




A Graph Embedding Based Real-Time Social Event Matching Model for EBSNs Recommendation

Gang Wu^{1,2}(✉) , Xueyu Li¹, Kaiqian Cui¹, Zhiyong Chen¹, Baiyou Qiao¹, Donghong Han¹, and Li Xia¹

¹ Northeastern University, Shenyang, China
wugang@mail.neu.edu.cn

² State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, China

Abstract. Event-based social networks (EBSNs), are platforms that provide users with event scheduling and publishing. In recent years, the number of users and events on such platforms has increased dramatically, and interactions have become more complicated, which has made modeling heterogeneous networks more difficult. Moreover, the requirement of real-time matching between users and events becomes urgent because of the significant dynamics brought by the widespread use of mobile devices on such platforms. Therefore, we proposed a graph embedding based real-time social event matching model called GERM. We first model heterogeneous EBSNs into heterogeneous graphs, and use graph embedding technology to represent the nodes and their relationships in the graph which can more effectively reflect the hidden features of nodes and mine user preferences. Then a real-time social event matching algorithm is proposed, which matches users and events on the premise of fully considering user preferences and spatio-temporal characteristics, and recommends suitable events to users in real-time efficiently. We conducted experiments on the Meetup dataset to verify the effectiveness of our method by comparison with the mainstream algorithms. The results show that the proposed algorithm has a good improvement on the matching success rate, user satisfaction, and user waiting time.

Keywords: EBSNs · Event publishing · Graph embedding · Real-time matching

1 Introduction

With the popularization of mobile Internet and smart devices, the way people organize and publish events is gradually networked. Event-based social networks

Supported by the National Key R&D Program of China (Grant No. 2019YFB1405302), the NSFC (Grant No. 61872072 and No. 61672144), and the State Key Laboratory of Computer Software New Technology Open Project Fund (Grant No. KFKT2018B05).

© Springer Nature Switzerland AG 2020

Z. Huang et al. (Eds.): WISE 2020, LNCS 12342, pp. 41–55, 2020.

https://doi.org/10.1007/978-3-030-62005-9_4

(EBSNs) platforms, such as Meetup¹ and Plancast², provide a new type of social network that connects people through events. On these platforms, users can create or join different social groups. Group users can communicate online or hold events offline, which significantly enriches people’s social ways and brings huge business value through a large number of active users. For example, Meetup currently has 16 million users with more than 300,000 monthly events [9]. Obviously, in order to improve user satisfaction, effective social event recommendation methods are needed to help users select more appropriate events.

However, existing recommendation methods face two challenges in social event recommendation scenarios. On the one hand, as the increase of user and event types and complexity of interactions, traditional graph-based event recommendation algorithms cannot make good use of the rich hidden information in EBSNs. On the other hand, existing event recommendation algorithms are usually designed for those highly planned events rather than scenarios where impromptu events may be initiated and responded immediately. In the paper, recommending impromptu events to users is called “real-time events matching”.

To address the above problems, we proposed a graph embedding based real-time social event matching model for EBSNs recommendation called GERM. It considers not only the rich potential preferences of users, but also the spatio-temporal dynamics of both users and events. For the first challenge, in order to discover more hidden information in the EBSNs, we constructed a heterogeneous information network using various types of EBSNs entities (i.e., user, event, group, tag, venue, and session) and their relations. Then, a meta-path node sampling [16] is performed on the heterogeneous information network to obtain node sequences that retain both the structure information and the interpretable semantics. Hence negative sampling based skip-gram model [11] is used for training the vector representation of different types of nodes in the same feature space by taking previous sampled node sequences as sentences and nodes as words. For the second challenge, we defined the interest similarity matrix, the cost matrix between the user and the event, and a set of matching constraints. Finally, a greedy heuristic algorithm is designed to perform the real-time matching between users and events based on the above definitions. We also designed and implemented a real-time social event publishing prototype system for the experimental comparisons.

In summary, the main contributions of this paper are as follows:

- (1) Analyzed the characteristics of EBSNs. Constructed heterogeneous information network by considering the rich types and relations of entity nodes in the EBSNs. First applied the graph embedding method to event recommendation systems based on EBSNs.
- (2) Proposed a real-time social event matching algorithm for improving users’ satisfaction. According to the real-time user location, time, travel expense, and venue capacity constraints, it dynamically and efficiently matches events for users to maximize the overall matching degree.

¹ <https://www.meetup.com/>.

² <https://www.plancast.com/>.

- (3) We verified the proposed algorithm on the real dataset from Meetup and compared it with the existing mainstream algorithms. The experimental results confirm the effectiveness of the proposed algorithm.

2 Related Work

2.1 Recommendation Algorithms for EBSNs

There are two main categories of recommendation algorithms for EBSNs, i.e., algorithms based on history data and algorithms based on graph model.

Algorithms Based on History Data: These algorithms use the historical data in the system to learn the model parameters, which is used to estimate the user’s selection preferences and make recommendations. Heterogeneous online+offline social relationships, geographical information of events, and implicit rating data from users are widely used in the event recommendation problems, e.g., [10,14]. Lu Chen et al. considered the constraints of users’ spatial information to find highly correlated user groups in social networks more efficiently [2]. Li Gao et al. further considered the influence of social groups to improve the event recommendation performance [6]. I²Rec [5] is an Iterative and Interactive Recommendation System, which couples both online user activities and offline social events together for providing more accurate recommendation.

Algorithms Based on Graph Model: This kind of algorithms model the system as a graph, and take the event recommendation problem as a proximity node query calculation problem. Since the graph embedding technology can represent the structural and semantic information of the nodes in a graph, it has been widely used in these algorithms.

Tuan-Anh NP et al. [13] proposed a general graph-based model, called HeterRS, which considers the recommendation problem as a query-dependent node proximity problem. After that, more recommendation algorithms based on heterogeneous graphs were proposed. Then Yijun Mo et al. proposed a scale control algorithm based on RRWR-S to coordinate the arrangement of users and activities [12]. Lu Chen et al. [1] proposed a new parameter-free contextual community model, which improves the efficiency of community search and the matching effect. The MKASG [3] model considers multiple constraints, which effectively narrows the search space, and then finds the best matching result.

For the recommendation of events, most existing research work focus on exploiting historical data. Literature [7,8,13], and [12] used heterogeneous information networks while they did not use graph embedding techniques to learn the feature vectors of nodes. Considering that the graph embedding technology has been widely used in the recommendation system and achieved great success, we propose to employ it to calculate more meaningful feature vectors of nodes and further construct nodes similarity matrix for better matching results.

2.2 Event Planning

Event planning refers to developing a plan for users to choose events for participate. Yongxin Tong [19], Jieying She [15], Yurong Cheng [4] and others proved that such kind of event scheduling is a NP-hard problem. They respectively proposed greedy-based heuristic method, two-stage approximation algorithm, and other approximate solutions to solve the problem. These methods do not take into account the needs for users to participate in events dynamically in real-time. We focus on designing real-time social event matching algorithm in this paper.

3 Graph Embedding Based Real-Time Social Event Matching Model

3.1 Heterogeneous Information Network of EBSNs

We follow the definition of heterogeneous information network introduced by [13]. It identifies a group of basic EBSNs entity types including *User*, *Event*, *interest Groups*, *Tag*, and *Venues*. Further, a composite entity, namely *Session*, is defined to model the temporal periodicity of a user participating certain event. In this way, it is natural to connect various types of entities through appropriate relations to construct a heterogeneous information network for EBSNs.

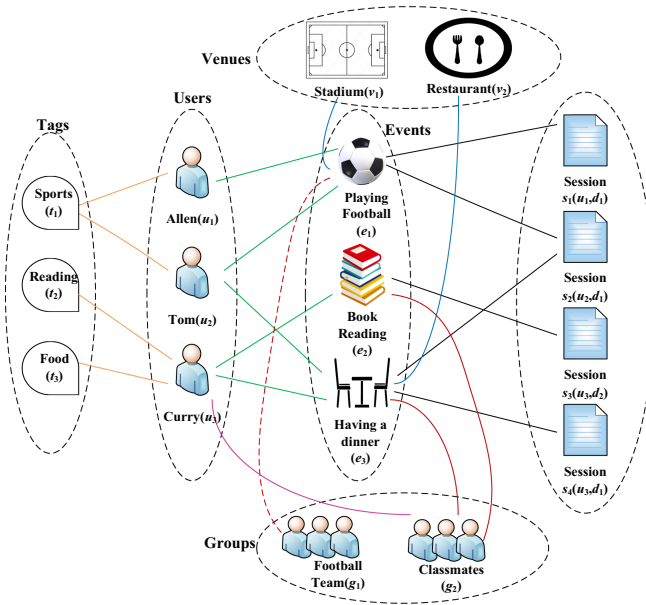


Fig. 1. The heterogeneous information network of EBSNs

Definition 1 (EBSNs Heterogeneous Information Network [13]). Let $U = \{u_1, u_2, \dots, u_{|U|}\}$, $E = \{e_1, e_2, \dots, e_{|E|}\}$, $G = \{g_1, g_2, \dots, g_{|G|}\}$, $T = \{t_1, t_2, \dots, t_{|T|}\}$, $V = \{v_1, v_2, \dots, v_{|V|}\}$, and $S = \{s_1, s_2, \dots, s_{|S|}\}$ be the entity types for describing sets of users, events, groups, tags, venues, and sessions respectively. Let $C = \{U, E, G, T, V, S\}$ be the set of entity types. Let $R = \{\langle U, E \rangle, \langle E, G \rangle, \langle E, V \rangle, \langle U, G \rangle, \langle U, T \rangle, \langle G, T \rangle, \langle E, S \rangle\}$ be the set of relations. EBSNs heterogeneous information network is defined to be a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V} = U \cup E \cup G \cup T \cup V \cup S$ is the node set. And $\mathcal{E} = \{\langle v_1, v_2 \rangle \mid v_1 \in C_1 \wedge v_2 \in C_2 \wedge \{\langle C_1, C_2 \rangle, \langle C_2, C_1 \rangle\} \cap R \neq \emptyset\}$ is the edge set where the entity type $C_1 \in C$ and $C_2 \in C$.

The heterogeneous network can clearly express the user’s interest preference and spatio-temporal preference for events. The detailed description of a sample heterogeneous network is as follows: As shown in Fig. 1, users u_1 and u_2 together participated in the event e_1 held by the group g_1 in the football field v_1 . Both u_1 and u_2 have tag t_1 , indicating that they both like sports. We can also find that u_2 and u_3 together participated in the dinner event e_3 held by group g_2 at venue v_2 on the same date d_1 . Furthermore, session nodes are created for associating date information with events and users. Suppose that events e_1 and e_3 occurred on the same date d_1 , and users u_1 and u_2 together participated in e_1 . Then the session node $s_1(u_1, d_1)$ and $s_2(u_2, d_1)$ are created. At the same time, u_2 participated in event e_3 on d_1 as well. Then the event e_3 is connected to s_2 .

3.2 Feature Vector Representation Method

In order to make full use of the rich information embedded in the heterogeneous information network for event matching, an effective node feature vector representation method is needed.

Meta-Path Sampling. The first step is to sample the heterogeneous information network. Since both the structure of the network and the semantic relationship between nodes are useful, the meta-path sampling method [16] is employed here. The meta-path is defined as “the sequence of relationships between different types of nodes” [16]. Therefore, meta-path based sampling is more interpretable compared with the traditional random walk sampling method, and has been proved to be helpful for mining heterogeneous information networks [4, 17].

Definition 2 (Meta-path Pattern). Given a heterogeneous information network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the meta-path pattern \mathcal{P} is defined as $C_1 \xrightarrow{R_1} C_2 \xrightarrow{R_2} \dots C_t \xrightarrow{R_t} C_{t+1} \dots \xrightarrow{R_{l-1}} C_l$, where $\mathcal{R} = R_1 \circ R_2 \circ \dots \circ R_{l-1}$ is a composite relation defined between the node types C_1 and C_l . And $C_t \in C$ and $R_t \in R$ are the entity type and the relation type under Definition 1.

For example, the relation “ UEU ” indicates that two users have participated in the event together.

Given a meta-path pattern \mathcal{P} and a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, then the transition probability of the meta-path-based sampling is defined as Eq. 1:

$$p(v^{i+1}|v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & \langle v^{i+1}, v_t^{i+1} \rangle \in \mathcal{E}, v^{i+1} \in C_{t+1} \\ 0 & \langle v^{i+1}, v_t^{i+1} \rangle \in \mathcal{E}, v^{i+1} \notin C_{t+1} \\ 0 & \langle v^{i+1}, v_t^{i+1} \rangle \notin \mathcal{E} \end{cases} \quad (1)$$

where $v_t^i \in C_t$, and $N_{t+1}(v_t^i)$ represents the set of neighbor nodes whose type is C_{t+1} , i.e., $v^{i+1} \in C_{t+1}$. Then the next node selected at the current node v_t^i depends on the previously defined meta-path pattern \mathcal{P} .

For the case of social event matching for EBSNs recommendation, we defined a meta-path pattern “*UTUGUESEVESEUGUT*” for sampling. In the pattern, the relation $\langle U, T \rangle$ represents the interest tag chosen by the user. The relation $\langle U, G \rangle$ represents the group to which the user belongs. The relation $\langle U, E \rangle$ represents those events that the user participated in. The relation $\langle E, V \rangle$ represents the venue of the event. And the relation $\langle E, S \rangle$ represents when the user participates in the event.

Therefore, given a heterogeneous information network \mathcal{G} , by applying meta-path sampling with pattern “*UTUGUESEVESEUGUT*” according to Eq. 1, a set of sampled paths will be generated as a corpus for node feature vector representation learning in the next step.

Representation Learning. As we know, graph embedding is a common technique for node feature vector representation. Given a heterogeneous network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the graph embedding based node feature representation for \mathcal{G} is $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ where d is the dimension of the learnt feature vector and $d \ll |\mathcal{V}|$. We expect the dense matrix \mathbf{X} to effectively capture both the structural and semantic information of nodes and relations.

Here, the Skip-gram model based on negative sampling [11] is used to obtain the feature vector that contains the hidden information of the node context. Let the obtained sampled path set by inputting \mathcal{G} and the meta path pattern \mathcal{P} to the meta-path sampling algorithm be θ . In order to apply the negative sampling Skip-gram model, θ is used as a corpus for subsequent representation learning. First, initialize the node feature matrix \mathbf{X} , the iteration termination threshold ϵ , the learning rate η , and the maximum number of iterations *Max.iter*. Second, train each path in the set θ with respect to the skip-gram objective function in Eq. 2.

$$\arg \max_{\theta} \prod_{j=1}^{2c} P(D = 1|v, t_j, \theta) + \prod_{m=1}^M P(D = 0|v, u_m, \theta) \quad (2)$$

where t_j is the j -th positive sample, u_m is the m -th negative sampling sample, θ is the corpus, D indicates whether the current sample is a positive sample. Each node v in the path is taken as a central word whose context with window size c forms a positive sample node set T . And the set of nodes U that do not belong

to the context of v are obtained through negative sampling, which is recorded as a negative sample of the central word v . We maximize the probability of $D = 1$ when taking positive samples and the probability of $D = 0$ when taking negative samples.

Then, we can derive Eq. 3 from Eq. 2:

$$\arg \min_{\theta} -\log \sigma (X_{t_j} \cdot X_v) - \sum_{m=1}^M E_{u_m \sim p(u)} [\log \sigma (-X_{u_m} \cdot X_v)] \quad (3)$$

Where $\sigma(x) = \frac{1}{1+e^{-x}}$, X_v is the v -th row in X , referring to the feature vector of node v , $p(u)$ is the probability of sampling negative samples. The method of gradient descent is used to optimize the objective function, and the iteration is stopped when it is less than the termination threshold ϵ . The above is the method of learning the hidden feature representation of nodes.

Measuring Proximity. In this paper, ‘‘proximity’’ is employed to represent the similarity between nodes in the heterogeneous information network of EBSNs. The proximity is measured by the cosine similarity between two node feature vectors. Therefore, a similarity matrix \mathbf{S} can be constructed for recording the proximity between any two nodes in a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where element S_{ij} is the proximity between node $v_i \in \mathcal{V}$ and node $v_j \in \mathcal{V}$. The greater the value of S_{ij} , the more similarities between the two nodes, and vice versa.

Suppose $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the learnt node feature matrix representation of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. And $X_i \left(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)} \right)$ ($i = 1, 2, \dots, |\mathcal{V}|$) is the i -th d -dimensional feature vector of \mathbf{X} corresponding to node v_i . Then the calculation of similarity between node v_i and v_j is as follows:

$$S_{ij} = \frac{X_i \cdot X_j}{\|X_i\| \cdot \|X_j\|} = \frac{\sum_{k=1}^d X_i^{(k)} \times X_j^{(k)}}{\sqrt{\sum_{k=1}^d \left(X_i^{(k)} \right)^2} \times \sqrt{\sum_{k=1}^d \left(X_j^{(k)} \right)^2}} \quad (4)$$

3.3 Real-Time Social Event Matching

Though the above EBSNs representation learning approach facilitates the evaluation of the similarity between users and events, to make the matching achieve real-time EBSNs status response, the dynamics of constraints on users and events should be effectively expressed.

User Profile. Firstly, as a location-based platform, each EBSNs event can set its own unique location information, while users have their respective location at the time when initiating event participation request. Secondly, in order to avoid time conflicts, users may clearly define their available time intervals for participating in events. Moreover, the travel expense is another important

factor affecting event participation. Therefore, a user is defined to be a quintuple $u(s, t, l, r, b)$, where $u.s$ and $u.t$ respectively represent the earliest start time (lower limit) and the latest terminate time (upper limit) respectively that the user can participate in an event, $u.l$ represents the current location, $u.r$ is the maximum radius that the user can travel, and $u.b$ is the travel budget.

Event Profile. Events usually have limits on the number of participants. Some sports are typical examples, such as basketball usually involves at least 3 to 5 people. Furthermore, due to the limitation of the venue size, the upper limit on the number of participants should be controlled. In addition, it is also necessary to specify the start and end time of each event. In summary, an event is defined to be a quintuple $e(s, t, l, min, max)$, where $e.s$ and $e.t$ represent the start and terminate time of the event respectively, $e.l$ is the location of the event venue, $e.min$ and $e.max$ are the minimum and maximum number of participants allowed by the event.

Matching Constraints. According to the definitions of user profile and event profile, the following constraints are essential to be applied in our real-time social event matching.

- (1) **Budget.** The travel expense of user u_i participating event e_j should not exceed the user's budget, i.e., $\delta_{ij} \leq u_i.b$, where δ_{ij} is the expense that u_i move from his location $u_i.l$ to the event venue $e_j.l$. In Eq. 5, δ_{ij} is calculated based on the Manhattan distance between the two geographic locations where $cost()$ is a given cost function, and (lon_i, lat_i) and $(long_j, lat_j)$ are the coordinate representations of $u_i.l$ and $e_j.l$ respectively.

$$\delta_{ij} = cost(|lon_i - lon_j| + |lat_i - lat_j|) \quad (5)$$

- (2) **Time.** The duration of the event must be within the user's predefined available time intervals so as to guarantee the successful participation, i.e., $u.s < e.s < e.t \leq u.t$ should hold for any matched user u and event e .
- (3) **Capacity.** The number of people participating in an event must meet the constraint on the number of people in the event. Let $\sigma_e = |\{u | \forall (u, e) \in \mathcal{E}\}|$ be the total number of users participating in the event e , then there must be $e.min \leq \sigma_e \leq e.max$.

Problem Definition. Once we have determined the similarity measurement and clarified the necessary constraints, the problem of real-time social event matching can be defined as follow.

Definition 3 (Real-Time Social Event Matching). Let U be the current available user set and E be the current available event set. The real-time social event matching is to assign each $e \in E$ a group of users $U_e = \{u_e^1, u_e^2, \dots, u_e^i, \dots\}$ under constraints (1)–(3) so that the overall matching degree of all current events

can be maximized. In other words, it is to recommend each $u \in U$ a set of events $E_u = \{e_u^1, e_u^2, \dots, e_u^j, \dots\}$ under constraints (1)–(3) for choosing from so that the overall matching degree of all current events can be maximized.

Here, A_{ij} is used to denote the matching degree between user u_i and event e_j , which considers the effects from both the proximity S_{ij} (Eq. 4) and the travel expense δ_{ij} (Eq. 5). Therefore, we have Eq. 6.

$$A_{ij} = \alpha * S_{ij} - \beta * F(\delta_{ij}) \quad (6)$$

where weights $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$, $\alpha + \beta = 1$, and $F(x)$ is a mapping function with value range $(0, 1)$. Hence, the problem of real-time social event matching is a optimization problem as shown in Eq. 7.

$$\begin{aligned} \max \quad & \sum_{e_j \in E} \sum_{u_i \in U} A_{ij} \\ \text{s.t.} \quad & \delta_{ij} \leq u_i.b \\ & u.s < e.s < e.t \leq u.t \\ & e.min \leq \sigma_e = |\{u | \forall (u, e) \in \mathcal{E}\}| \leq e.max \end{aligned} \quad (7)$$

Matching Algorithm. According to the above problem definition of real-time social event matching, we propose a greedy-based heuristic real-time algorithm. The workflow is shown in Fig. 2.

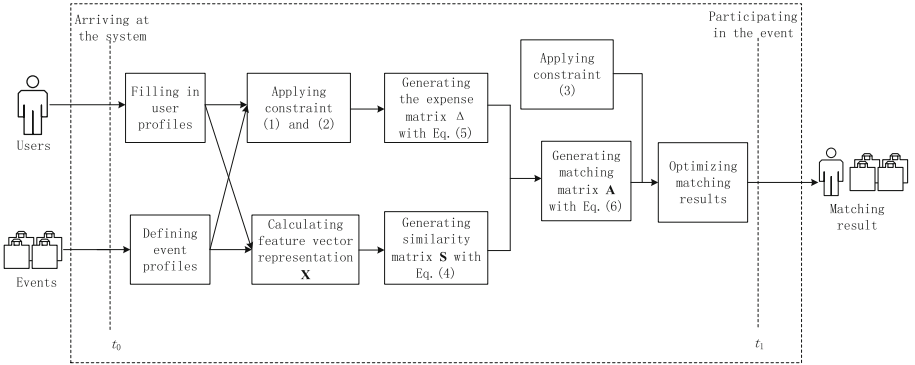


Fig. 2. The workflow of real-time social event matching algorithm

Firstly, using the method described above to calculate the matching degree matrix \mathbf{A} between users and events, and then use the greedy-based heuristic real-time matching algorithm to calculate the final matching results for recommendation as shown in Algorithm 1.

In line 2, a queue Q is initialized to store the currently active events in the system, and a dictionary R is initialized to store the matching results with

users as keys and corresponding sorted candidate events as values. In line 3, events are sorted in ascending order according to their start time, and then stored in Q , which is based on the assumption that events with earlier start time are preferentially matched to users for real-time matching purpose. From line 4 to 15, events are taken from the head of the queue to match with the users in the corresponding column of the matching degree matrix according to the constraints (3), where matched events are stored in R as candidates for future recommendations. Line 6 sorts the columns corresponding to the event e_j in descending order according to the value of the matching degree to ensure that users who meet the constraints (3) are those with a higher matching degree. From line 8 to 14, for an event $e_j \in E$, if the number of users who meet the constraints is greater than $e_j.max$, to maximize the overall matching degree, any user $u_i \in U$ with lower matching degree value A_{ij} should be ignored, so that $e_j.min \leq \sigma_e = |\{u | \forall (u, e) \in \mathcal{E}\}| \leq e_j.max$ is satisfied. Finally, R is returned in line 16.

Algorithm 1. Greedy-based heuristic real-time matching algorithm

Input: user set U , event set E , matching matrix \mathbf{A}

Output: recommended matching for each user $\{u_1 : E_{u_1}, \dots\}$

```

1: function GREEDYMATCH( $U, E, \mathbf{A}$ )
2:   Initialize a queue  $Q$ , a dictionary  $R$ 
3:    $Q \leftarrow \text{sorted}(E)$  by  $e.s$  asc
4:   while  $Q$  is not empty do
5:      $e_j \leftarrow Q.poll()$ 
6:      $\text{sorted}(\mathbf{A}.j)$  by  $A_{ij}$  desc
7:      $\text{sum} \leftarrow 0$ 
8:     for  $u_i \in \mathbf{A}.j$  and  $\text{sum} \leq e.max$  do
9:       if  $u_i$  not in  $R$  then
10:         $R[u_i] \leftarrow \emptyset$ 
11:       end if
12:         $R[u_i] \leftarrow R[u_i] \cup e_j$ 
13:         $\text{sum} \leftarrow \text{sum}+1$ 
14:     end for
15:   end while
16:   return  $R$ 
17: end function

```

After the matching algorithm obtains the final matching result, the matching event lists are recommended to users. Then the user can choose to participate in one of the events. Since all the recommended matching events are to be held within the user's expected participation interval, the proposed algorithm is able to meet the user's real-time event participation requirements, and maximize the overall matching degree of the system as well, i.e., maximizing users' interest preferences and minimizing travel expenses.

4 Experiments and Evaluation

In this section, we show the experiments and results, including the data set description, parameter settings and model comparisons.

4.1 Dataset Description

In order to make the evaluation meaningful, we use a real data set that is from Meetup records of user participation event in California (CA) from January 1 to December 31, 2012 [13]. After preprocessing, the data set consists of users who have participated in at least 5 events, events that have at least 5 participants, and groups that have initiated more than 20 events. See Table 1 for detailed.

Table 1. The details of the dataset

Entity type	Events	Users	Groups	Tags	Venues
Quantity	15588	59989	631	21228	4507

The experiment uses 20,000 users and 2,000 events as the test data set. Due to the lack of user budget, event capacity, location and time information in the profile of user and event in the original data set, the above attribute values were randomly set for each user and event. The start and terminate time attributes were integers generated randomly between 10 and 24. The user’s budget was randomly generated between 0 and 2000. The location coordinate is a pair of numbers between 0 and 2000, which means the locations of users and events appear in a rectangle of 2000×2000 . The minimal number of participants *min* was randomly generated between 5 and 10, while the upper limit *max* was randomly generated between *min* + 5 and 20.

A heterogeneous information network was constructed according to the method in Sect. 3.1. The experiment was conducted in batches of 10 days, and 2,000 users and 400 events were randomly selected from the test data set every day. User nodes were connected to the network according to their tags relations. Event nodes were connected to the network according to their group relations. Every day after applying the matching algorithm, users and events in the matched result were appended to the network. The above steps repeated for the next day.

4.2 Evaluation Criteria

Real-time social event matching technology not only needs to satisfy users’ interest preferences, but also ensures that users can participate in events in a timely manner. For this purpose, the performance of matching algorithms is measured from the following aspects.

1. **User matching success rate.** This value represents the ratio of the everyday number of users who successfully participate in the event to the total number of users. The higher the user matching success rate, the better the algorithm performance.
2. **Event matching success rate.** This value represents the ratio of the everyday number of events successfully held to the total number of events. Similarly, the higher the value, the better the performance of the matching algorithm.
3. **The average matching degree of users,** which represents the ratio of the sum of the matching degrees of users who successfully participated in the events to the total number of successful participants. The higher the value, the better the performance of the matching algorithm. Since the matching degree defined in Eq. 6 involves both the preference similarity and the travel expense satisfaction, this measurement somewhat reflects the ability of the algorithm to reveal implicit feature representations and save expenses.
4. **The average user waiting time,** which represents the average value of the differences between the time t_0 of all users arriving to the system who participating in the event and t_1 of the start time of the corresponding event. Apparently, it is an important indicator for measuring real-time performance of the algorithm. The smaller the average waiting time, the better the performance of the algorithm.

4.3 Performance Comparisons

For the convenience of description, we named our model as *GERM*. The random matching algorithm *Random* and *AGMR* [13] are the baselines for comparing with *GERM*.

Figure 3 shows the experimental results. Doubtless, the Random algorithm has a stable but relative low performance in all aspects.

For *GERM* and *AGMR*, the success rate of user matching is even worse at the very beginning of the experiment because of the lack of data in the network as shown in Fig. 3(a). And hence, since the total number of events remains unchanged (i.e., 2000), the success rate of event matching is not better than that of *Random* as shown in Fig. 3(b). However, as the experiment progressed, the heterogeneous information network became more complicated, which provides richer information to *GERM* and *AGMR* for improving the overall accuracy of matching. Thus, the matching success rate of user and event, and the average matching degree gradually increased until they became stable as shown in Fig. 3(a),(b),(c). Moreover, the result plots of *GERM* was generally above those of *AGMR*. Because *GERM* uses the graph embedding method to obtain node feature vectors and calculate the similarity of nodes, as the data gets richer, it can better mine feature information. Secondly, *AGMR* recommends static event lists for users, but *GERM* can dynamically recommend events for users according to location and time. Figure 3(d) shows the comparisons of the average user waiting time. It can be seen that the *GERM* result curve is always below those

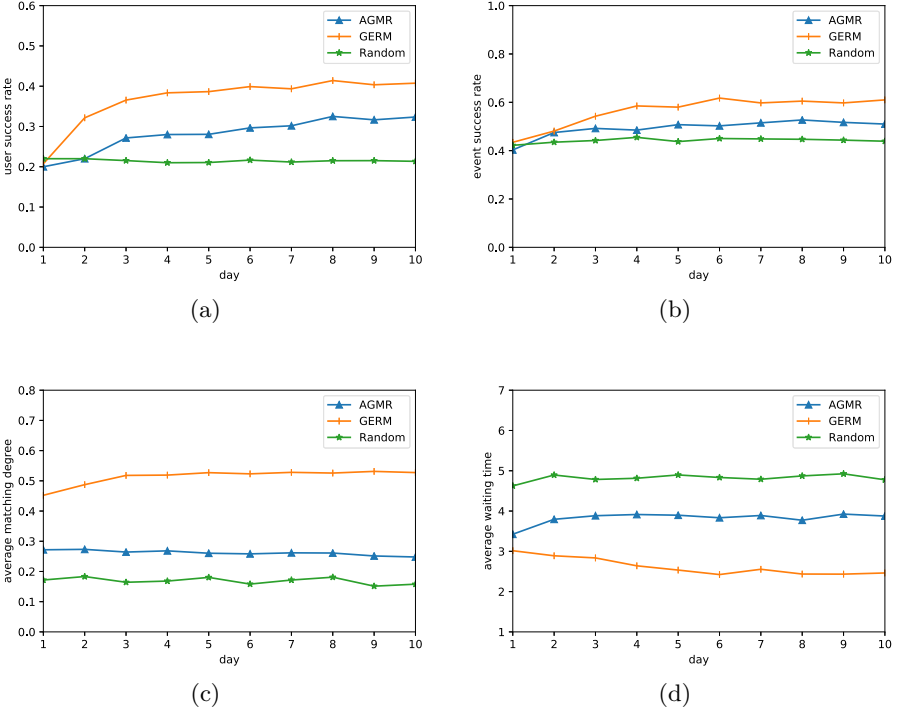


Fig. 3. Model performance comparison result

of AGMR and Random. The average user waiting time is the smallest, which can meet the user’s real-time requirements.

In summary, the model GERM proposed in this paper is superior to the AGMR and Random.

4.4 Discussion on Graph Embedding

We also performed clustering analysis to verify the effectiveness of the graph embedding based feature representation method used in this paper.

The cluster analysis was conducted on the feature vectors of the five types of entities shown in Table 1. Because the original data set is too large to visualize, only 100 samples were randomly selected from each type of data in the experiment. The KMeans algorithm in Sklearn toolkit [18] was used to cluster the extracted samples. The visualization result after clustering is shown in Fig. 4. We can see from Fig. 4(b) that after embedding into feature vectors, entities of the same type will still be clustered together, which shows that the feature vectors generated based on the graph embedding method in this paper can effectively reflect both the structural information and semantic relationship of the original data.

Furthermore, since the experimental data contains five types of entities, we tried to set different number of clusters to calculate corresponding Calinski-

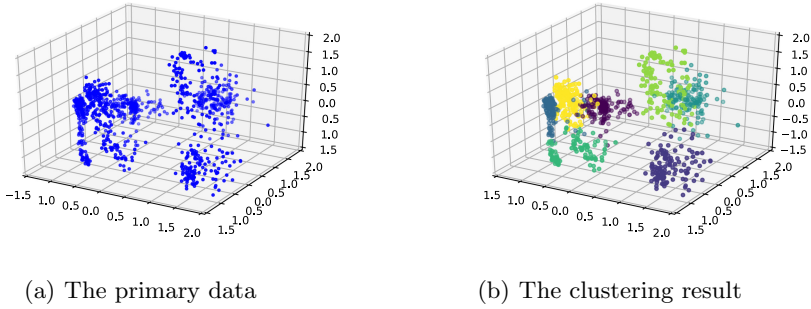


Fig. 4. Performance verification of graph embedding

Harabasz score (CH for short) values. As shown in Table 2, CH becomes the highest when k is 5, and hence the clustering effect is the best. This verifies the effectiveness of the feature representation algorithm used in this paper.

Table 2. The details of data set

k	3	4	5	6	7
CH-score	48.23	50.47	59.72	42.69	39.13

5 Conclusion and Future Work

In this paper, we have proposed a real-time social event matching model based on heterogeneous graph embedding, named GERM. Compared with existing models, our model uses graph embedding method to calculate the feature vector of nodes, which can effectively mine deeper preferences of users' interests and events. Considering the user's preferences and the spatio-temporal characteristics of users and events, to achieve more accurate real-time matching of users and events. Experiment results on Meetup dataset have shown that the proposed algorithm has improvements on the matching success rate, average matching degree, user satisfaction, and user waiting time.

In this study, there are still many aspects worth continuing to improve. As the scale of heterogeneous information network increases, more time will be consumed in the sampling process of node sequences. Therefore, it will be very interesting to explore efficient sampling algorithms. In addition, GERM lacks the analysis of social relationships between users. It will be meaningful to take social relationships into consideration to help users discover more interesting events.

References

1. Chen, L., Liu, C., Liao, K., Li, J., Zhou, R.: Contextual community search over large social networks. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE) (2019)
2. Chen, L., Liu, C., Zhou, R., Li, J., Wang, B.: Maximum co-located community search in large scale social networks. *Proc. VLDB Endow.* **11**, 1233–1246 (2018)
3. Chen, L., Liu, C., Zhou, R., Xu, J., Li, J.: Finding effective geo-social group for impromptu activity with multiple demands (2019)
4. Cheng, Y., Yuan, Y., Chen, L., Giraudcarrier, C., Wang, G.: Complex event-participant planning and its incremental variant, pp. 859–870 (2017)
5. Dong, C., Shen, Y., Zhou, B., Jin, H.: I²Rec: an iterative and interactive recommendation system for event-based social networks, pp. 250–261 (2016)
6. Gao, L., Wu, J., Qiao, Z., Zhou, C., Yang, H., Hu, Y.: Collaborative social group influence for event recommendation, pp. 1941–1944 (2016)
7. Li, B., Bang, W., Mo, Y., Yang, L.T.: A novel random walk and scale control method for event recommendation. In: 2016 International IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCOM/IoP/SmartWorld) (2016)
8. Liu, S., Bang, W., Xu, M.: Event recommendation based on graph random walking and history preference reranking. In: International ACM SIGIR Conference (2017)
9. Liu, X., He, Q., Tian, Y., Lee, W., Mcpherson, J., Han, J.: Event-based social networks: linking the online and offline social worlds, pp. 1032–1040 (2012)
10. Macedo, A.Q.D., Marinho, L.B.: Event recommendation in event-based social networks. In: 1st International Workshop on Social Personalisation (SP 2014) (2014)
11. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space (2013)
12. Mo, Y., Li, B., Bang, W., Yang, L.T., Xu, M.: Event recommendation in social networks based on reverse random walk and participant scale control. *Future Gener. Comput. Syst.* **79**(PT.1), 383–395 (2018)
13. Pham, T.A.N., Li, X., Gao, C., Zhang, Z.: A general graph-based model for recommendation in event-based social networks. In: 2015 IEEE 31st International Conference on Data Engineering (2015)
14. Qiao, Z., Zhang, P., Cao, Y., Zhou, C., Guo, L., Fang, B.: Combining heterogenous social and geographical information for event recommendation, pp. 145–151 (2014)
15. She, J., Tong, Y., Chen, L.: Utility-aware social event-participant planning, pp. 1629–1643 (2015)
16. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: meta path-based top-k similarity search in heterogeneous information networks. *Proc. VLDB Endow.* **4**(11), 992–1003 (2011)
17. Sun, Y., Norick, B., Han, J., Yan, X., Yu, P.S., Yu, X.: Pathselclus: integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Trans. Knowl. Discov. Data* **7**(3), 11 (2013)
18. Swami, A., Jain, R.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**(10), 2825–2830 (2012)
19. Tong, Y., Meng, R., She, J.: On bottleneck-aware arrangement for event-based social networks, pp. 216–223 (2015)