ induced by $\vec{\Gamma}_u^r$ as a proxy of the r-ego network G_u^r, and denote $\vec{S}_u^t = \{s_v^t : v \in \vec{\Gamma}_u^r \setminus \{u\}\}$ to be the action statuses of $u's$ sampled neighbors. When we use RWR, the starting node can be ego user or its active neighbors. The purpose of setting as described above is to make the starting node in the sequence obtained by walking as much as possible to keep in touch with surrounding neighbors, instead of being relatively single, so as to support the purpose of people being more susceptible to be influenced of active neighbors.

4.2 Graph Neural Networks Model

We design an effective graph neural networks model to incorporate both the structural properties in $\vec{G}_u^r$ and action statuses in $\vec{S}_u^t$, learned a hidden embedding vector for each ego user, then used to predict the action statuses of the ego user in the next time period $s_v^{t+\Delta t}$. As shown in Fig. 2, the graph neural networks model includes embedding layer, instance normalization layer, input layer, graph neural networks layer, and output layer.

Embedding Layer. For graph structure data such as social networks, we want to learn the users' social representation from users' relationship network data, that is, our main purpose is to discover network structural property and encode them into low-dimensional latent space. More formally, network embedding learns an embedding matrix $X \in R^{d\times|V|}$, with each column corresponding to the representation of a vertex (user) in the network G. In our scheme, we learn a low-dimensional dense real number vector $x_v \in R^d$ for each node v in the network, where $d \ll N$. The process of network representation learning can be unsupervised or semi-supervised.

In social networks, when considering the structural information, we can take the triadic closure, patterns characteristic of strong ties in social networks. As shown in Fig. 3, there will be such a case: The figure on the left contains a triadic closure. For the green node, it is equivalent to a different tree structure on the right (there is no triadic closure) after two neighborhood aggregations, which ignore the structural information of triadic closure. Therefore, there is a need for a method of graph representation learning that can adapt to different local structures.

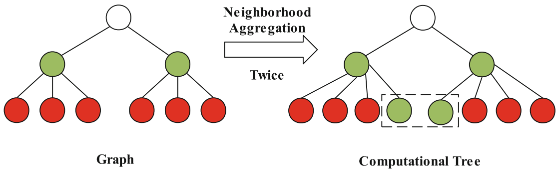

Fig. 3. Computational tree of a triadic closure graph. (Color figure online)

In our work, we utilize the GraLSP model [10] for graph representation learning, which explicitly incorporates local structural patterns into the neighborhood aggregation through random anonymous walks. Specifically, the framework captures the local structural patterns via random anonymous walks, and then these walk sequences are fed into the feature aggregation, where various mechanisms are designed to address the impact of structural features, including adaptive receptive radius, attention, and amplification. In addition, GraLSP can capture similarities between structures and are optimized jointly with near objectives of node. The process of GraLSP model for graph representation learning is shown in Fig. 2(b). In the case of making full use of the structural model, the GraLSP can outperform competitors in various prediction tasks in multiple datasets.

Instance Normalization Layer. In the training process of UserRBPM model, we applied Instance Normalization (IN) [25] to prevent overfitting, which is a regularization technique that loosens the model and allows for greater generalization. And for such tasks that focus on each sample, the information from each sample is very important. Therefore, we adopt such a technique in the task of retweet behavior prediction. After original data is normalized, the indicators are between [0, 1], which is suitable for comprehensive comparative analysis. Furthermore, it helps to speed up learning and also reduces overfitting.

Input Layer. As claimed in Fig. 2(d), the input layer constructs a feature vector for each user. The feature vector considered in our work consists of three parts: 1) the normalized low-dimensional embedding comes from the up-stream instance normalization layer; 2) two binary variables are also considered. The first variable represents the user's action statuses, and the second variable represents whether the user is an ego user; 3) the input layer also includes other personalized vertex features, such as spatial-level features (e.g., social roles) and temporal-level features (e.g., similarity, exposure, retweet rate.)

GCN Layer. The recently developed GCN [6] is a successful attempt to generalize the convolutional neural networks used in Euclidean space to graph structure data modeling. The GCN model naturally integrates the connection mode and feature attributes of graph structure data, and it is much better than many state-of-the-art methods on benchmarks. Graph Convolutional Network (GCN) is a semi-supervised learning algorithm for graph structure data, which can effectively extract spatial features for machine learning on such a network topology. Simultaneously, it can perform end-to-end learning of node feature information and structure information, which is one of the best choices for graph data learning tasks at present.

Graph Attention Networks. Essentially, both GCN and GAT are aggregation operations that aggregate the characteristics of neighbor nodes into the central node. GCN uses the Laplacian matrix to perform graph convolution operations, while GAT introduces the attention mechanism into GCN, which can add weight to the influence of neighboring nodes, thereby differentiating the influence of neighboring nodes. GAT assigns different weights to each node, paying attention to those nodes with greater effects, while ignoring some nodes with smaller effects. To a certain extent, the performance ability of GAT will be stronger, because the correlation between node features will be better integrated into the model.

Output Layer. In the output layer, each node corresponds to a two-dimensional representation, which is used to represent the user's behavior state (retweet/unretweet, cite/uncite, etc.). By comparing the representation of the ego user with groud truth, we then optimize the log-likelihood loss.

5 Experiment Setup

In this subsection, we first introduce the construction process and statistical characteristics of the dataset. Then, we present the existing representative methods and evaluation metrics. Finally, we introduce the implementation details of the UserRBPM framework.

5.1 Dataset Presentation and Processing

Presentation to Raw Datasets. We use real-world datasets to quantitatively and qualitatively evaluate the proposed UserRBPM framework. We used the Weibo dataset in the work of Tang et al. [17,34] and Wu et al. [28] also used the Weibo dataset in the work, and then we performed data preprocessing according to our research question. The microblogging network used in our research work is to crawl data from Sina Weibo. Particularly, when user u_1 follows user u_2, u_2's activities (such as tweet and retweet) will be visible to u_1. User u_1 can choose to tweet or retweet by user u_2. User u_1 is called the follower of user u_2 and user u_2 is called the followee of user u_1.

The Generation and Processing of Samples. In the problem of retweeting behavior prediction, since we can directly learn from the microblogs record which users have retweeted the microblogs, the extraction process of positive samples is relatively simple. Thus, for a user v who is affected by others, he performs a social action a at a certain timestamp t, and then we generate a positive sample. Compared with the extraction of positive samples, it is impossible to directly know from the microblogs records which users saw the message but did not retweet the microblogs. Therefore, the extraction method of negative samples is much more complicated.

For our research scenarios, there are two data imbalance problems. The first one comes from the number of active neighbors. As Zhang et al. [34] observed, structural features are significantly related to user retweeting behavior when the ego user has a relatively large number of active neighbors. For example, in the Weibo dataset, 80% of users have only one active neighbor and users with more than 3 active neighbors account for only 8.57%. Therefore, the model will be controlled by observation samples with few active neighbors. To illustrate the superiority of our proposed model in capturing local structural information, we established a balanced sub-dataset *Edata* (as shown in Table 1.) for fair data analysis and further training-test scheme. Specifically, we filter out samples in which the followers or followees did not have Weibo content. In addition, we only considered samples in which ego users have at least 3 active neighbors.

Table 1. Statistics of sub-dataset *Edata*.

Edataset	#Users	#Follow-relationships	#Oroginal-microblogs	#Retweet	#Ego Users
Weibo	1,500,290	20,297,550	274,150	15,755,810	151,300

The second problem is imbalanced labels. For instance, in our Weibo data set, the ratio between positive instances and negative instances is about 1:300. To address this problem, the most direct way is to select a relatively balanced dataset, that is, set the ratio of positive samples and negative samples to 1:3. In addition, we also used the global random downsampling method and microblog granularity-based down-sampling method to process imbalanced datasets. Among them, when we use the global random down-sampling method, the number of microblogs involved in the negative samples in the obtained dataset is small, and there is a case where only positive samples of the same microblog are not sampled to their corresponding negative samples. The down-sampling method based on microblog granularity can try its best to ensure that the number of positive and negative samples of the same microblog is also the same.

The Features of Our Design. We made detailed data observation and analyzed how the characteristics of users at the spatial and temporal levels influence retweeting behavior in addition to the structural attributes of social networks. To visualize the observation results, we design several statistical information, which respectively represent spatial-level features and temporal-level features. These characteristics can be regarded as user node features. In our work, the spatial-level features are specifically analyzed in terms of social roles. We studied the influence of social roles played by different users on the prediction performance of retweeting behavior. Inspired by the previous research work of Wu et al. [28], we divide users into three groups according to their network attributes: opinion leaders (OpnLdr), structural hole spanners (StrHole)) and ordinary users (OrdUsr). A detailed analysis of users' social roles and behaviors is shown in Table 2. For the temporal-level feature, we mainly analyzed the content of the messages posted by users. The features of our design are shown in Table 3.

Table 2. The statistics of social roles and relation statuses.

Social role	OrdUsr	OpnLdr	StrHole	Sum
Retweet Behavior	6,617,440(42%)	3,623,836(23%)	5,514,534(35%)	15,755,810
Original Post	68,537(25%)	123,367(45%)	82,245(30%)	274,150
Sum	1,125,217(75%)	150,029(10%)	225,043(15%)	1,500,290

Table 3. List of features used in our work.

Spatial-level features	Social role_Opinion leader (OpnLdr)
	Social role_Structure hole (StrHole)
	Social role_Ordinary users (OrdUsr)
Temporal-level features	The TF-IDF similarity between ego user and its followees' post content (Similarity)
	The number of microblogs posted by the followees (Exposure)
	The retweet rate of ego users to their followees (Retweet rate)
Handcrafted ego-network features [22]	The number/ratio of active neighbors
	Density of subnetwork induced by active neighbors
	Connected of components formed by active neighbors

5.2 Comparison Methods

In order to verify the effectiveness of our proposed framework, we compared the prediction performance of UserRBPM in this paper with existing representative methods. Firstly, we compared UserRBPM with previous retweeting behavior prediction methods which usually extract rule-based features. Secondly, by comparing the GraLSP method with other network embedding methods, it is verified that the local structure information plays a more important role in the prediction of forwarding behavior than the global information. The comparison method is as follows:

- Hand-crafted features + Logistic Regression(LR): We use logistic to train the classification model. The features we constructed manually include two categories: one is the user node features designed in our work, including

spatial-level and temporal-level features; the other is the ego network features designed by Qiu et al. [17].
- Hand-crafted features + Support Vector Machine(SVM): We also use SVM as the classification model. The model use the same features as Logistic Regression.
- DeepWalk: DeepWalk [16] is a network embedding method that learns a social representation of a network by truncated random walks to obtain the structural information of each vertex.
- Node2vec: Node2vec [8] designs a biased random walks that can trade off between homophily and structural equivalence of the network.
- Our Proposed Method: In our proposed UserRBPM framework, we use GraLSP to extract the structural attributes of the r-ego network, design the user node features at the spatial-level and temporal-level, and finally apply GCN and GAT to learn latent predictive signals.

In order to quantitatively evaluate our proposed framework, we use the four popular metrics to evaluate the performance of retweeting behavior prediction. Specifically, we evaluate the performance of the UserRBPM in terms of Area Under Curve (AUC), Precision, Recall, and F1-Score.

5.3 Implementation Details

There are two stages for training our UserRBPM framework. In the first stage, we pretrain each module of UserRBPM, and in the second stage, we integrate the three modules of UserRBPM for fine-tuning.

Stage I: Pretrain of Each Module. For our framework, UserRBPM, we first perform a random walk with a restart probability of 0.8 and set the size of the sampled sub-network to be 30. For the embedding layer, the embedding dimension of the GraLSP model is set to three dimensions of 32, 64, and 128, and train GraLSP for 1000 epochs. Then we choose to use a three-layer GCN or GAT network structure, the first and second GCN/GAT layers both contain 128 hidden units, while the third layer (output layer) contains 2 hidden units for binary prediction. In particular, for UserRBPM with multi-head graph attention, both the first and second layers consist of K = 8 attention heads, and each attention head computes 16 hidden units (total $8 \times 16 = 128$ hidden units). The network is optimized by the Adam optimizer with the learning rate of 0.1, weight decay 5e-4, and dropout rate of 0.2. To evaluate the model performance and prevent information leakage, we performed five-fold cross-validation on our datasets. Specifically, we select 75% instances for training, 12.5% instances for validation, and 12.5% instances for test. In addition, the mini-batch size is set to be 1024 in our experiments.

Stage II: Global Fine-Tuning. In the global fine-tuning stage, if the dimension of embedding layer is set too large, the training process will be too slow, while a small setting will affect the performance of our model. After fine-tuning the model, we found that the model performance is relatively stable when the embedding dimension is set to 64. Then, we fix the parameters of pre-trained embedding module, and train the GCN/GAT layer with Adam optimizer for 1000 epochs, with the learning rate of 0.001. As the larger learning rate can make the model learn faster, thereby accelerating the convergence speed, but the performance of the model will be affected to some extent. Therefore, we set a relatively large learning rate at the beginning, and then gradually decrease with the training. Finally, we choose the best model by stopping using the loss on the validation sets as early as possible.

6 Experiment Results

6.1 Prediction Performance Analysis

Overall Performance Analysis. To verify the influence of the structural attributes of users' ego network and user nodes characteristics (extracted from the spatial and temporal level) on the prediction performance in social networks, as well as the interaction between features at different levels, we made the following comparison. As shown in Table 4, showing the prediction performance of different models.

Table 4. Prediction performance of different methods for retweeting behavior (%).

Methods	Precision	Recall	F1-score	AUC
Spatial-& Temporal-level& Handcrafted features + LR (ST& HC+LR)	69.74	71.58	70.65	77.27
Spatial-& Temporal-level& Handcrafted features + SVM (ST& HC+LR)	68.38	69.15	68.76	78.01
DeepWalk+ST+GAT	78.21	78.46	78.28	82.81
DeepWalk+ST& HC+GAT	79.68	80.24	79.96	82.75
Node2vec+ST+GAT	78.54	81.50	79.99	82.53
Node2vec+ST& HC+GAT	79.88	81.25	80.55	82.96
Our Method(UserRBPM)	81.97	82.58	82.27	83.21

Based on the analysis of four evaluation metrics used in our work, the performance of UserRBPM is better than the above-mentioned benchmark methods, which demonstrate that the effectiveness of our proposed framework. From the comparison among DeepWalk+ST&HC+GAT, Node2vec+ST&HC+GAT, and UserRBPM, we can observe that the GraLSP model we leverage in the embedding layer can indeed capture local structural patterns and significantly outperforms the first two methods in the experiment, confirming the GraLSP can indeed capture local structural patterns in retweeting behavior prediction. Meanwhile, from the comparison among ST&HC+LR, ST&HC+SVM, and UserRBPM, we notice that UserRBPM achieve an improvement of 13.59% in terms of precision. Such improvement verifies that the end-to-end learning framework UserRBPM can effectively detect potential driving factors and predictive signals in retweeting behavior prediction.

Comparing the first four methods with UserRBPM, it can be shown that the model which taking hand-crafted features as input hardly represent interaction effects, while network embedding technology can effectively extract high-dimensional structural attributes. Figure 4 shows that ST&HC+LR is notably better than HC+LR for retweeting behavior prediction. It reveals that users' spatial-level features and temporal-level features are the potential driving factors of retweeting behavior in social networks. Additionally, we observe that ST&HC+LR performs 4.42% better than HC+LR in terms of precision, verifying that the spatial-level and temporal-level features we designed have improved the prediction performance to a certain extent.

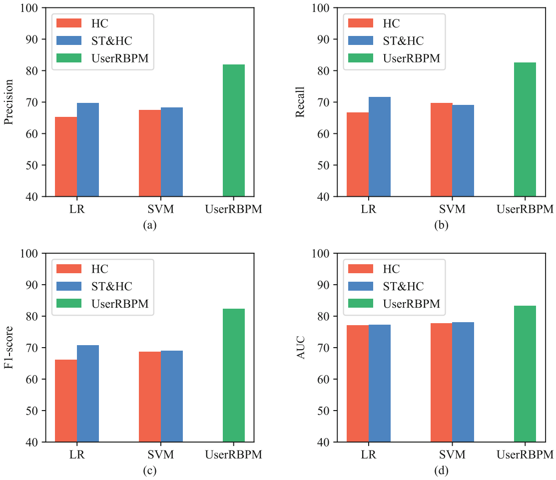

Fig. 4. Analysis results of different features.

Prediction Performance of Different Sampling Strategies. We use three sampling methods to obtain different training models. Among them, the directly sampling method (DSM) represents that we directly extract relatively balanced samples based on the ratio of the original positive and negative samples, that is, the ratio between positive and negative samples is set to 1:3. The number of positive samples and negative samples in completely random down-sampling (CRDM) and our down-sampling method (ODM) is the same. Experiment results are illustrated in Fig. 5. Compare to the completely random down-sampling method, the model trained with the samples obtained by our down-sampling method has better prediction performance. The better the prediction effect of the model obtained by the training data training, the more it shows that the dataset has universal significance and the learned model has a stronger generalization ability. In the original imbalanced datasets, the direct extraction of positive and negative samples with a ratio of 1:3 is simple, but the difference in the number of microblogs covered by the positive and negative samples is ignored. Therefore, the down-sampling method based on microblog granularity is more suitable for the user retweeting behavior prediction problem that we researched.

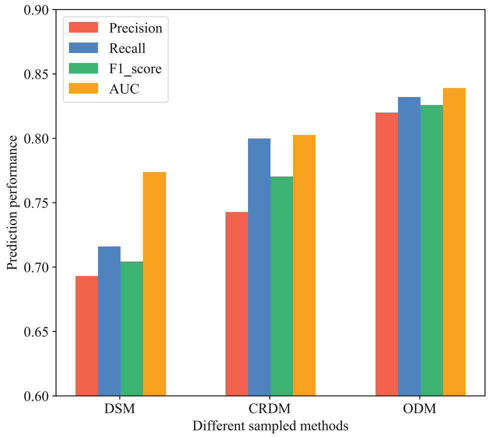

Fig. 5. Prediction performance of different sampling strategies.

Comparative Analysis of GCN and GAT. Table 5 is the prediction performance of two variants of graph deep learning, that is, the experimental results of using GCN and GAT to build models, respectively. In the application scenarios of our work, we observe that the performance of GCN in models constructed by different graph embedding technologies is generally worse than that of GAT. We attribute its disadvantage to the homophily assumption of GCN. This homophily exists in many real networks, but in our research scenario, different neighbor nodes may have different importance. Therefore, GAT is introduced to assign different weights to different neighboring nodes.

Table 5. Prediction performance of variants of UserRBPM (%).

Methods	Precision	Recall	F1-score	AUC
DeepWalk+ST& HC+GCN	77.49	79.28	78.37	74.89
DeepWalk+ST& HC+GAT	79.68	80.24	79.96	82.75
Node2vec+ST& HC+GCN	78.56	80.07	79.31	79.64
Node2vec+ST& HC+GAT	79.88	81.25	80.55	82.96
UserRBPM_GCN	80.82	80.58	80.70	82.23
UserRBPM_GAT	81.97	82.58	82.27	83.21

Besides, we wanted to avoid using hand-crafted features and make UserRBPM a pure end-to-end learning framework, so we compared the prediction performance with additional vertex features and no additional vertex features. Comparison results of prediction performance with/without vertex features, we observed that UserRBPM_GAT with hand-c rafted vertex features outperforms UserRBPM_GAT without hand-crafted vertex features by 1.48% in terms of precision, 0.81% in terms of recall, 1.15% in terms of F1-score, and 1.29% in terms of AUC. Experiment results demonstrate that, in addition to the pre-trained network embedding, we can still obtain comparable performance even without considering hand-crafted features.

6.2 Parameter Sensitivity Analysis

In addition, we consider parameter sensitivity in our work. We analyzed several hyper-parameters in the model and tested how different hyper-parameter choices affect the prediction performance.

Robustness Analysis. To verify the robustness of the UserRBPM framework, we changed the proportion of training set, validation set and test set and then redo the experiments. The results in Fig. 6 show that the model is effective under limited training data size. Even with small size of training set (20%–40%), our model can still have an acceptable and steady performance.

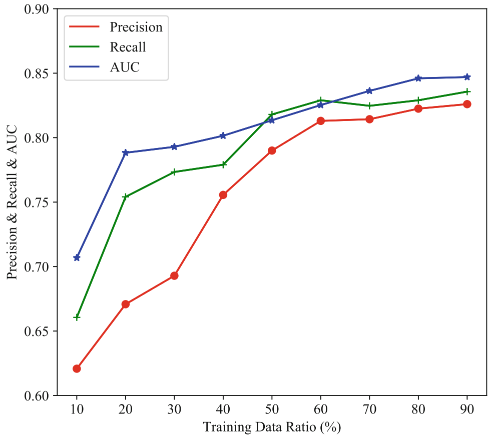

Fig. 6. Prediction performance with different training and test data size.

Effect of Instance Normalization. As mentioned in Sect. 4, this paper studied the technique used to accelerate model learning called *Instance Normalization (IN)*. This technique provides benefits to improve the classification performance. For instance, it can learn faster while maintaining or even increasing accuracy. Moreover, it also partially serves as a parameter tuning method. Therefore, we applied IN and obtained a boost in both performance and generalization. Figure 7 shows that the changes in the training loss of UserRBPM-GAT with/without IN layer during training. We can see that when there is an instance normalization layer, as the number of epochs increases, the training loss first drops rapidly and then remain stable. Instance normalization significantly avoids overfitting and makes the training process more stable.

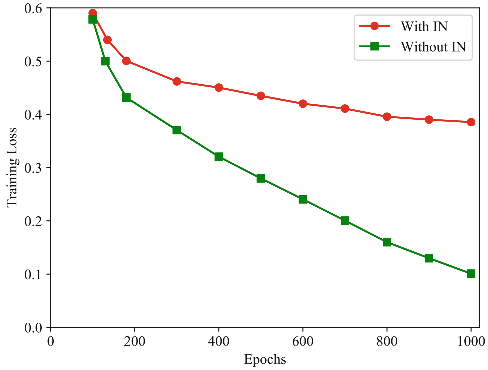

Fig. 7. The effect of instance normalization.

However, as shown in Fig. 8, we observe that the model without IN layer takes about 1011 s per epoch during the training process, while the model with IN layer takes about 1892 s per epoch. It was calculated that the model with IN layer increased the training time for each epoch by about 87% compared to the model without IN layer. Yet, we believe it is worthwhile to apply IN, as the additional training time is compensated with a faster learning rate (it requires less number of epochs to reach the same level of precision) and can ultimately achieve higher testing precision.

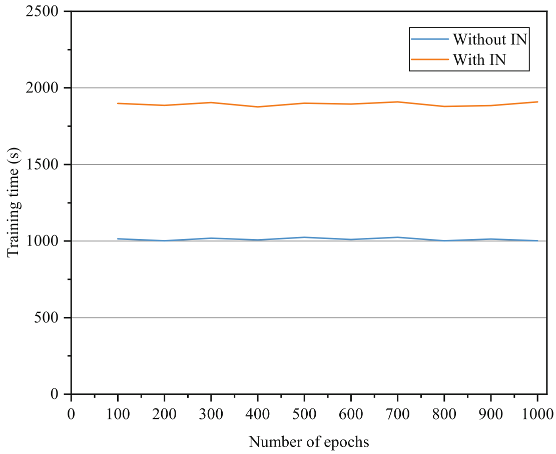

Fig. 8. Time overhead during each epoch.

7 Conclusion

In this work, we focus on user-level social influence in social networks and formulate the user retweet behavior prediction problem from a deep learning perspective. Unlike previous work that built a prediction model of retweet behavior based on network topology maps of information dissemination or conventional feature engineering-based approaches, we proposed UserRBPM framework to predict the action status of a user given the action statuses of her near neighbors and her local structural information. Experiments on a large-scale real-world dataset have shown that the UserRBPM significantly outperforms baselines with hand-crafted features in user retweet behavior prediction. This work explores the potential driving factors and predictable signals in user retweet behavior in hope that the deep learning framework has better expressive ability and prediction performance.

For future researches, the experimental dataset related to this research field still contains rich social dynamics that deserve further exploring. We can study user behavior in a semi-supervised manner, develop a generic solution based on heterogeneous graph learning, and then extend it to many network mining tasks, such as link prediction, social recommendation, similarity search, etc. Through such a learning scheme, we can leverage both unsupervised information and limited labels of users to build the predictor, and verify the effectiveness and rationality of user behavior analysis on real-world datasets.

References

1. Backstrom, L., Huttenlocher, D., Kleinberg, J., Lan, X.: Group formation in large social networks: membership, growth, and evolution. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 44–54 (2006)
2. Bakshy, E., Eckles, D., Yan, R., Rosenn, I.: Social influence in social advertising: evidence from field experiments. In: Proceedings of the 13th ACM Conference on Electronic Commerce, pp. 146–161 (2012)
3. Bond, R.M., et al.: A 61-million-person experiment in social influence and political mobilization. Nature **489**(7415), 295–298 (2012)
4. Chen, M., Wei, Z., Huang, Z., Ding, B., Li, Y.: Simple and deep graph convolutional networks. In: International Conference on Machine Learning, pp. 1725–1735. PMLR (2020)
5. Chen, W., et al.: Semi-supervised user profiling with heterogeneous graph attention networks. In: IJCAI. vol. 19, pp. 2116–2122 (2019)
6. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. arXiv preprint arXiv:1606.09375 (2016)
7. Donnat, C., Zitnik, M., Hallac, D., Leskovec, J.: Learning structural node embeddings via diffusion wavelets. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1320–1329 (2018)
8. 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)

9. Jiang, B., et al.: Retweeting behavior prediction based on one-class collaborative filtering in social networks. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 977–980 (2016)
10. Jin, Y., Song, G., Shi, C.: Gralsp: graph neural networks with local structural patterns. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 4361–4368 (2020)
11. Kim, Y.A., Srivastava, J.: Impact of social influence in e-commerce decision making. In: Proceedings of the Ninth International Conference on Electronic Commerce, pp. 293–302 (2007)
12. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
13. Liu, Y., Zhao, J., Xiao, Y.: C-rbfnn: a user retweet behavior prediction method for hotspot topics based on improved rbf neural network. Neurocomputing **275**, 733–746 (2018)
14. Luo, Z., Osborne, M., Tang, J., Wang, T.: Who will retweet me? finding retweeters in twitter. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 869–872 (2013)
15. Matsubara, Y., Sakurai, Y., Prakash, B.A., Li, L., Faloutsos, C.: Rise and fall patterns of information diffusion: model and implications. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 6–14 (2012)
16. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
17. Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., Tang, J.: Deepinf: social influence prediction with deep learning. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2110–2119 (2018)
18. Riquelme, F., González-Cantergiani, P.: Measuring user influence on twitter: a survey. Inform. Process. Manag. **52**(5), 949–975 (2016)
19. Sun, F.Y., Hoffmann, J., Verma, V., Tang, J.: Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. arXiv preprint arXiv:1908.01000 (2019)
20. Sun, J., Wang, X., Xiong, N., Shao, J.: Learning sparse representation with variational auto-encoder for anomaly detection. IEEE Access **6**, 33353–33361 (2018)
21. Tang, J., Sun, J., Wang, C., Yang, Z.: Social influence analysis in large-scale networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 807–816 (2009)
22. Tang, J., Wu, S., Sun, J.: Confluence: conformity influence in large social networks. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 347–355 (2013)
23. Tong, H., Faloutsos, C., Pan, J.Y.: Fast random walk with restart and its applications. In: Sixth International Conference on Data Mining (ICDM'06), pp. 613–622. IEEE (2006)
24. Ugander, J., Backstrom, L., Marlow, C., Kleinberg, J.: Structural diversity in social contagion. Proc. Nat. Acad. Sci. **109**(16), 5962–5966 (2012)
25. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)

26. Wang, J., Wang, Y.Q.: Sir rumor spreading model with network medium in complex social networks. Chinese Journal of Physics (2015)
27. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. Nature **393**(6684), 440–442 (1998)
28. Wu, H., Hu, Z., Jia, J., Bu, Y., He, X., Chua, T.S.: Mining unfollow behavior in large-scale online social networks via spatial-temporal interaction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 254–261 (2020)
29. Xiong, J., Chen, X., Yang, Q., Chen, L., Yao, Z.: A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing. IEEE Transactions on Network Science and Engineering (2019)
30. Yang, M.C., Lee, J.T., Lee, S.W., Rim, H.C.: Finding interesting posts in twitter based on retweet graph analysis. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1073–1074 (2012)
31. Yi, B., et al.: Deep matrix factorization with implicit feedback embedding for recommendation system. IEEE Trans. Ind. Inform. **15**(8), 4591–4601 (2019)
32. Yuan, N.J., Zhong, Y., Zhang, F., Xie, X., Lin, C.Y., Rui, Y.: Who will reply to/retweet this tweet? the dynamics of intimacy from online social interactions. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pp. 3–12 (2016)
33. Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V.: Heterogeneous graph neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 793–803 (2019)
34. Zhang, J., Tang, J., Li, J., Liu, Y., Xing, C.: Who influenced you? predicting retweet via social influence locality. ACM Trans. Knowl. Discovery Data (TKDD) **9**(3), 1–26 (2015)
35. Zhang, J., Tang, J., Ma, C., Tong, H., Jing, Y., Li, J.: Panther: Fast top-k similarity search on large networks. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1445–1454 (2015)
36. Zhang, M., Cui, Z., Neumann, M., Chen, Y.: An end-to-end deep learning architecture for graph classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
37. Zhang, X., Han, D.D., Yang, R., Zhang, Z.: Users’ participation and social influence during information spreading on twitter. PloS One **12**(9), e0183290 (2017)
38. Zhao, Q., Erdogdu, M.A., He, H.Y., Rajaraman, A., Leskovec, J.: Seismic: a self-exciting point process model for predicting tweet popularity. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1513–1522 (2015)