IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 19, 2018, accepted December 18, 2018, date of publication January 3, 2019, date of current version January 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2890826

Predicting Topic Participation by Jointly Learning User Intrinsic and Extrinsic Preference

FAN ZHOU[1], LEI LIU[1], KUNPENG ZHANG[2], GOCE TRAJCEVSKI[3], AND JIN WU[1]

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[2]Department of Decision, Operations and Information Technologies, University of Maryland at College Park, College Park, MD 20742, USA
[3]Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA

Corresponding author: Jin Wu (wj@uestc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602097 and Grant 61472064, in part by the NSF under Grant III 1213038 and Grant CNS 1646107, in part by ONR under Grant N00014-14-10215, in part by HERE under Grant 30046005, and in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2015J072.

ABSTRACT Understanding the preferences of social media participants plays a crucial role in many business applications. A specific aspect of interest is to predict which topics a particular user is more likely to be involved in. Existing efforts on such topic participation forecasting mainly focus on learning from historical user-generated texts to infer their preferred topics, or leverage on an information propagation theory on networks to predict the topics of potential interest. However, jointly utilizing both the sources of data to provide a holistic prediction for such a task has not been exploited. We present a novel joint learning framework that takes advantage of both users' *intrinsic* (learned from their past social media postings) and *extrinsic* preference (learned from the social network influence) and that embeds them into low-dimensional space vectors. To facilitate effective learning, we encode latent continuous embedded vectors into binary ones via locality-sensitive hashing. Furthermore, to explain the predictions made by our ''black-box'' model, we investigate the importance of each training sample on the topic prediction performance of a testing instance to demonstrate its interpretability. The experiments conducted on datasets collected from several popular social media platforms demonstrate the effectiveness of our proposed method when compared with existing baselines.

INDEX TERMS Topic participation prediction, locality-sensitive hashing, graph convolutional networks, influence function, interpretability.

I. INTRODUCTION

Social media platforms and online discussion forums, such as Twitter, Facebook, and Weibo, are attracting millions of users daily, and encourage users to generate content, participate in discussions on topics of interest, even share their personal experiences with friends, etc. Most have mechanisms to facilitate user engagement by allowing individuals to efficiently seek and categorize their content into specific themes, such as the hashtags (widely used in Twitter) for referring to topics. For instance, ''#tbt'' (or ThrowbackThursday) has been a popular hashtag where users often post nostalgic pictures or stories of their past. With the prevalence and popularity of hashtag-based topic participation, understanding which topics a user is interested in has become an emerging research problem. Accurately predicting individual's topic participation can benefit many downstream business applications – including commodity recommendation, online precise targeting, customer retention management, even disease spread detection and public health monitoring [1], etc.

Activities reported by users on social media platforms are usually relevant to their interests to some extent. For example, a user who often posts tweets related to basketball games or players is very likely to pay attention to sports-related topics. Conversely, user interests can be influenced by other crowds, especially their social friends. According to herding behavior theory in psychology [2], users are more willing to get involved in some trending or emerging topics being extensively discussed in their social circles. For example, when topic ''Ice Bucket Challenge'' became an Internet sensation, many users joined the challenge and posted relevant videos/messages. Therefore, to better predict the topics that users may participate on, we should understand both intrinsic preference, reflected by their historical activities (e.g., social media postings) and extrinsic preference (mainly from their social network influence).

Previous works on similar problems focus on understanding users intrinsic preferences or extrinsic preferences separately. In the former line of research, different approaches have been proposed to capture the preference propensity between a user and topics via user's posting history [3]. This, however, limits the performance by the local optimum because other important information encoded in various sources of data – e.g., networks and the topics participated by their social circles or the crowds, are ignored. In contrast, the latter research leverages theories in information propagation to forecast which topics are likely to go viral and then recommend those emerging topics to relevant users [4], [5]. Typically, these approaches build upon information diffusion model and rely on network structure to predict the topic propagation. One of the major drawbacks of these methods is the effectiveness, which heavily depends on the hypothesis of the underlying diffusion model and the structural features that are often hard to specify in practice [5]. Thus, despite successes in propagation prediction, their disadvantage for next topic participation prediction exists due to lack of modeling users intrinsic preferences.

We approach the Next Topic Participation Prediction Problem (NTP^3) by learning user's latent topic preference through leveraging both posting history and the social network structure. We formally model it as a multi-label classification where posting content and social network structures are jointly learned to characterize latent relationships between topic participation and users. Our proposed method consists of two major components: (1) *topic* embedding; and (2) *network* embedding – combined to learn user intrinsic and extrinsic preferences. Instead of simply applying word2vec [6]-based word embedding, we embed topic(s) that users have participated on to represent users' intrinsic preference. Additionally, we develop a convolutional graph embedding approach to capture spatial and dynamic structures existing in social networks. The content of individual's historical posting (normally a short-text or a photo(s)) and the network structures are embedded as low-dimensional vectors. To facilitate effective learning, we further propose a novel framework that converts continuous latent embedding vectors into binary representations via Locality-Sensitive Hashing (LSH), serving as the input to the convolutional neural networks for the final topic participation classification. LSH preserves both content and network similarities that reflect user intrinsic and extrinsic preferences. Such a design via LSH transformation can improve the efficiency of convolution operations – more importantly, it can well distinguish users in the low-dimensional space while preserving their similarities, significantly enhancing the prediction performance. To explain the predictions of the proposed "black-box" model, we borrow the idea from [7] to interpret its modeling behavior by investigating the importance of each training sample on the prediction performance of every testing instance. To the best of our knowledge, this is the first attempt to predict next topic participation via combining both intrinsic and extrinsic preferences while incorporating LSH to significantly improve the prediction accuracy, as well as making the model interpretable. In summary, our main contributions are:

- We study an important and practical task in various application domains: the Next Topic Participation Prediction Problem (NTP^3). We leverage both user content history and social network structure to understand user latent preference. To make our "black-box" model more explicitly explainable, we borrow the idea of influence maximization to interpret the performance of our model using robust statistics.
- We incorporate Locality-Sensitive Hashing to efficiently encode representations learned from both content and network structure into a low-dimensional binary vectors and subsequently use them as the input to a convolutional neural network for the final user-topic participation prediction.
- We collect real-world datasets from several major social networks (content-sharing platforms Twitter and Weibo, and knowledge sharing platforms Zhihu and Douban) to evaluate our proposed method. The experimental results demonstrate that our method can significantly boost the prediction accuracy compared to existing baselines.

In the rest of this paper, we review relevant literatures in Section II and present model-free evidence to motivate our model in Section III. In Section IV, we describe the main methodology including modeling personal and social topic preference, and efficient binary representation via LSH. Experimental evaluations on effectiveness, empirical visualization, and interpretability of our proposed method are presented in Section V, and Section VI concludes the paper.

II. RELATED WORK

We overview the relevant works in three most related areas: topic prediction and recommendation on social media, network representation, and Locality-Sensitive Hashing (LSH).

A. TOPIC PREDICTION AND RECOMMENDATION

Extensive research exists on topic recommendation in online social media. An implicit information network formed by different types of relationships among various entities (e.g., users, topics and micro-blogs) was used to profile user topic interests and make personalized topic recommendation [8] and to link tweets with online articles [9]. Gerani *et al.* [10] address the blog post opinion retrieval task and propose methods that rank blog posts according to their relevance and opinionatedness toward a topic. Then, they propose and investigate different models for aggregating the opinion density at query terms positions to estimate the opinion score of every document. Tan *et al.* [11] exploit textual review information, as well as ratings, to model user preferences and item features in a shared topic space and subsequently introduce them into a matrix factorization model for recommendation. However, user's social preferences are not considered in this work. Zhang and Balog [12] consider semantic representations based on continuous vector representations of words and of entities (i.e., word and graph embeddings) for ad hoc table retrieval. They introduce a framework that handles matching in different semantic spaces in a uniform way, by modeling both the table and the query as sets of semantic vectors. Collaborative filtering, a well-established technique, has been widely used for social content recommendation – particularly for Twitter where metadata and contextual information including source of content, topic interest and social voting are leveraged [13]. Meanwhile, graphical models and ranking algorithms have also been successfully applied for recommending relevant topics to users [14] and enhance the quality of topic-sensitive influence modeling [15]. Liang *et al.* [16] propose a User Clustering Topic (UCT) model to track changes of user's time-varying topic distributions based on both the short postings by users during a given time period and previously estimated distributions. It uses a Gibbs sampling based algorithm to infer user's current topic distribution. In addition, various techniques have been proposed to learn topics based on texts. For example, Cheng *et al.* [17] develops a Biterm Topic Model (BTM) to learn short text topics by directly modeling the generation of word co-occurrence patterns (i.e., biterms) in the corpus, making the inference effective with the rich corpus-level information. To deal with large-scale data, two online algorithms are proposed for efficient topic learning.

Another line of research on topic-popularity forecast aims to predict whether a topic would become prevalent [18]. An incremental clustering framework as well as a range of content and temporal features was developed to detect hot emerging topics [19]. Modeling topics from both sequential and temporal views based on time series data using recurrent neural networks to learn and forecast the popularity of a topic is addressed in [20]. Works also exist focusing on predicting various information cascades [21], exploring individual influences on the network [22], finding the seed nodes and relevant tags that maximize the influence [23] or understanding the underlying mechanism governing the popularity dynamics of cascades [5], [24]. However, these methods provide general understanding of popularity of topics, without modeling the relations of topics to individual users.

The most related work proposed in [3], studies the problem of next topic participation prediction i.e., predicting which topics a user will join in the future on social media. A convolutional neural network based matching method was developed to learn the relations between users and topics, and an external memory network for modeling users' posting history and their topics. The approach, however, fails to consider "macro-cosmic" features such as network structures and social relationship that would evidently have impact on the popularity of a topic in social media and affect the choices of the users. We consider both personal preference and social influence to jointly model the next topic participation prediction, and leverage a heterogeneous embedding network to improve the predictions.

B. NETWORK REPRESENTATION

Since the surge of deep learning, especially word2vec [6] in natural language processing, recent advances in network representation (NR) abound: DeepWalk [25], LINE [26], node2vec [27] and struc2vec [28] approaches. The main goal of these approaches is to encode graph structure into a low-dimensional embedding, which can be used for many downstream applications such as node classification, community detection and link prediction. Node representation approaches can be categorized into: (1) matrix factorization-based approaches [29], which are inspired by the fact that the strength of the relationship between two nodes is proportional to the dot product of their embeddings (low-dimensional vectors) [30]. (2) word2vec-based models, such as DeepWalk [25], LINE [26], node2vec [27], etc., where random walk and word2vec are essential methods for constructing co-occurrence statistics and embedding nodes. As the skip-gram model with negative sampling inherent in word2vec has been shown to be an implicit factorization of a certain word-context matrix [31], recent effort attempts to explain and unify this line of NR models into matrix factorization framework [32]. (3) Graph convolutional networks based methods – e.g., graphCNN [33] and GCN [34], where nodes are normally represented as a function of its surrounding neighborhoods and further embedded using the convolutional kernels. While effective for semi-supervised learning (which can be used for topic prediction), original design assumed learning with the presence of both training and test data, and is not applicable to large and dense graphs [35]. Meanwhile, how to incorporate associated features of nodes and/or their validity in node classification remains elusive for spectrum-based methods.

Network representation models focus on how to efficiently capture the (global or local) network structures. However, they cannot be directly applied in addressing the NTP[3] since it requires information regarding personal and/or in-network posts and corresponding topics. Recently proposed methods for heterogeneous network embedding such as CANE [36] might be well-suited. While considering different types or roles of nodes (such that the random walks are restricted to the transition between particular nodes) and context-aware node embeddings, these methods cannot involve large-scale data (tweets in our task) into the embedding space of nodes, not to mention the historical and temporal topics of individuals.

C. LOCALITY-SENSITIVE HASHING

Locality-Sensitive Hashing (LSH) [37] is a state-of-the-art sub-linear time algorithm for approximate nearest-neighbor search and recently has been introduced to deep learning for various applications such as recommendation [38], metric learning [39], image retrieval and feature learning [40]. It approximately preserves similarity but with a significant dimensionality reduction. Minwise-independent permutations form an LSH for Jaccard coefficient, which was originally proposed in 1998 [41]. It was later used for similarity search in high dimensional data [42]. The basic idea is to hash the data points so as to ensure that the probability of collision is much higher for objects that are close to each other than for those that are far apart. Researchers introduced

TABLE 1. KL divergence of topic distribution (with different numbers of topics) on (1) past postings and recent postings for a particular user; and (2) postings from user and her friends. "Positive sample for a user" means both past and recent social postings are from the same user; "Positive sample for a user and her friends" means social postings from the user and her real friends; "Negative sample for a user and her friends" means social postings from the user *u* and users who are not friends of *u*.

Sampling strategy	Type	5	10	15	20	25	30	35	40	45	50	Average
Positive sample	A user	0.0054	0.1344	0.1352	0.5547	0.4648	0.4862	0.7620	0.6975	0.8226	0.9169	0.4979
	A user and friends	0.4127	0.7758	0.8699	1.3622	1.3402	1.1867	1.3706	1.2807	1.0362	1.2750	1.0910
Negative sample	A user and non-friends	1.2593	2.7085	3.2076	3.6262	3.9594	3.7675	3.9182	4.1276	3.2419	3.8993	3.3715

the idea of using random-hyperplanes to summarize items in a way that respects the cosine distance for real-valued data. It was also suggested that random hyperplanes plus LSH could be more accurate at detecting similar documents than minhashing plus LSH [43]. Techniques for summarizing data points in a Euclidean space, working directly on points without embedding (i.e., in "native" space) have provided data structure up to 40 times faster than *kd-tree*. As data becomes huge, the distributed layered LSH scheme focusing on the Euclidean space under L_2 norm was proposed in [44]. It can exponentially decrease the network cost while maintaining a good load balance between different machines.

More recently, Dasgupta *et al.* [45] apply LSH to alleviate the computational overhead of neural networks via hashing weights and bias into binary representations and preserving inner products. In contrast, we leverage LSH to encode continuous embedding vectors into binary ones while preserving locally similarities for efficient user interest predition.

III. MODEL-FREE EVIDENCE

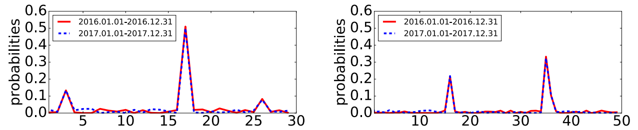
In this section we present model-free evidence to assess the factors affecting user topic participation to motivate our model.

A. INTRINSIC PREFERENCE

We assume user interests tend to be unchanged and consistent with their historical preferences, although they are dynamic and can evolve over time for some individuals. We empirically measure the similarity of users' interests to their past for a set of randomly selected users from each platform in our dataset. For each user we first obtain their social postings in two different time periods: 2016/01/01 – 2016/12/31 and 2017/01/01 – 2017/12/31. Then we run a classic topic modeling (LDA) to obtain two topic distributions. As shown in Fig. 1 (a particular randomly chosen user), the topic distribution in the first period overlaps with that in the second period very well, regardless of different numbers of topics (30 and 50 here). We observe similar patterns for other users as well.

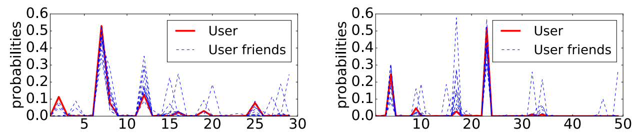
B. EXTRINSIC PREFERENCE

According to the herding theory in psychology [2], user's behavior is more likely to be influenced by their friends or the crowds on social media – i.e., users might be interested in topics that most of their friends are focusing on. To empirically verify this hypothesis, we collect social postings of users and their friends and run LDA to see how similar



(a) (b)

FIGURE 1. Overlap of topic distribution on historical social postings for two time periods on Douban.com. (a) The number of topics = 30. (b) The number of topics = 50.



(a) (b)

FIGURE 2. Topic distribution for social postings of a randomly picked user and his 10 randomly selected friends on Douban.com. (a) The number of topics = 30. (b) The number of topics = 50.

they are. Fig. 2 shows that a randomly selected user and his 10 randomly picked friends on Douban.com, indeed exhibit similar patterns in terms of topic interest in the past.

Moreover, we randomly select positive samples and negative samples (see Table 1 for explanations of positive vs. negative samples) and validate our assumption by varying the numbers of topics. We compute the KL divergence of topic distributions of social postings between two time periods and take the average on all users. Similarly, we calculate the average KL divergence of topic distributions of postings from the user and each of his friends. We find (shown in Table 1) that: (1) the average KL divergence for users is smaller than user and friends, which indicates that user's interest is more similar to her past as compared to her friends'; (2) All KL divergence values are relatively small, which means that user topic interest is similar to his past and friends' preferences, which further consolidates our conclusions. To draw more statistically significant conclusions, we further examine 100 randomly selected users for each platform and calculate two average KL divergences. Table 2 supports our claim (with lowest value of 32%) that majority of users have consistent topic interest with their past and have similar preference to their friends. A possible reason that the percentage for Twitter and Weibo is smaller than the other two platforms is because the length of messages on the former is limited and much shorter than the latter, which restricts the performance of LDA topic modeling. Overall, these findings motivate us to model user historical interest and their friends' preference

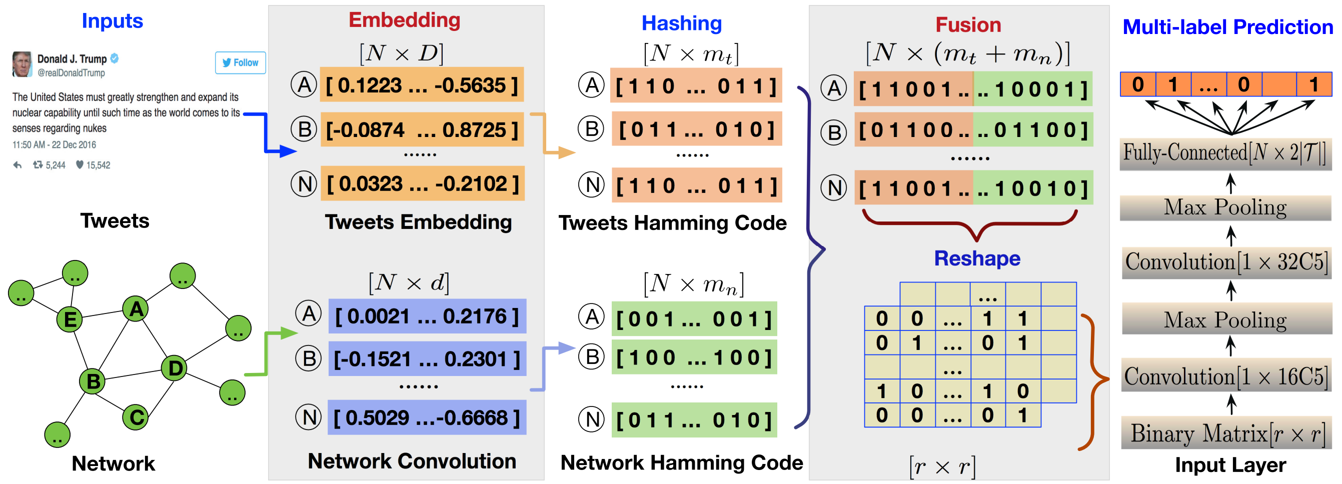


FIGURE 3. Overview of our proposed framework NTP³. The inputs are historical postings for all users and their social relationships/networks (e.g., friendship, following-ship, common interests, etc.). Text embedding and graph/network convolution operations will be applied to two types of inputs to learn user representations in a low-dimensional vector format. LSH subsequently converts them into binary representations which are then combined and reshaped as the input to the final multi-lable classifier to predict the topic participation for a given user. The classifier consists of multi-layer convolutional neural networks. Training and testing are done in such a similar manner.

TABLE 2. Percentage of users (user-friends) with average KL divergence on topic distributions less than 2.0 for each dataset. The number of topics is 50.

Type	Twitter	Weibo	Zhihu	Douban
Users	63%	71%	98%	96%
User-friends	32%	41%	90%	60%

when predicting which topics a user is likely to join in the future.

LSH uses the bucket hashing mechanism to find similar items in a large dataset with statistical guarantee. We apply it in this work to convert continuous embedding vectors into binary hamming encodings. Such a design can make the entire system more efficient because all convolution operations are now on integers rather than floating numbers. In addition, it is possible that two different but similar vectors (e.g.,two vectors are different only in scales, they have the same rank correlation, or each corresponding element in two vectors are in the same specific range, etc.), say $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$, become identical after LSH, which might significantly save computation cost in many cases – i.e., instead of separate operations for $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ before LSH, one of them will suffice after LSH (the other one can be cached).

IV. OUR PROPOSED FRAMEWORK

We now formally define the problem and describe the details of our proposed methodology consisting of four components: text embedding, network embedding, heterogeneous hashing and neural network based prediction.

To model user intrinsic and extrinsic preferences, we require two types of input: textual content (e.g., user historical tweets) and a social network (e.g., following relationship on Twitter). All the latent patterns among users, topics, and friends represented in texts and networks will be modeled by embeddings. LSH is then applied to transform implicit relationships (continuous embedding vectors) into more efficient and effective binary representations. Finally, a multi-label classifier is trained to predict which topics a user is likely to participate in next. The overall framework is shown in Fig. 3.

A. PROBLEM DEFINITION

We use a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of vertices $\mathcal{V}$ ($|\mathcal{V}| = N$) represents users, and the set of edges $\mathcal{E}$ denotes connections between users – i.e., $e_{i,j} \in \mathcal{E}$ indicates a relationship between user u_i and user u_j. Each user $u_i \in \mathcal{V}$ has a list of topics $T_i \in \mathcal{T}$ ($\mathcal{T}$ is the vocabulary of all topics discussed by users in $\mathcal{G}$), and our objective is to predict which topics the given user is likely to join, which can be formally defined as a multi-label classification problem – i.e., learning a classifier that links users to their potentially interested topics: $u_i \in \mathcal{V} \mapsto T_i \in \mathcal{T}$.

As discussed in Section III, the topics a particular user is interested in can be influenced by her past interests and (or) her friends' preferences. Thus, we model both personal topic preference and social influence in our framework, as detailed in the sequel.

B. MODELING INTRINSIC TOPIC PREFERENCE

Instead of embedding all words in postings of a user into vectors (e.g., via word2vec [6]) and then combining them to represent user posting behavior as in [3], we first concatenate all historical content from a user u_e to form a paragraph a_e and then embed the entire paragraph into a vector representation. Consequently, we obtain a set of paragraphs $\mathcal{A}$ as follows:

$$\mathcal{A} = \begin{vmatrix} - & a_1 & - \\ - & . & - \\ - & a_N & - \end{vmatrix}, \quad \mathcal{A} \in \mathbb{R}^{N \times l^*} \tag{1}$$

where N is the number of users, and l^* is the varied length of each paragraph. Now the challenge is how to learn a fixed-length vector $\mathbf{a}_e \in \mathbb{R}^D$ (associated with a user u_e) for each paragraph which can serve as (part of) the input of our model, also referred to as a paragraph representation problem [46]. Inspired by the vector representations in [6], paragraph token is introduced here – which can be considered as a different "word" acting as a memory to remember what is missing from the current context – to map each paragraph into an unique column vector in the (paragraph) embedding matrix $\mathbf{A}$. In turn, every word in each paragraph is also mapped to an unique column vector in the (word) embedding matrix $\mathbf{W}$. Matrices $\mathbf{A}$ and $\mathbf{W}$ are then concatenated to predict the next word in the context sampled from a sliding window over the paragraph. The paragraph vector $\mathbf{a}_e$ is shared for the same paragraph a_e, but not across paragraphs; whereas the word vectors are shared across paragraphs.

Let S denote the number of samples in each paragraph and $\mathbf{Y}^o \in \mathbb{R}^{S\times K}$ denote the output matrix of S samples. We can obtain the output vector for i^{th} sample as follows:

$$\mathbf{y}_i^o = \mathbf{Q}\phi(\phi(\mathbf{w}_{t-c}, \ldots, \mathbf{w}_{t+c}; \mathbf{W}), \mathbf{a}_e; \mathbf{A}) + \mathbf{b} \tag{2}$$

where: vector $\mathbf{y}_i^o \in \mathbb{R}^{1\times K}$ is the i^{th}-row of the matrix $\mathbf{Y}^o$; $\phi(\cdot)$ is constructed by a concatenating operation over vectors for paragraphs or contexts; $(\mathbf{w}_{t-c}, \ldots, \mathbf{w}_{t+c})$ are the context vectors from $\mathbf{W}$ and [t−c, t+c] is the slide window; $\mathbf{a}_e$ is the paragraph vector from $\mathbf{A}$; $\mathbf{Q}$ and $\mathbf{b}$ are the weight and bias, respectively. The probability of each word (or paragraph token) $y_{ij}^o \in \mathbf{y}_i^o$ is calculated with a softmax:

$$y_{ij}^o = \frac{e^{y_{ij}^o}}{\sum_{k=1}^{K} e^{y_{ik}^o}}, \quad \forall j = 1, \ldots, K \tag{3}$$

where K is the word vocabulary size. Assuming y_{it}^o is the probability we want to maximize in $\mathbf{y}_i^o$, we aim at maximizing the following average log-probability for each paragraph:

$$\frac{1}{S}\sum_{i=1}^{S}\log(y_{it}^o) \tag{4}$$

To facilitate stochastic gradient descent for parameter estimation in the training, including $\mathbf{A}$, $\mathbf{W}$, $\mathbf{Q}$, $\mathbf{b}$ – we rewrite the above average log-probability function as an equivalent minimization objective in a way of reflecting the cross entropy:

$$\ell(\mathbf{Y}, \mathbf{Y}^o) = \arg\min_{A,W}\Big(-\frac{1}{S}\sum_{i=1}^{S}\frac{1}{K}\sum_{j=1}^{K} y_{ij}\log(y_{ij}^o)\Big) \tag{5}$$

where $\mathbf{Y}$ is the ground-truth of $\mathbf{Y}^o$ and $y_{ij} \in \mathbf{Y}$.

After the model converges, we can finally obtain the paragraph vector a_e by looking up the matrix A. We denote the paragraph vector a_e as u_e^t (user u_e topic embedding) for the purpose of consistency with the social representation in the sequel.

C. MODELING EXTRINSIC TOPIC PREFERENCE

To model the effect of friends' topic preference on a user's topic interest, we need to find a good representation of each user in a network and capture the latent network structural semantics. This can be accomplished by various network embedding (NE)-like methods.

In the context of NTP[3], we are more interested in neighborhood nodes than distant ones – when predicting the potential interested topic(s) of node u_i, the attributes and connectivity of nearby nodes provide more useful side information or additional context. That is, *local* structure attributes play more important role in predicting one's interests than high-order proximity of the network. Therefore, Graph Convolutional Networks (GCN) [34] is a natural choice since it focuses on aggregating 1−hop or 2−hop neighborhood of the graph in the spatial domain and is, more importantly, efficient compared to spectral method [47] which requires computationally expensive graph Fourier transformation. Nevertheless, interest groups in a social network may dynamically change. For example, a user usually follows another user(s) because of joining a new topic discussion. However, traditional *transductive* methods such as GCN [34], DeepWalk [25] and node2vec [27] learn embedding on a fixed network, which cannot be generalized to unseen/new nodes.

Taking into account above considerations, we propose an *inductive* GCN to learn the *extrinsic* representation of users.

1) SPECTRAL CONVOLUTION ON GRAPH

To be specific, we represent $\mathcal{G}$ with a symmetric adjacency matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ ($S_{ij} = 0$ if $(i,j) \notin \mathcal{E}$ and $S_{ij} > 0$ otherwise) and a diagonal degree matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ ($D_{ii} = \sum_j S_{ij}$). The (unnormalized) graph Laplacian is a symmetric positive-semidefinite matrix ($\in \mathbb{R}^{N\times N}$) $\mathbf{\Delta}_u = \mathbf{D} - \mathbf{S}$. By a symmetric normalization, one can define a normalized graph Laplacian $\mathbf{\Delta} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{S})\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{I}$ is the identity matrix. Since $\mathbf{\Delta}$ is also an symmetric positive-semidefinite matrix, it can be diagonalized as $\mathbf{\Delta} = \mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Psi}^\intercal$, where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_N)$ are the spectrum (non-negative eigenvalues) of $\mathbf{\Delta}$ and $\mathbf{\Psi} = (\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_N)$ are the corresponding orthonormal eigenvectors.

The convolution operator on graph $\mathcal{G}$ is defined in the Fourier domain for a signal $\mathbf{x} \in \mathbb{R}^N$ such that

$$g_\theta \star \mathbf{x} = \mathbf{\Psi} g_\theta \mathbf{\Psi}^\intercal \mathbf{x} = \mathbf{\Psi} g_\theta(\mathbf{\Lambda})\hat{\mathbf{x}} \tag{6}$$

where $\hat{\mathbf{x}} = \mathbf{\Psi}^\intercal\mathbf{x}$ is *graph Fourier transform* of $\mathbf{x}$ defined as the expansion of a signal in terms of the eigenvectors of the graph Laplacian, and $g_\theta(\mathbf{\Lambda})$ is a function of the eigenvalues of $\mathbf{\Delta}$ parameterized by Fourier coefficients $\theta \in \mathbb{R}^N$.

2) SPATIALLY LOCALIZED CONVOLUTIONS

Defferrard *et al.* [33] used Chebyshev polynomials (of the first kind) of degree K defined in a recursive manner

$$g'_\theta \star \mathbf{x} \approx \sum_{k=0}^{K}\theta'_k T_k(\hat{\mathbf{\Delta}})\mathbf{x} \tag{7}$$

where $T_k(\hat{\mathbf{\Delta}}) \in \mathbb{R}^{N\times N}$ is the Chebyshev polynomial forming an orthogonal basis for the space of polynomials of order K and is evaluated at the rescaled $\hat{\mathbf{\Delta}} = 2\lambda_{max}^{-1}\mathbf{\Delta} - \mathbf{I}$. Consequently, the spectral convolution of the K^{th} order polynomials of the Laplacian equals to K-localized filter which combines the information of a vertex and its K-hop neighbors.

It has been proved that Chebyshev polynomials is hard to produce narrow-band filters [48]. However, we intuitively expect that it can capture local neighborhood structures such as community structures (i.e., interest group in topic prediction problem) where the Laplacian has clusters of eigenvalues concentrated around a few frequencies with large spectral gap.

To overcome the overfitting issue and improve the computational efficiency, we use the first order polynomial proposed in GCN [34] and assume $\lambda_{max} = 2$, $K = 1$ for approximating graph convolution as

$$g'_\theta \star \mathbf{x} \approx \theta(\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}})\mathbf{x} = \theta(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{S}}\hat{\mathbf{D}}^{-\frac{1}{2}})\mathbf{x} \tag{8}$$

where the last step is a re-parameterization trick w.r.t the summation of the adjacency matrix of the undirected graph $\mathcal{G}$ and the identity matrix, i.e., $\hat{\mathbf{S}} = \mathbf{S} + \mathbf{I}$, due to the eigenvalues of $\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$ are in the range $[0, 2]$. When optimizing with deep neural networks, renormalization is applied at each layer to retain numerical stability and alleviate the problem of exploding/vanishing gradients [34]. In addition, Eq.(8) takes the self-connections $\hat{D}_{ii} = \sum_j \hat{S}_{ij}$ into account.

3) TOPIC PREDICTION WITH EXTRINSIC PREFERENCE

Let $\mathbf{X} \in \mathbb{R}^{N\times d}$ be a signal, where each user u_i has a d-dimensional feature vector $\mathbf{x}_i$. To obtain the convolved signal matrix $\mathbf{Z} \in \mathbf{R}^{N\times F}$, we generalize convolution (Eq.(8)) and F filters on signal $\mathbf{X}$ as

$$\mathbf{Z} = \hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{S}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{F} \tag{9}$$

where $\mathbf{F} \in \mathbb{R}^{d\times F}$ denote the matrix of parameters. Assume we have a L-layers neural network, each layer of which can be trained as

$$\mathbf{H}^l = \text{ReLU}(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{S}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{l-1}\mathbf{\Omega}^l) \tag{10}$$

where $\mathbf{\Omega}_t^l$ is a layer-specific trainable weight matrix, ReLU is the activation function, and $\mathbf{H}^0 = \mathbf{X}$, i.e., the input feature matrix of the data. Finally, we can predict the potential interested topic(s) for N users by the output layer $\mathbf{H}^L$ of the graph convolutional network.

In consideration of the dynamic changes of the network, we extend above spatially localized graph convolution to an inductive manner by simply leveraging a mean operation over a filter on user u_i as : $\mathbf{h}_i^l = \text{MEAN}(\{\mathbf{h}_i^{l-1}\} \cup \{\mathbf{h}_k^{l-1}, \forall k \in \mathcal{N}(i)\})$, where $\mathbf{h}_i^l$ is the i^{th} row of $\mathbf{H}^l$ and $\mathcal{N}(i)$ denotes the neighborhood nodes of u_i. That is, we consider the local feature information rather than merely node itself when updating Eq.(10). Note that this modification indeed reinforce the influence from neighborhood nodes, since it has been considered in graph convolution as we stated above. However, this trick allows the unseen/new-join nodes to be initialized appropriately and an inductive graph embedding as observed in [30] – note this modification differs from mean aggregator used in [30] in that we adapt the mean operation in the graph convolutional embedding while GraphSage [30] only embeds graph with various aggregation such as mean and pooling.

4) VARIANTS

Since our ultimate goal is to predict topic(s) a user is likely to be participating in, the social link plays a very important role in identifying the extrinsic interests as observed in Section III. Based on the social connections, we can infer the topics using above graph convolution method. In Eq.(10), there are two variants corresponding to different input feature matrices $\mathbf{X}$. First, if we consider only the network information, $\mathbf{X}$ can be initialized as the identity matrix $\mathbf{I}$. On the other hand, if we take the node feature (e.g., the word frequency of user postings), we can also embed the personal features into network representation. In both cases, one can directly predict the topics without using users' intrinsic preference – strictly speaking, the coarse-grained posting feature has been utilized in the latter case. However, as we will observe in Section V, the complete network structured based methods (and the coarse-grained heterogeneous semi-supervised GCN based method) perform worse compared to our hashing based method introduced in Section V-F.

5) IMPLEMENTATION

In practical implementation, we use a nonlinear graph autoencoder for embedding networks inspired by [49]. Specifically, a two-layer neural network is used to reconstruct the *normalized* adjacency matrix as $\hat{\mathbf{A}} = \sigma(\mathbf{Z}^\intercal\mathbf{Z})$, where $\mathbf{Z}$ is the latent space learned with Eq.(9) and Eq.(10). We denote the social representation of each in-network user by $\mathbf{u}_i^n$, i.e., the i^{th} row of $\hat{\mathbf{Z}}$ is denoted as $\hat{\mathbf{z}}_i = \mathbf{u}_i^n$ for convenience of discussion. Note that the extrinsic learning requires computational complexity of $\mathcal{O}(N)(F \cdot d \cdot |\mathcal{E}|)$ and the $\mathcal{O}(N)(|\mathcal{E}|)$ memory, i.e., both are in linear w.r.t the number of edges.

D. THE JOINT LEARNING MODEL WITH LSH

After above two learning processes, we obtain the vectors $\mathbf{u}_i^t \in \mathbb{R}^D$ and $\mathbf{u}_i^n \in \mathbb{R}^d$ for u_i, respectively denoting personal topic preference and social topic preference. Now we explain how to combine the two preferences together to make an effective and efficient prediction. We propose to convert each continuous vector into a binary representation while preserving similarities using Locality-Sensitive Hashing (LSH) [37]. A hash function h is (S, cS, p_1, p_2)-sensitive if, for any two D (or d)-dimensional points $x, y \in \mathbb{R}^{D(d)}$, it satisfies the following: 1.

1) if $sim(x, y) \geq S$, then $Pr_h(h(x) = h(y)) \geq p_1$;
2) if $sim(x, y) \leq cS$, then $Pr_h(h(x) = h(y)) \leq p_2$;

where S is a threshold of interest; $c < 1$ and $p_1 > p_2$ for efficiently approximating nearest neighbor search; sim is a similarity measure – cosine similarity in the proposed NTP^3. The motivation behind applying LSH here is to maximize the probability that similar data points are projected to similar binary representations. Thus, we perform the following operations for vectors here: 1.

1) Generate Hamming encoding $\mathbf{m}_i^t \in \{0,1\}^{m_t}$ for $\mathbf{u}_i^t$ via LSH.
2) Generate Hamming encoding $\mathbf{m}_i^n \in \{0,1\}^{m_n}$ for $\mathbf{u}_i^n$ via LSH.
3) Concatenate $\mathbf{m}_i^t$ and $\mathbf{m}_i^n$ to form $\mathbf{m}_i = [\mathbf{m}_i^t; \mathbf{m}_i^n] \in \{0,1\}^{m_t+m_n}$.

where we project the heterogeneous representations into two separate binary vectors. They work similarly to Sign Random Projection (SRP) [50], i.e., given a vector $\mathbf{u}_i^*$, it utilizes a random vector $\mathbf{r}$ with each bit $\mathbf{r}(b)$ generated by an i.i.d. normal distribution. In particular, it only requires to store the sign of the projection:

$$h_{\mathbf{r}}(\mathbf{u}_i^*) = \begin{cases} 1, & \text{if } \mathbf{r}^\intercal \mathbf{u}_i^* \geq 0; \\ 0, & \text{if } \mathbf{r}^\intercal \mathbf{u}_i^* < 0. \end{cases} \tag{11}$$

Choosing a normally distributed random variable for each dimension is same as picking a random hyperplane [50]. It has been proven in [51] that with rounding scheme for the semidefinite programming relaxation of MAX-CUT, the collision probability between two vectors $\mathbf{u}_i^*$ and $\mathbf{u}_j^*$ under SRP is:

$$Pr(h_{\mathbf{r}}(\mathbf{u}_i^*) = h_{\mathbf{r}}(\mathbf{u}_j^*)) = 1 - \frac{1}{\pi}\left(\cos^{-1}\left(\frac{\mathbf{u}_i^{*\intercal}\mathbf{u}_j^*}{||\mathbf{u}_i^*||_2||\mathbf{u}_j^*||_2}\right)\right) \tag{12}$$

where term $\frac{\mathbf{u}_i^{*\intercal}\mathbf{u}_j^*}{||\mathbf{u}_i^*||_2||\mathbf{u}_j^*||_2}$ refers to the cosine similarity. Note that we have a combined binary vector $\mathbf{m}_i \in \{0,1\}^{m_t+m_n}$ (step 3), where m_* denotes the bits of random vector $\mathbf{r}$, i.e., the number of hyperplane used for hash projection in Eq.(11).

After mapping the heterogeneous embeddings into a Hamming space, we reshape the dimensional vector $\mathbf{m}_i$ to a binary matrix $\mathbf{B} \in \mathbb{R}^{r\times r}$, and apply a convolutional operation on $\mathbf{B}$ with a kernel $\mathbf{k} \in \mathbb{R}^{w\times w}$ to obtain the feature maps $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_f\}$. Max-over-time pooling is used on the feature maps to learn the underlying representation of matrix $\mathbf{B}$.

E. MULTI-LABEL CLASSIFIER FOR NEXT TOPIC PARICIPATION PREDICTION

A user may be interested in more than one topic, we formulate the next topic participation prediction task as a multi-label classification problem. A common and popular method is to map the output to a value between 0 and 1 with an activation function (e.g., sigmoid/softmax) and use a predefined threshold ξ (e.g., 0.5) to make a classification. However, this method introduces an additional parameter ξ and requires tuning ξ for different datasets. To overcome this problem, we design a multi-label classifier to learn the relationships between users and topics.

Specifically, suppose the true label vector for user u_i is $\mathbf{l}_i \in \mathbb{R}^{1\times|\mathcal{T}|}$, where each element denotes whether u_i is interested in a topic $t_j \in \mathcal{T}$. We first obtain a *element inverse* vector (negative label vector) $\bar{\mathbf{l}}_i$ of $\mathbf{l}_i$ and concatenate the two to form a new vector $\mathbf{L}_i = [\mathbf{l}_i; \bar{\mathbf{l}}_i] \in \mathbb{R}^{1\times 2|\mathcal{T}|}$.

Algorithm 1 Learning of NTP^3

Input: $\mathcal{A}$: user posting content; $\mathcal{G}$: social networks; $\mathcal{T}$: topic set; $u_i \in \mathcal{V}$: users.
`/* Intrinsic Preference Learning */`
Generate content embedding $\mathbf{u_i^t}$ via Eq.(5).
Hashing $\mathbf{u_i^t}$ to binary code $\mathbf{m}_i^t \in \{0,1\}^{m_t}$ for $\mathbf{u_i^t}$ via Eq.(11).
`/* Extrinsic Preference Learning */`
Generate network embedding $\mathbf{u_i^n}$ via Eq.(10)
Hashing $\mathbf{u_i^n}$ to binary code $\mathbf{m}_i^n \in \{0,1\}^{m_n}$ via Eq.(11).
`/* Efficient Binary Representation */`
Obtain $\mathbf{m_i}$=[$\mathbf{m^t}$;$\mathbf{m^n}$].
Reshape $\mathbf{m_i}$ to matrix $\mathbf{B} \in \mathbb{R}^{r\times r}$.
`/* Learning with CNN */`
repeat
  Training with CNN according to Eq.(13).
until *converge*;
`/* predicting */`
Output: Topics predicted by Eq.(14).

Our convolutional neural network (CNN)-based classifier predicts both the positive and negative labels for a user and stores the results in a vector $\hat{\mathbf{L}}_i$. The training objective of CNN is as follows:

$$\underset{\theta\in\Theta}{\arg\min}\left\{-\frac{1}{2|\mathcal{T}|}\sum_{j=1}^{2|\mathcal{T}|}[\mathbf{L}_i^j \log \hat{\mathbf{L}}_i^j + (1-\mathbf{L}_i^j)\log(1-\hat{\mathbf{L}}_i^j)]\right\} \tag{13}$$

where $\hat{\mathbf{L}}_i^j$ denotes the predicted result for j^{th} element in the vector $\hat{\mathbf{L}}_i$, $\mathbf{L}_i^j$ is the corresponding ground-truth, and $\theta \in \Theta$ are the parameters.

Finally, we use an element (bit) operation function $\kappa(\cdot)$ to map the vector $\hat{\mathbf{L}}_i \in \mathbf{R}^{1\times 2|\mathcal{T}|}$ to the vector $\hat{\mathbf{l}}_i \in \mathbb{R}^{1\times|\mathcal{T}|}$ as:

$$\hat{\mathbf{l}}_i = \kappa(\hat{\mathbf{L}}_i) = \begin{cases} 1, & \text{if } \hat{\mathbf{L}}_i^j \geq \hat{\mathbf{L}}_i^{j+|\mathcal{T}|}; \\ 0, & \text{if } \hat{\mathbf{L}}_i^j < \hat{\mathbf{L}}_i^{j+|\mathcal{T}|}. \end{cases}, \quad \forall j = 1, \cdots, |\mathcal{T}| \tag{14}$$

Algorithm 1 illustrates the process of our approach. Note that it does not depend on specific node embedding approach. The inductive graph convolutional method proposed in this work can be substituted with alternatives. Meanwhile, more sophisticated LSH methods such as ALSH [52] and RLSH [53] can also be adopted for unifying heterogeneous preferences.

F. MODEL INTERPRETABILITY

We use influence functions [54] to explain the predicting behavior of NTP^3 by investigating the importance of each user on the topic prediction performance of a testing instance $\mathbf{u}^{\text{test}}$.

Following [7], removing a data point $\mathbf{u}$ from the training set results in a change of $\hat{\theta}_{-\mathbf{u}} - \hat{\theta}$, where $\hat{\theta}_{-\mathbf{u}}$ is the optimal θ with the minimum total loss without the data point u, denoted by: $\hat{\theta}_{-\mathbf{u}} \overset{\text{def}}{=} \arg\min_{\theta\in\Theta} \sum_{\mathbf{u}_i \neq \mathbf{u}} \ell_\theta(\mathbf{L}_i, \hat{\mathbf{L}}_i) - \ell_\theta(\cdot)$ is the loss for $\mathbf{u}_i$ (c.f Eq.(13)). To avoid re-training the model for each removed $\mathbf{u}$, Koh and Liang [7] use the influence functions to efficiently approximate this behavior. The basic idea is to compute the parameter changes if $\mathbf{u}$ was upweighted by some small ϵ, which gives the new parameters $\hat{\theta}_{\epsilon,\mathbf{u}} \overset{\text{def}}{=} \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{i=1}^{N} \ell_\theta(\mathbf{L}_i, \hat{\mathbf{L}}_i) + \epsilon\ell_\theta(\mathbf{L}_\mathbf{u}, \hat{\mathbf{L}}_\mathbf{u})$, N is the total number of instances (users). The influence of upweighting $\mathbf{u}$ on the parameters $\hat{\theta}$ is given by

$$\mathcal{I}_{\text{up},\hat{\theta}}(\mathbf{u}) \overset{\text{def}}{=} \frac{\partial\hat{\theta}_{\epsilon,\mathbf{u}}}{\partial\epsilon}\Big|_{\epsilon=0} = -\mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell_{\hat{\theta}}(\mathbf{L}_\mathbf{u}, \hat{\mathbf{L}}_\mathbf{u}) \tag{15}$$

where $\mathbf{H}_{\hat{\theta}}$ is the Hessian matrix. Eq.(15) implies that removing $\mathbf{u}$ is the same as upweighting it by a factor $\epsilon = -\frac{1}{N}$. It allows us to linearly approximate the parameter change of removing $\mathbf{u}$ as $\hat{\theta}_{-\mathbf{u}} - \hat{\theta} \approx -\frac{1}{N}\mathcal{I}_{\text{up},\hat{\theta}}(\mathbf{u})$ *without* re-training the model. Therefore, the influence of upweighting a train point $\mathbf{u}$ on the loss for a testing point $\mathbf{u}^{\texttt{test}}$ can be computed by [7]:

$$\mathcal{I}_{\text{up,loss}}(\mathbf{u}, \mathbf{u}^{\text{test}}) = -\nabla_\theta\ell_{\hat{\theta}}(\mathbf{L}_{\mathbf{u}^{\text{test}}}, \hat{\mathbf{L}}_{\mathbf{u}^{\text{test}}})^\intercal \mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell_{\hat{\theta}}(\mathbf{L}_\mathbf{u}, \hat{\mathbf{L}}_\mathbf{u}) \tag{16}$$

In order to speed up the computation, one usually uses implicit Hessian-vector products (HVPs) to approximate $\mathcal{S}_{\text{test}} \overset{\text{def}}{=} \mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell_{\hat{\theta}}(\mathbf{L}_{\mathbf{u}^{\text{test}}}, \hat{\mathbf{L}}_{\mathbf{u}^{\text{test}}})$. The $\mathcal{I}_{\text{up,loss}}(\mathbf{u}, \mathbf{u}^{\text{test}})$ can be rewritten as $\mathcal{I}_{\text{up,loss}}(\mathbf{u}, \mathbf{u}^{\text{test}}) = -\mathcal{S}_{\text{test}}\nabla_\theta\ell_{\hat{\theta}}(\mathbf{L}_\mathbf{u}, \hat{\mathbf{L}}_\mathbf{u})$. In addition, due to the the fact $\mathbf{H}_{\hat{\theta}} \succ 0$, we have:

$$\begin{aligned}&\mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell_{\hat{\theta}}(\mathbf{L}_{\mathbf{u}^{\text{test}}}, \hat{\mathbf{L}}_{\mathbf{u}^{\text{test}}}) \\ &\quad = \arg\min_{\boldsymbol{\nu}}\{\frac{1}{2}\boldsymbol{\nu}^\intercal\mathbf{H}_{\hat{\theta}}\boldsymbol{\nu} - \nabla_\theta\ell_{\hat{\theta}}(\mathbf{L}_{\mathbf{u}^{\text{test}}}, \hat{\mathbf{L}}_{\mathbf{u}^{\text{test}}})^\intercal\boldsymbol{\nu}\}\end{aligned}$$

where the exact solution $\boldsymbol{\nu}$ can be obtained with conjugate gradients that only requires evaluation of $\mathbf{H}_{\hat{\theta}}\boldsymbol{\nu}$ instead of explicitly computing $\mathbf{H}_{\hat{\theta}}^{-1}$. More details of investigating the importance of a particular training sample are referred to [7].

V. EVALUATION

We now present our experimental observations by comparing our approach with several baseline methods on four real-world datasets.

A. EXPERIMENTAL SETTINGS

The main parameters involved in our framework are listed as follows:

- The dimensionality for paragraph embedding and network embedding are both 128 (i.e., $D = d = 128$).
- We hash the text and network embedding representations into hamming codes with the length of $m_t = 240$ and $m_n = 784$, respectively. We then combine two hamming codes and reshape into rXr square matrices ($r = 32$).
- Our final topic participation prediction is a multi-lable classifier which consists of a two-layer CNN architecture: "(1 × 16C5)-MP-(1 × 32C5)-MP-FC", where 16C5 denotes 16 kernels of size 5 × 5, MP indicates the max-pooling layer and FC is the fully-connected layer.
- We use ReLU as the activation function.
- The batch size is 64.
- The dropout rate is 0.7.
- The learning rate is initialized with 0.001 and decays with a rate of 0.98.

TABLE 3. Descriptives of datasets. # AvgTopics denotes the average number of topics a user participates in.

	Twitter	Weibo	Douban	Zhihu
# Topics	55	12	154	89
# AvgTopics	8.5	1.5	56.7	6.1
# Users	2,673	6,732	2,241	2,368
# Tweets	828,254	84,168	1,362,789	754,015
# Friends	106,244	96,496	145,962	374,925
Training period	2015.01.01-2016.12.31	2013.01.14-2013.12.31	2016.01.01-2016.12.31	2015.06.01-2016.12.31
Testing period	2017.01.01-2017.12.31	2014.01.01-2014.05.12	2017.01.01-2017.12.31	2017.01.01-2017.12.22

B. DATASETS

We collect data from four major social media platforms:

- **Weibo** (www.weibo.com) – one of the most popular Chinese micro-blogging websites with 140 million daily active users, akin to a hybrid of Twitter and Facebook.
- **Zhihu** (www.zhihu.com) – the largest online Q&A website in China akin to Quora. Users can follow others, answer questions, and join topic discussions in groups. The quality of questions, answers, and discussion is relatively high, since users are more focusing on their preferred and experienced domains.
- **Douban** (www.douban.com) – a social network service for users to share content on topics of movies, books, music,and off-line events in Chinese cities, with over 300 million monthly unique visitors.
- **Twitter** (https://www.twitter.com) – an US-based online news and social networking service.

To build and validate our classifier, we need to construct training and testing dataset. For each dataset, we split activities into training set (ranging from 5 months to 2 years) and testing set (1 year). Since the quality of data is important for model performance, we preprocess the data as follows: (i) we remove users who participated in a very few number of topics (e.g., $< \alpha$); (ii) we ignore isolated users in the social network; and (iii) we remove topics being discussed by few participants (e.g., $< \beta$). α and β are parameters that can be tuned manually. The data after preprocessing is shown in Table 3.

C. BASELINES

Traditional multi-lable classification methods with various supervised learning models:

- **Logistic Regression (LR)**
- **Decision Tree (DT)**
- **Random Forest (RF)**
- **Support Vector Machine (SVM)**
- **K-Nearest Neighbors (KNN)**
- **Naive Bayes (NB)**

Collaborative filtering based model:

- We use a user-based collaborative filtering approach to find top-5 similar users and use their topics as the predicted topics for a focal user.

Embedding-based methods:

- **DeepWalk**: it performs random walks over the network and employs Skip-Gram model to learn node embeddings upon which user similarity is computed [25]. The predicted topics for the focal user will be the topics predicted by a MLP whose input is the representation learned by DeepWalk.
- **LINE**: it is a scalable network embedding approach capturing both the first-order (direct connection within the network) and the second-order (nodes sharing many common neighbors have higher similarities in the embedded space) proximities [26]. The predicted topics will be selected similarly to DeepWalk.
- **node2vec**: performs the 2^{nd}-order random walks to explore neighborhood architecture and embed nodes with the Skip-Gram model [27].
- **struc2vec**: it involves in generating a series of weighted auxiliary graphs, capturing structural similarities among nodes' k-hop neighborhoods, and running node2vec for embedding the auxiliary graphs [28].
- **TADW**: incorporates text features of vertices into network representation learning under the framework of matrix factorization, which can be thought as a heterogeneous embedding method [55].
- **CANE**: learns context-aware embeddings for vertices and models the semantic relationships between vertices. Following [36], we associate the posting content with each note to obtain embeddings, where the text is embedded by a CNN with a mutual attention mechanism and network structure is embedded by LINE [26].

Deep-learning based methods:

- **LRCNN**: is a CNN-based architecture for re-ranking pairs of short texts to learn the representation of text pairs and a similarity function to relate them in a supervised way, which is used here to model the posting similarity [56].
- **GCN-F/GCN-I**: GCN is a semi-supervised classification method based on graph convolution. We derive two topic prediction methods called GCN-F and GCN-I, respectively using posting embeddings and identity matrix as the input feature matrix in GCN [34].
- **MACNN**: is a CNN-based method for next topic prediction, where user's posting history and topics are modeled with an external neural memory architecture and attention mechanism. The relations between users and topics are constructed with a CNN-based matching method [3].

D. METRICS

To evaluate the multi-label classification performance of our model and baselines, we choose standard metrics: macro-Precision (macro-P), macro-Recall (macro-R), macro-F1 and Accuracy, formally defined as:

$$\text{macro-P} = \frac{1}{N}\sum \frac{\#\text{correctly predicted participation topics}}{\#\text{predicted participation topics}}$$

$$\text{macro-R} = \frac{1}{N}\sum \frac{\#\text{correctly predicted participation topics}}{\#\text{ truly participation topics}}$$

$$\text{macro-F1} = \frac{2 \times \text{macro-P} \times \text{macro-R}}{\text{macro-P} + \text{macro-R}}$$

$$\text{Accuracy} = \frac{1}{N}\sum \frac{\#\text{ correctly predicted participation topics}}{\#\text{ truly} \cup \#\text{ predicted participation topics}}$$

where N is the number of users.

E. EXPERIMENTAL RESULTS

Table 4 shows the performance of each method in terms of macro-P, macro-F1 and Accuracy – we omit macro-R due to lack of space, and macro-F1 reflects the harmonic mean of macro-P and macro-R. Overall, we have following observations:

1) OVERALL PERFORMANCE

Our model outperforms other 17 baseline methods across all metrics, e.g., we achieve 2%-22%, 2%-22%, 5%-15% improvement over baselines on *Twitter*, 11%-65%, 13%-62%, 11%-56% on *Weibo*, 1%-23%, 1%-24%, 1%-27% on *Douban*, and 4%-13%, 10%-11%, 3%-6% on *Zhihu* in terms of macro-P, macro-F1 and Accuracy, respectively.

2) COMPARISON DETAILS

Traditional classification methods that only learn the intrinsic preference exhibit inferior performance than deep learning based embedding methods. This phenomenon demonstrates that social relations play an important role on the prediction of which topics individuals will join to discuss.

On one hand, *network embedding methods* that only leverage structural information yield reasonable performance which further proves the effectiveness of incorporating social influence into the framework. Among the NE approaches, we empirically find that two random walk based methods (DeepWalk and node2vec) perform better on capturing network structures. Surprisingly, the most recent NE method struc2vec performs the worst across four datasets. We speculate that its original purpose is to find similar structures in networks and thus is not well-suited to embedding "close" relationships in our problem – assuming that users are influenced largely by immediate friends rather than long-distance neighbors.

On the other hand, two *heterogeneous embedding methods* (TADW and CANE) fail to achieve expected performance and sometimes are even worse than DeepWalk and node2vec (which are purely NE methods). The reason behind is that without specific tunings for applications such as the task

TABLE 4. Performance comparison among different algorithms on four datasets.

Method	Twitter			Weibo			Douban			Zhihu		
	macro-P	macro-F1	Accuracy	macro-P	macro-F1	Accuracy	macro-P	macro-F1	Accuracy	macro-P	macro-F1	Accuracy
LR	0.1721	0.0789	0.0520	0.3912	0.3624	0.3273	0.4244	0.3199	0.2236	0.0766	0.0385	0.0229
DT	0.1742	0.1363	0.0840	0.3028	0.3241	0.2639	0.3820	0.3558	0.2426	0.0713	0.0727	0.0420
RF	0.1992	0.1363	0.0854	0.3562	0.3288	0.2958	0.4039	0.3248	0.2195	0.0485	0.0162	0.0092
SVM	0.1824	0.1243	0.0781	0.3875	0.3804	0.3295	0.3962	0.3394	0.2333	0.0919	0.0766	0.0445
KNN	0.2168	0.1923	0.1165	0.1850	0.1601	0.1476	0.3877	0.2519	0.1630	0.0876	0.0744	0.0432
NB	0.0952	0.0588	0.0388	0.2270	0.2368	0.2023	0.2313	0.2011	0.1563	0.0725	0.0579	0.0371
CF	0.1930	0.1857	0.1117	0.2687	0.2911	0.2299	0.3857	0.2747	0.1786	0.0880	0.0849	0.0484
DeepWalk	0.2916	0.1894	0.1169	0.4492	0.4013	0.3659	0.4531	0.3410	0.2289	0.0934	0.0472	0.0319
LINE	0.2784	0.0889	0.0557	0.4266	0.3705	0.3380	0.4479	0.3841	0.2746	0.0375	0.0119	0.0078
node2vec	0.2690	0.1297	0.0793	0.4238	0.3803	0.3480	0.4478	0.3418	0.2336	0.1062	0.0597	0.0387
struc2vec	0.0856	0.0241	0.0146	0.0203	0.0145	0.0113	0.4056	0.2038	0.1273	0.0688	0.0724	0.0404
TADW	0.2394	0.1270	0.0805	0.4739	0.4208	0.3870	0.4530	0.1582	0.0959	0.0176	0.0070	0.0047
CANE	0.2125	0.0981	0.0586	0.2832	0.2529	0.2299	0.4454	0.3390	0.2449	0.0399	0.0204	0.0138
LRCNN	0.1994	0.1520	0.0952	0.3550	0.3200	0.2923	0.4472	0.3269	0.2261	0.1061	0.0700	0.0441
MACNN	0.2670	0.1129	0.0724	0.5640	0.5016	0.4645	0.4451	0.3782	0.2768	0.0767	0.0245	0.0155
GCN-**F**	0.1544	0.2229	0.1306	0.19999	0.2725	0.1812	0.3848	0.3727	0.2450	0.0707	0.1155	0.0637
GCN-**I**	0.1538	0.2127	0.1239	0.1750	0.2196	0.1534	0.3811	0.3694	0.2405	0.0711	0.1137	0.0631
NTP^3-1D	0.2407	0.2245	0.1362	0.5118	0.4950	0.4320	0.4223	0.3661	0.2477	0.1026	0.0898	0.0521
NTP^3	**0.3134**	**0.2542**	**0.1618**	**0.6751**	**0.6334**	**0.5737**	**0.4636**	**0.3996**	**0.2839**	**0.1421**	**0.1135**	**0.0712**

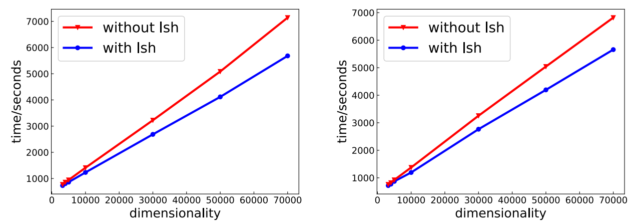
here, the heterogeneous information (posting text and network structure) may "combat" for each other and cannot reach a balance to predict the individual topic preference. This may (partially) show that there is no absolutely good network embedding methods regardless of applications.

Further, the two methods, *LRCNN and MACNN*, specifically tailored for topic participation prediction task are not comparable to our model and sometimes even not comparable to the NE methods such DeepWalk, LINE and node2vec. In fact, LRCNN and MACNN are constructed based on the motivation that personal interest remains consistent for a long time which we also observe in the datasets. However, the results demonstrate that no matter how minor the interest shifts, they can be "glimpsed" and quantified according to the social relationships – which are attributed to advances in network embeddings.

Graph convolutional neural networks for node classification (prediction in our setting), no matter including node features (GCN-*F*) or excluding node features (GCN-*I*), do not exhibit expected performance. The main reason is that classical GCNs are efficient for small and sparse networks and is hard to be generalized to denser network and specific applications such as topic participation prediction. Moreover, the validity of incorporating features in GCN remains elusive as observed in previous works [34], [57].

3) PERFORMANCE DIFFERENCE ACROSS DATASETS

We now turn towards understanding the performance difference among datasets. First, all methods exhibit higher prediction accuracy on *Weibo* data than *Twitter* – the two similar micro-blog sites. A possible reason is that *Weibo* has smaller and more focused number of topics. However, all methods perform well on *Douban* even though *Douban* has the largest number of topics, which increases the difficulty of prediction. It is because *Douban* is a platform where users usually rate and comment on limited topics of their interests, e.g., movies, books, musics, etc. – which could make it easier to profile their specific interests, compared to other sites such as *Twitter* where people may be interested in a broad range of topics from politics to entertainments. Finally, the performance on *Zhihu* is unsatisfying for all methods. The reason is that *Zhihu* is a Q&A website where users usually join to discuss trending topics and(or) topics specific to their domain knowledge while content they posted are mainly from other platforms (refer to as "retweet").



(a) (b)

FIGURE 4. Improvement of efficiency with LSH. (a) Douban Training time. (b) Twitter Training time.

4) EFFICIENCY

Fig. 4 depicts the efficiency improvement due to the hashing operation of *NTP*^{3}, where we purposely increase the dimensionality of (both intrinsic and extrinsic) embeddings. Increasing the dimensionality of embedding would obviously makes neural networks learning inefficient, while binary hash coding can significantly reduce computational cost which has already been illustrated in previous work [45].

5) EFFECTIVENESS OF RESHAPE

To demonstrate the effectiveness of reshaping (1-D to 2-D), we implement NTP^3-1D which is only different from NTP^3 in that it performs the convolutional operations over the 1-D vector. The results illustrate that directly concatenating

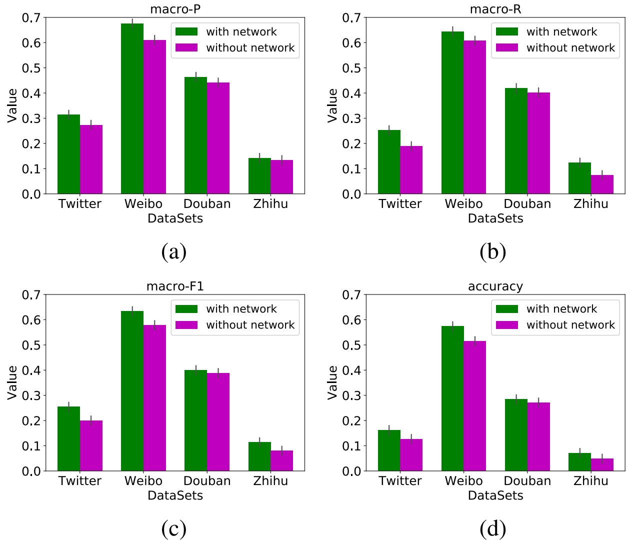


FIGURE 5. Effectiveness comparison with network structure across datasets. (a) macro-P. (b) macro-R. (c) macro-F1. (d) accuracy.

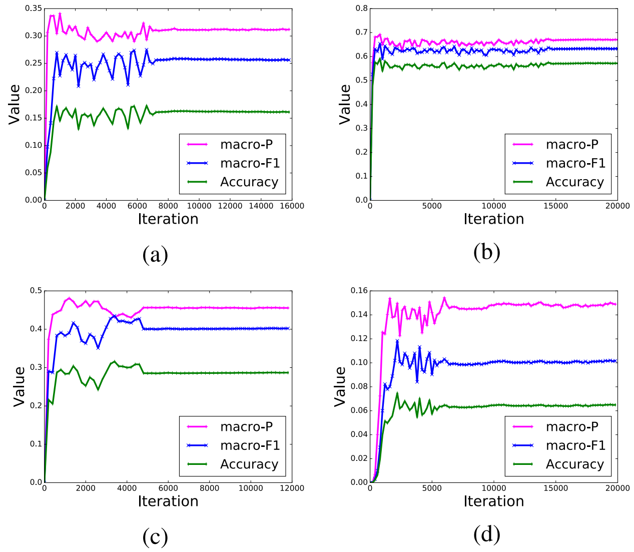


FIGURE 6. Training process of our framework on four datasets. (a) Twitter.com. (b) Weibo.com. (c) Douban.com. (d) Zhihu.com.

of vector representations does not improve, sometimes even deteriorate, the prediction performance.

6) EFFECTIVENESS OF EXTRINSIC PREFERENCE

Fig. 5 depicts the effectiveness of network structure on predicting potential interested topics, which states that incorporating structural information improves the prediction performance on all datasets as expected in our motivation.

Model Convergence: In Fig. 6, we plot the results of macro-P, macro-F1 and accuracy on four datasets along with the training process. We can clearly see that our method converges faster as epoch increases, except on *Weibo* data which has fewest average number of topics a user may participate in.

7) VISUALIZATION

To understand how well LSH captures the latent patterns among data, we compare and visualize embedding vectors in a 2D space with and without LSH, as shown in Fig. 7 where data distribution in (a) and (b) are fairly similar, as well as (c) and (d) – indicating that LSH preserve similarities very well. In addition, similar data after LSH tend to be more closer (their Hamming encodings become identical), which makes patterns embedded in data more salient.

F. MODEL INTERPRETATION

We now turn to interpret the modeling behavior of NTP^3 by investigating the influence of individual training data point on the final predicting performance with the method introduced in IV-F. More specifically, we randomly pick one testing user **u** from both *Douban* and *Weibo*, and compute the influence score of each training point. We then calculate the distance of each data point in the training set to **u** from two aspects: (i) network distance using the number of hops to measure the contribution of extrinsic preference (see Fig. 8(a) and 8(c)), and (ii) hamming distance of binary hashing vectors to measure the contribution of both intrinsic and extrinsic preference (see Fig. 8(b) and 8(d)). From Fig. 8, we observe that: (1) in the nearest neighbor of **u** (1-hop), most immediate neighbors in the training set are positively contributing to predict **u**'s topic participation. The most influential (both positive and negative) instances in the training set are those users with 2-hops and 3-hops away from **u** – note that 3-hops cover almost all (more than 99%) of users in the network. (2) Those data points closer to the user **u** (i.e., smaller Hamming distance (blue points towards left in Fig. 8(b) and 8(d))) are more helpful than nodes further away from **u**.

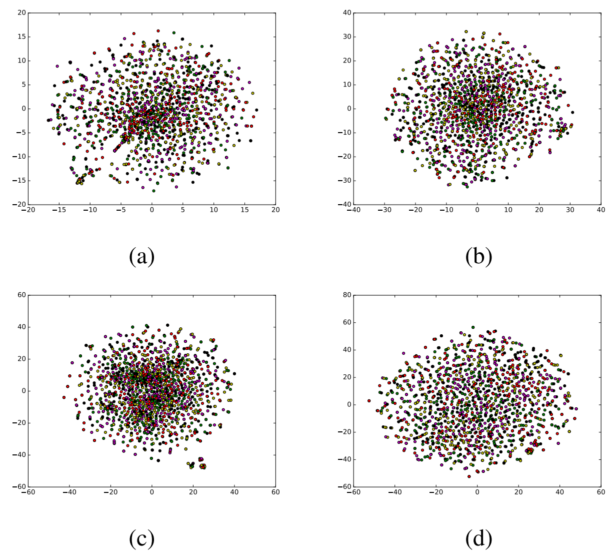


FIGURE 7. Embedding visualization by TSNE. (a) Original tweets embedding. (b) Tweets with LSH. (c) Original network embedding. (d) Network with LSH.

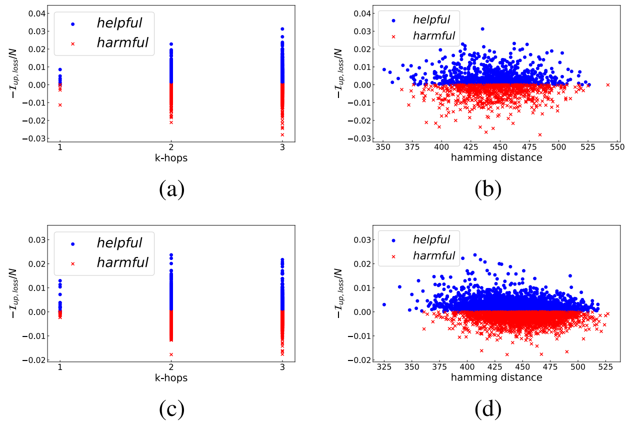


FIGURE 8. Model interpretability. Blue • are helpful training data points that contribute positively to predicting participating specific topics of a certain testing user *u*, while red × denote harmful data points with negative contribution to the learned prediction function of classifying *u* to topic participation. (a) Network distance (Douban). (b) Hamming distance (Douban). (c) Network distance (Weibo). (d) Hamming distance (Weibo).

VI. CONCLUDING REMARKS

We present an effective and efficient framework for predicting the next topics that users are likely to participate in on social media platforms, by modeling/coupling users intrinsic and extrinsic preferences. Historical textual postings and social networked data are combined to jointly learn user topic preferences via embeddings. Locality Sensitive Hashing is used in our framework to transform continuous latent embedding vectors into binary representations. Such a novel design not only makes our prediction efficient, but also well captures latent patterns among data which, in turn, significantly boosts the performance. We also demonstrate the interpretability of our model using influence functions by investigating the importance of each training sample on the topic prediction performance of a testing instance. Incorporating other characteristics, such as user demographics, geo-locations, current trending/emerging topics, etc. into the model might enhance the prediction accuracy. This, along with a more sophisticated model of capturing relationships among users and topics, is the focus of our future work.

REFERENCES

[1] A. Gough *et al.*, "Tweet for behavior change: Using social media for the dissemination of public health messages," *JMIR Public Health Surveill.*, vol. 3, no. 1, p. e14, 2017.

[2] M. Baddeley, "Herding, social influence and economic decision-making: Socio-psychological and neuroscientific analyses," *Philos. Trans. Roy. Soc. B, Biol. Sci.*, vol. 365, pp. 281–290, Jan. 2010.

[3] H. Huang, Q. Zhang, J. Wu, and X. Huang, "Predicting which topics you will join in the future on social media," in *Proc. ACM Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2017, pp. 733–742.

[4] Y. Wang, H. Shen, S. Liu, J. Gao, and X. Cheng, "Cascade dynamics modeling with attention-based recurrent neural network," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2017, pp. 2985–2991.

[5] K. Subbian, B. A. Prakash, and L. A. Adamic, "Detecting large reshare cascades in social networks," in *Proc. World Wide Web Conf. (WWW)*, 2017, pp. 597–605.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013). "Efficient estimation of word representations in vector space." [Online]. Available: https://arxiv.org/abs/1301.3781

[7] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1–11.

[8] H. Liang, Y. Xu, D. Tjondronegoro, and P. Christen, "Time-aware topic recommendation based on micro-blogs," in *Proc. ACM Conf. Inf. Knowl. Manage. (CIKM)*, 2012, pp. 1657–1661.

[9] A. Alsaudi, M. Sadri, Y. Altowim, and S. Mehrotra, "Adaptive topic follow-up on Twitter," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 1385–1386.

[10] S. Gerani, M. J. Carman, and F. Crestani, "Aggregation methods for proximity-based opinion retrieval," *ACM Trans. Inf. Syst.*, vol. 30, no. 4, pp. 1–36, 2012.

[11] Y. Tan, M. Zhang, Y. Liu, and S. Ma, "Rating-boosted latent topics: Understanding users and items with ratings and reviews," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 2640–2646.

[12] S. Zhang and K. Balog, "Ad hoc table retrieval using semantic similarity," in *Proc. World Wide Web Conf. (WWW)*, 2018, pp. 1553–1562

[13] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu, "Collaborative personalized tweet recommendation," in *Proc. ACM Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2012, pp. 661–670.

[14] R. Yan, M. Lapata, and X. Li, "Tweet recommendation with graph co-ranking," in *Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2012, pp. 516–525.

[15] D. Kim, J.-G. Lee, and B. S. Lee, "Topical influence modeling via topic-level interests and interactions on social curation services," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, May 2016, pp. 13–24.

[16] S. Liang, Z. Ren, Y. Zhao, J. Ma, E. Yilmaz, and M. de Rijke, "Inferring dynamic user interests in streams of short texts for user clustering," *ACM Trans. Inf. Syst.*, vol. 36, no. 1, pp. 1–37, 2017.

[17] X. Cheng, X. Yan, Y. Lan, and J. Guo, "BTM: Topic modeling over short texts," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2928–2941, Dec. 2014.

[18] M. X. Hoang, X.-H. Dang, X. Wu, Z. Yan, and A. K. Singh, "GPOP: Scalable group-level popularity prediction for online content in social networks," in *Proc. World Wide Web Conf. (WWW)*, 2017, pp. 725–733.

[19] Y. Chen, H. Amiri, Z. Li, and T.-S. Chua, "Emerging topic detection for organizations from microblogs," in *Proc. ACM Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2013, pp. 43–52.

[20] B. Wu, W.-H. Cheng, Y. Zhang, Q. Huang, J. Li, and T. Mei, "Sequential prediction of social media popularity with deep temporal context networks," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2017, pp. 1–7.

[21] C. Li, J. Ma, X. Guo, and Q. Mei, "DeepCas: An end-to-end predictor of information cascades," in *Proc. World Wide Web Conf. (WWW)*, 2017, pp. 577–586.

[22] Y. Li, J. Fan, D. Zhang, and K.-L. Tan, "Discovering your selling points: Personalized social influential tags exploration," in *Proc. ACM Int. Conf. Manage. Data (SIGMOD)*, 2017, pp. 619–634.

[23] X. Ke, A. Khan, and G. Cong, "Finding seeds and relevant tags jointly: For targeted influence maximization in social networks," in *Proc. ACM Int. Conf. Manage. Data (SIGMOD)*, 2018, pp. 1097–1111.

[24] R. Rotabi, K. Kamath, J. Kleinberg, J. M. Kleinberg, and A. Sharma, "Cascades: A view from audience," in *Proc. World Wide Web Conf. (WWW)*, 2017, pp. 587–596.

[25] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining (SIGKDD)*, 2014, pp. 701–710.

[26] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. World Wide Web Conf. (WWW)*, 2015, pp. 1067–1077.

[27] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining (SIGKDD)*, 2016, pp. 855–864.

[28] F. R. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining (SIGKDD)*, 2017, pp. 385–394.

[29] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining (SIGKDD)*, 2016, pp. 1105–1114.

[30] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52–74, 2017.

[31] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2177–2185.

[32] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec," in *Proc. ACM Int. Conf. Web Search Data Mining (WSDM)*, 2018, pp. 459–467.

[33] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 3844–3852.
[34] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14.
[35] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–15.
[36] C. Tu, H. Liu, Z. Liu, and M. Sun, "CANE: Context-aware network embedding for relation modeling," in *Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2017, pp. 1722–1731.
[37] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. Symp. Theory Comput. (STOC)*, 1998, pp. 604–613.
[38] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proc. World Wide Web Conf. (WWW)*, 2015, pp. 278–288.
[39] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. J. Belongie, and D. Estrin, "Collaborative metric learning," in *Proc. World Wide Web Conf. (WWW)*, 2017, pp. 193–201.
[40] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3270–3278.
[41] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, "Min-wise independent permutations," in *Proc. Symp. Theory Comput. (STOC)*, 1998, pp. 630–659.
[42] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. Int. Conf. Very Large Data Bases (VLDB)*, 1999, pp. 518–529.
[43] M. R. Henzinger, "Finding near-duplicate Web pages: A large-scale evaluation of algorithms," in *Proc. ACM Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2006, pp. 284–291.
[44] B. Bahmani, A. Goel, and R. Shinde, "Efficient distributed locality sensitive hashing," in *Proc. ACM Conf. Inf. Knowl. Manage. (CIKM)*, 2012, pp. 2174–2178.
[45] S. Dasgupta, C. F. Stevens, and S. Navlakha, "A neural algorithm for a fundamental computing problem," *Science*, vol. 358, no. 6364, pp. 793–796, 2017.
[46] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 1188–1196.
[47] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1188–1196.
[48] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein. (2017). "Cayleynets: Graph convolutional neural networks with complex rational spectral filters." [Online]. Available: https://arxiv.org/abs/1705.07664
[49] T. N. Kipf and M. Welling. (2016). "Variational graph auto-encoders," [Online]. Available: https://arxiv.org/abs/1611.07308
[50] M. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proc. Symp. Theory Comput. (STOC)*, 2002, pp. 380–388.
[51] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.
[52] A. Shrivastava and P. Li, "Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS)," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2321–2329.
[53] Q. Huang, J. Feng, and Q. Fang, "Reverse query-aware locality-sensitive hashing for high-dimensional furthest neighbor search," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 167–170.
[54] R. D. Cook and S. Weisberg, "Characterizations of an empirical influence function for detecting influential cases in regression," *Technometrics*, vol. 22, no. 4, pp. 495–508, 1980.
[55] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang, "Network representation learning with rich text information," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2015, pp. 2111–2117.
[56] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *Proc. ACM Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2015, pp. 373–382.
[57] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1024–1034.

FAN ZHOU received the B.S. degree in computer science from Sichuan University, China, in 2003, and the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China, in 2006 and 2011, respectively, where he is currently an Associate Professor with the School of Information and Software Engineering. His research interests include machine learning, spatio–temporal data management, and social network knowledge discovery.

LEI LIU is currently pursuing the M.S. degree with the University of Electronic Science and Technology of China. His current research interests include recommender systems, big data mining, and deep learning.

KUNPENG ZHANG received the Ph.D. degree in computer science from Northwestern University. He is currently a Researcher in the area of large-scale data analysis with particular focus on mining social media data through machine learning, network analysis, and natural language processing techniques. He is also an Assistant Professor with the Department of Information Systems, Smith School of Business, University of Maryland at College Park, College Park. He has published papers in the areas of social media, text mining, network analysis, and information systems in top conference and journals. He serves on the program committees for many international conferences. He is currently an Associate Editor of the *Journal of Electronic Commerce Research*.

GOCE TRAJCEVSKI received the B.Sc. degree from the University of Sts. Kiril i Metodij and the M.S. and Ph.D. degrees from the University of Illinois at Chicago. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Iowa State University. His main research interests include spatio–temporal data management, uncertainty, and reactive behavior management in different application settings and incorporating multiple contexts. His research has been funded by the NSF, ONR, BEA, and Northrop Grumman Corporation. In addition to a book chapter and three encyclopedia chapters, he has co-authored over 110 publications in refereed conferences and journals and has received two Best Paper Awards (CoopIS 2000 and IEEE MDM 2010), the Best Short-Paper Award (ACM MSWiM 2013), and the Best Demo Award (SSTD 2017). He was a General Co-Chair of IEEE ICDE 2014 and a PC Co-Chair of ADBIS 2018 and ACM SIGSPATIAL 2016 and 2017 and has served in various roles in organizing committees in numerous conferences and workshops. He is an Associate Editor of ACM TSAS and *Geoinformatica* journals.

JIN WU received the B.S. degree in automatic control and the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China, in 1993, 1996, and 2004, respectively, where she is currently an Associate Professor. Her research interests include machine learning, knowledge mapping, software development techniques, and process technology.