*electronics*

MDPI

*Article*

# A Survey of Context-Aware Recommendation Schemes in Event-Based Social Networks

**Xiaomei Huang [1,2], Guoqiong Liao [1,*], Naixue Xiong [3], Athanasios V. Vasilakos [4,5,6] and Tianming Lan [1]**

[1] School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330013, China; huangxm501@126.com (X.H.); ltm_30@163.com (T.L.)
[2] School of Mathematics and Statistics, Jiangxi Normal University, Nanchang 330022, China
[3] Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA; xiong31@nsuok.edu
[4] School of Electrical and Data Engineering, University of Technology Sydney, New South Wales, NSW 2007, Australia; th.vasilakos@gmail.com
[5] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China
[6] Department of Computer Science, Electrical and Space Engineering, Lulea University of Technology, 97187 Lulea, Sweden
* Correspondence: liaoguoqiong@163.com

Received: 16 August 2020; Accepted: 24 September 2020; Published: 27 September 2020

**Abstract:** In recent years, Event-based social network (EBSN) applications, such as Meetup and DoubanEvent, have received popularity and rapid growth. They provide convenient online platforms for users to create, publish, and organize social events, which will be held in physical places. Additionally, they not only support typical online social networking facilities (e.g., sharing comments and photos), but also promote face-to-face offline social interactions. To provide better service for users, Context-Aware Recommender Systems (CARS) in EBSNs have recently been singled out as a fascinating area of research. CARS in EBSNs provide the suitable recommendation to target users by incorporating the contextual factors into the recommendation process. This paper provides an overview on the development of CARS in EBSNs. We begin by illustrating the concept of the term context and the paradigms of conventional context-aware recommendation process. Subsequently, we introduce the formal definition of an EBSN, the characteristics of EBSNs, the challenges that are faced by CARS in EBSNs, and the implementation process of CARS in EBSNs. We also investigate which contextual factors are considered and how they are represented in the recommendation process. Next, we focus on the state-of-the-art computational techniques regarding CARS in EBSNs. We also overview the datasets and evaluation metrics for evaluation in this research area, and discuss the applications of context-aware recommendation in EBSNs. Finally, we point out research opportunities for the research community.

**Keywords:** context-aware recommendation; event-based social networks; recommender systems; contextual factors; computing techniques

## 1. Introduction

With the popularity of social networks and developments of wireless technology [1–3], event-based social networks (EBSNs) platforms, such as Meetup (www.meetup.com), Plancast (www.plancast.com) and Douban Event (www.douban.com), are growing up quickly in recent years. They are changing the people's way of life, leisure, and entertainment. The main goal of such platforms is facilitating online users to create, distribute, organize, manage, and register offline social events, so as to help users with the same interests participate in their common interested events. The events here could be formal activities, like academic meetings and business exhibitions, or informal get-togethers, like dining out and movies night.

EBSNs generate a large number of new events and new online groups every day. For example, till July 2020, Meetup.com has more than 44 million users and 330,000 groups in 190 countries and 2000 cities worldwide, and more than 84,000 Meetup events are held per week [4]. Through EBSNs, event organizers hope to recruit more participant, while ordinary users expect to attend more interesting events. However, faced with the massive information in EBSNs, it becomes increasingly difficult for users to find their favorite events or online groups. Therefore, it is necessary to study effective approaches to recommender systems in EBSNs.

In recent years, Context-Aware Recommender Systems (CARS) have become one of the most active research areas in recommender systems in EBSNs. It is important to incorporate the contextual information into the recommendation process. For example, when incorpating the temporal context, an outdoor meetup event recommendation in the winter will be different from the one in the summer. In the winter, the system may recommend users to go skiing, while, in the summer, the system will recommend them to go swimming or rafting. There are mainly two advantages for CARS in EBSNs: firstly, the recommendation task in EBSNs faces a serious cold-start problem, which is, events in EBSNs have short life time and candidate events are new events that have little or no trace of historical attendance. Additionally, there only exists implicit feedback information, i.e., users do not rate the events explicitly. Traditional approaches, like collaborative filtering methods, are not suitable for this scenario. Incorporating the contextual information, such as time, location into recommending process helps to alleviate the cold-start problem. Secondly, users make their decisions under certain contextual circumstances, so considering relevant contextual information when making recommendations helps to acquire more accurate prediction of the users' preferences.

In this paper, we provide a literature review of papers published since 2012 regarding CARS in EBSNs. The concrete objectives of this paper are as follows:

- Identify the contextual factors and summarize their representation methods;
- Classify the techniques for incorporating contextual factors to make recommendations in EBSNs;
- Scrutinize the datasets and the major methods used for evaluating the CARS in EBSNs;
- Summarize the specific applications of the context-aware recommendation approaches in EBSNs; and,
- Point out the promising future directions in this research area.

The rest of the paper is organized, as follows. In Section 2, we give a quick overview of the concept of context and the approaches used in conventional CARS. In Section 3, we first scrutinize contextual factors that are used for CARS in EBSNs and investigate how they are modeled in these works. Subsequently, we overview the techniques used for adopting contextual information to make recommendations. In Section 4, we discuss the selection of datasets and the evaluation metrics used for CARS in EBSNs. Next, in Section 5, we introduce the applications of context-aware recommendation in EBSNs. In Section 6, we point out the future research directions to guide the follow-up work and Section 7 is the conclusion of the survey.

## 2. Background

### 2.1. Context-Aware Recommender Systems

Because of the complexity and the broadness of the context concept, it has no unique definition across different disciplines. One widely used definition is given by Dey, as follows [5]: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, a place, or an object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". Contexts can be classified into representational contexts and interactional contexts [6]. Representational contexts are defined as a predefined set of observable attributes, the structure of which does not change significantly over time. In contrast, interactional contexts are assumed to be underlying and unobservable. The structure of interactional context may change over time. Additionally, there exists a bidirectional relationship between users' activities and interactional contexts, which means that they will influence with each other.

The conventional recommender systems work on estimating the rating function $R$: $User \times Item \rightarrow Rating$, once the function is estimated, the highest-rated items will be recommended for each user. CARS try to incorporate contextual information into conventional user-item space and its main task is to predict the rating function $R$: $User \times Item \times Context \rightarrow Rating$. Because the search space is muti-dimensional, it becomes computationally expensive. The challenge of CARS is to acquire user preferences in different contextual situations.

According to the phase when contextual information is incorporated, a context-aware recommendation process can take one of the following three paradigms [7]:

(1) Contextual pre-filtering. In this paradigm, contextual information is used for data selection or data construction. Subsequetly, ratings can be predicted using any traditional Two-dimensional (2D) recommender system on the selected data.

(2) Contextual post-filtering. In this paradigm, contextual information is initially ignored, and any traditional 2D recommender system could be used on the entire data to predict the ratings. Subsequently, the recommendation result is adjusted by using the contextual information.

(3) Contextual modeling. In this paradigm, contextual information is directly incorporated in the recommendation model as an explicit predictor of a user's rating for an item.

### 2.2. Context-Aware Recommender Systems in EBSNs

The definition of an EBSN is first given by Liu et al. in [8], which is, as follows: an EBSN is a heterogeneous network $G =< \mathcal{U}, A^{on}, A^{off} >$, where $\mathcal{U}$ represents the set of users (vertices) with $|\mathcal{U}| = n$, $A^{on}$ stands for the set of online social interactions (arcs), and $A^{off}$ denotes the set of offline social interactions (arcs). The online social interactions of an EBSN form an online social network $G^{on} =< \mathcal{U}, A^{on} >$, and the offline interactions of an EBSN compose an offline social network $G^{off} =< \mathcal{U}, A^{off} >$. Figure 1 shows the structure of an EBSN. In the figure, the events could be informal get-togethers, such as attending cocktail parties, going on a picnic, playing football, and watching movies, and they could also be formal activities, such as technical conferences and business meetings.

Note that the offline events are the events that will be held in physical places. To facilitate people organize or attend offline events, the event-based online social services provide online platforms. Through the platforms, organizers could create and publish events, and interested users could register these events and then meet with each other face-to-face in physical places. Therefore, the online events are the same as the offline events.

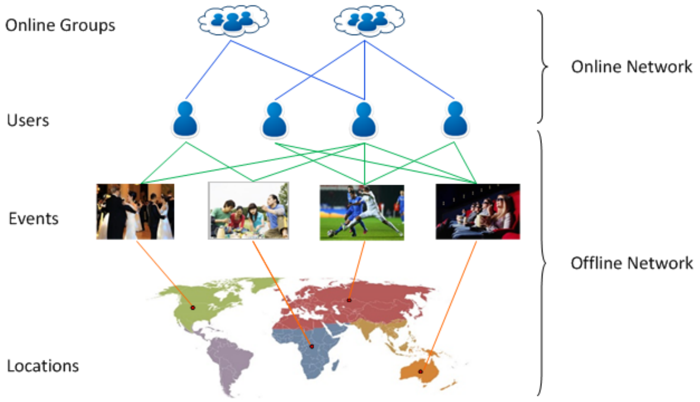

**Figure 1.** Structure of an event-based social network (EBSN).

The entities in EBSNs include users, events, and online groups. Different from other social networks, EBSNs have following characteristics:

- Implicit feedback. In EBSNs, there are no explicit ratings provided by users. Users express their willingness to participate in an event by RSVP with "yes" (RSVP is the French expression which means "please respond"). Because users who reply with "yes" are more likely to participate in the event than those who reply with "no" or do not answer, most of the existing works take the users with "yes" reply as the users who participated in the event offline.
- Short life time. Different from traditional items, like books, music, or movies, events in EBSNs have short life cycle. Once an event started or finished, it makes no sense to recommend it to users. Event recommendation is only valid after the event is created and before the event starts.
- Regular spatio-temporal patterns. Liu et al. [8] found that the occurrence of events shows a regular temporal pattern. For example, in every weekday, there is a small spike around 14:00 in the afternoon, followed by a higher spike at 20:00 in the evening; events on weekends are relatively evenly distributed throughout the day; events are mainly located in urban areas.
- Participation in groups. In EBSNs, users tend to participate in offline events together as a group [9]. For example, people often meet up to go to movies, take part in sports, or attend concerts. Therefore, groups of users become an important target for event organizers to be invited in order to participate in events.
- Diverse contexts. An EBSN contains a variety of context information, such as event context, user context, online group context, etc. For example, event context includes textual description, event topic, start time, geographic location, etc.; group context includes group label, semantics, etc. Besides, there also exist social contexts between users and groups. These rich context information provides effective support for EBSN recommendation.
- Multiple relations. There are various types of entities in EBSNs, such as online groups, users, events, locations, and hosts, etc. There exist multiple relations between these entities.

By employing the abundant contexts in EBSNs, CARS can help to find users' preference for items accurately, alleviate the data sparsity problem that is caused by insufficient user feedback, and address the new event cold start problem that is brought by the short life time of events. However, CARS in EBSNs are faced with several challenges, as follows:

- Mining different types of contexts. There are abundant contexts in EBSNs. Although some types of contexts, such as time and location, have been considered in the recommendation, more types of contexts that may have impacts on users' decision are to be discovered.
- Measuring the influences of contexts. Different types of contexts have different impacts on users' preferences. For example, the context of companion may be more important than the context of weather in a user's decision on watching a movie. Effective approaches need to be developed to measure the influences of various contexts.
- Incorporating contexts in the recommendation process. A traditional recommender system has a data record of the form <user, item, rating>. In contrast, CARS have the record of the form <user, item, context, rating>, where context is an additional dimension and may consist of any number of contexts. There are different methods of incorporating contexts in the process of recommendation. Additionally, dimensionality reduction is an issue that needs to be addressed.

The existing context-aware recommendation process in EBSNs generally takes the third paradigm, i.e., contextual modeling paradigm. The process includes the following four components:

(1) Collecting contexts: contextual information can either be explicitly introduced by the user, or implicitly be collected from the sensors of the user's device or from other sources;

(2) Representing contexts: there are different methods to represent the collected contextual data. One of the widely used representations is the hierarchical model, which organizes the contexts by hierarchical structures like trees;

(3) Incorporating contexts: how to incorporate context into the recommendation process depends on the computational techniques utilized. In this process, the key problem is to identify the user's preferences which is sensitive to the change of contexts; and,

(4) Making recommendation: the most suitable recommendation results are proposed to the target user in a given contexts considering the history of contextual preference of the user. Figure 2 shows the process of context-aware recommendation in EBSNs.

In Figure 2, the rectangular vertices represent the components that we mentioned above. The cylindrical vertex labeled "contextual information" represents the database that contains the context data in EBSNs. The cylindrical vertex labeled "User feedbacks for items" represents the database that contains the feedback data of users for the items they interacted in the past. The vertex labeled "Recommendation results" represents the recommended list of items for the target users.

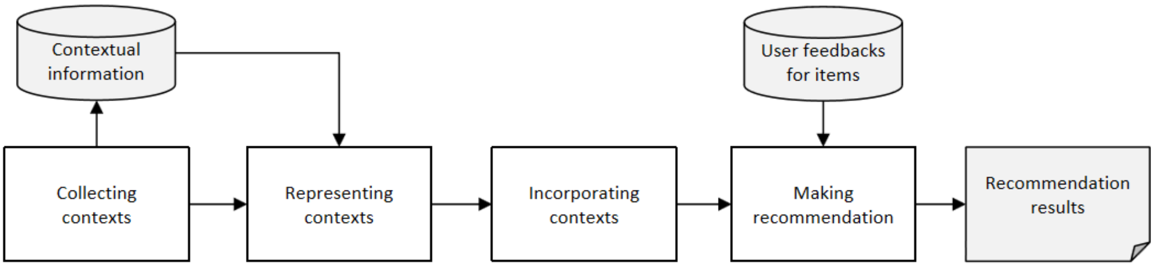

**Figure 2.** Process of context-aware recommendation in EBSNs.

## 3. Context-Aware Recommendation Models in EBSNs

### *3.1. Contextual Factors Used*

In this section, we summarize the different types of contexts that are used to enhance recommendations in EBSNs and how they are extracted and represented in recommendation models in the previous works. The major contextual factors include text content factor, temporal factor, spatial factor, and social factor.

#### 3.1.1. Text Content Contextual Factor

The content of an event is usually described in the form of a textual document. Usually, the content documents of all events are gathered to constitute a corpus, and the natural language processing techniques are utilized in order to map each word in the corpus into a vector. The most widely used techniques in literature are: Term-Frequency-Inverse-Document-Frequency (TF-IDF) [10], Latent Dirichlet Allocation (LDA) [11], Glove [12], and Convolutional Neural Network (CNN) [13]. Next, we review the related work to understand how to model the text content factor.

Gu et al. [14] extracted content feature from the introductions of events by using TF-IDF method, which represents each event by a vector. A user is represented by a vector by summing the vectors of all the events that (s)he has attended. Additionally, a user's content preference for an event is measured by computing the cosine similarity between the user vector and event vector.

TF-IDF is a traditional count-based embedding method that creates word vectors that have the same length as the size of the vocabulary. One of its limitations is that it results in high-dimensional vectors that redundantly encode similar information along many dimensions. Learning-based embedding methods [15], such as Glove [12], Continuous Bag-of-Words (CBOW) learn word vectors that have a much lower dimension, and the representation vectors, are learned by maximizing an objective for a specific learning task, for example, predicting a word based on texts. In [16], the textual description for an event is represented as word embedding vectors by the Glove model, which has proven to be computationally efficient as compared to count-based vectors.

With the assumption that the users' preferences on future events rely on underlying topics rather than word descriptions, Du et al. [17] used the LDA technique to represent the textual document of each event as a vector, which represents the probability distribution over latent topics. The idea of LDA is that a document is the mixture of topics, where a topic is a probability distribution over words. Subsequently, the content similarity between two events could be computed based on the Jensen–Shannon (JS) distance [18]. Additionally, the content preference of a user to a candidate event is obtained by considering the attenuation degree of the user's interests and the content similarity between the candidate event and each event that the user has attended in the past. Yuan et al. [19] also used the LDA to learn topics from the textual content of events. Additionally, the similarity between two events is measured by the standard cosine similarity between the vectors of the two events.

Du et al. [20] noticed that the classic topic model LDA may not infer topics with high quality from sparse corpuses, where each document is very short. Additionally, they also found that the events held by the same organizer may have more similar content than those that are held by distinct organizer. Therefore, they extended the classic topic model to discover content topics from both organizers and textual content of events in order to alleviate the sparseness problem of textual content.

In recent years, deep learning techniques have been used to effectively exploit contextual features from contents. Wang et al. [21] utilized CNN to capture the contextual information of description documents of events and integrated it with Probabilistic Matrix Factorization (PMF) model for accurate rating prediction. CNN is well suited to detecting spatial substructure and creating meaningful spatial substructure as a result.

Different from the above methods, Yin et al. [22] represented the content textual document of an event as a set of word nodes in an event-word bipartite graph. For each content word, there is an edge between the event and the word. Additionally, the standard TF-IDF is used to compute the edge weight. Subsequently, a graph-based embedding algorithm is developed in order to embed the bipartite graph into a low-dimension latent space, in which each event node is represented by a low-dimension vector, which captures the event's semantic information.

### 3.1.2. Temporal Contextual Factor

The temporal contextual factor that is considered in the literature in EBSNs is related to the start time of an event. Because the crawled data on these temporal contextual information take the form of continuous timestamps, it is necessary to discretize them beforehand, so that they can be mapped into the feature vectors conveniently. A frequently used method is to map the timestamps into discrete time slots. We overview the related methods in the literature, as follows.

Du et al. [17] found that the users' behavior shows strong daily and weekly periodical patterns. Therefore, they introduced two temporal factors: the day of the week factor and the hour of the day factor. The day of the week factor represents which day of the week when the user attended an event. Additionally, the day of the week preference of a user is computed by the sum of content similarity between the candidate event and each past event of the user that is held on the same weekday as the candidate event. The hour of the day factor represents which hour of the day when the user attended an event. Similarly, the hour of the day preference of a user is also computed by a weighted sum of content similarity between the candidate event and each past event of the user, but the weight is computed by utilizing a Gauss formula and the weight value is inversely proportional to the time interval between the two events.

Macedo et al. [23] assumed that users who attended events in the past at certain days of the week and at certain hours of the day will likely attend events with a similar temporal profile in the future. Accordingly, they represented each event as a $7 \times 24$ dimensional vector in the space of all possible days of the week and hours of the day, with a vector component set to one whenever the event happened at that particular day and hour. Additionally, each user is represented as the average of the events that (s)he attended in the past. The temporal score for a user and a candidate event is computed as the cosine between their vector representations.

Gu et al. [14] represented each event as a $7 \times 4$ dimensional vector in the space of all possible days of the week and time regions of the day. The four time regions are morning, afternoon, evening, and night. Accordingly, each user is represented as the average vector of the events that (s)he has attended in the past. Additionally, the temporal contextual feature is computed by the time similarity between the user and the event, which is based on the cosine function.

Yin et al. [22] defined 33 time slots according to three different temporal periodicity scales, i.e., hours of a day, days of a week, and weekday types. The 33 time slots includes 24 h slots, seven day slots, and two time slots that indicate weekday and weekend. For example, "2020-07-31 20:00" corresponds to three time slots: 20:00, Friday, and weekday. Because they adopted a graph-based embedding model, the above time slots are denoted as 33 time nodes in an event-time bipartite graph. Additionally, if an event is held at time t, then there are edges between the event node and the corresponding three time slot nodes. By utilizing the graph-based embedding algorithm, each time slot node will be embedded to a low-dimension vector.

Wang and Tang [24] also divided the timestamps to several time slots. Furthermore, the recommendation results of using various temporal patterns are compared in their experiments. There are four total temporal patterns compared, which are weekday-hour (e.g., three (day of the week), 2:00–3:00), day-hour (e.g., 21 (day of the month), 2:00–3:00), month-weekday-hour (e.g., 3 April (day of the week), 2:00–3:00), and month-day-hour patterns (e.g., 21 April (day of the month), 2:00–3:00). The result of experiments indicates that the weekday-hour pattern achieves the best performance, which indicates that people's behaviors exhibit stronger temporal cyclic patterns in a week than in a month.

Different from above works, Pham et al. [25] employed the multivariate Markov chain model in order to solve the recommendation problems in EBSNs. As for temporal factors, they considered the time duration (in days) between two successive events of each user. They found that most users take part in events with a weekly periodic schedule (i.e., weekly periodic patterns). To exploit the patterns of user behaviors, they constructed a new type of nodes, called session node. Specifically, if a user $u$ has joined an event in day $d$ of a week (e.g., Saturday), then a session node $s(u, d)$ is created and linked to that event. Subsequently, session nodes are employed in order to exploit the correlations between events. For example, events that are connected to the same session node are considered to be similar to each other. In addition to that, the authors believe that recent events tend to be more important than earlier ones, and theu defined the weight of an event to be reversely proportional to the time difference (in days) between the event and the last event.

### 3.1.3. Spatial Contextual Factor

Incorporating the influence of spatial contextual context has been a hot topic in the research of recommendation systems [26,27]. The crawled location data of events takes the form of geographic coordinates, denoting where the events are held. We overview the related methods, as follows.

Du et al. [17] noticed that the likelihood of a user attending an event decreases as the distance between the user's location and the event's location increases. Therefore, the user's spatial preference for a candidate event is computed by the weighted sum of the content similarity between the candidate event and each past event of the user, where the weight value is inversely proportional to the distance between the two events.

Qiao et al. [28] found that most of the events were located around centers and concentration areas. They used the $k$-means algorithm to cluster all events to obtain the $k$ regions, where the geographical feature of each event is denoted as a binary variant (latitude, longitude). Subsequently, the Gaussian distribution is used to model the relationship between the events and the regions. Additionally, the probability of an event belonging to a region could be computed. Lastly, the probability is employed as the weight value to compute the weighted user rating w.r.t. each region.

Gu et al. [14] defined two kinds of spatial contextual features: distance between user and event; location preference. The geographical distance of the home location of the user and the location of the event is computed by the widely used metric "the great-circle distance", which is the shortest distance on the surface of the Earth. The location preference of a user is the weighted sum of the distance between the event and each past event attended by the user.

Yin et al. [22] transformed the continuous spatial information into a set of discrete regions using DBSCAN (Density-Based Spatial Clustering of Applications with Noise) based on the geographic coordinates of events. Additionally, an event-location bipartite graph is constructed, in which an event node and a region node are connected if the event is held in the region. Subsequently, a graph-based embedding algorithm is employed in order to embed each region node to a low-dimension vector.

Wang and Tang [24] split the city into even grid cells according to coordinates, and each grid that spans 0.13 km corresponds to a location. Subsequently, the location of each event is represented by a one-hot vector, which will be combined with other feature vectors to be fed into a neural network in order to predict the users' preferences.

Macedo et al. [23] used a kernel-based density estimation approach in order to model the mobility patterns of users as distributions of geographic distances between the attended events. The geographic preferences of a given user are then represented by the sum of all Gaussian distributions centered at each lan-long coordinate of event. Aditionally, the candidate events are scored based on their distances to the events attended by the target user in the past.

Du et al. [20] employed a probability generative model to model the generative process of a group. To model the influence of spacial contextual factor, the model associates each venue with venue topics, which indicates some latent features of venues, such as the ticket price facility and capacity. Specifically, a venue topic in CVTM is represented by a multinomial distribution over venues and used to model group members' interests on venues.

Pramanik et al. [16] obtained two types of information associated with each venue: qualitative information, which includes reviews and tips posted for the venue; quantitative information which indicates the venue category and services that are available at that venue (such as WiFi, parking). The qualitative information of a venue is modeled as a vector by using the Glove tool. The quantitative information of a venue is represented by concatenating the one-hot vectors of category and services of the venue. Subsequently, the two types of vectors are respectively passed through a deep learning module that embeds them into a latent embedding space. Finally, the two lower dimension latent representations of qualitative vector and quantitative vector are concatenated to obtain a unified representation for the venue.

### 3.1.4. Social Contextual Factor

Social contexts refer to the various relationships among users. They can be the relationship between event hosts and the event participants, the friendships between users, or the memberships between a group and its members. We overview the related works, as follows.

Du et al. [17] focused on the influence of event hosts in Douban Events. The authors believed that the hosts usually play much more important roles in a user's attendance decision than ordinary followers, because the hosts can recommend event to its followers and users tend to attend events hosted by an influential host. The authors defined two types of social relationships between the user and the host: following relationship which represents whether the user follows the host; preferring relationship which represents whether the user attended the events that were hosted by the host. Finally, the user's social preference for a candidate event is computed by the weighted sum of the content similarity between the candidate event and each past event of the user, where the weight value is computed based on the above relationships.

Qiao et al. [28] modeled the heterogeneous social relationship between users. The offline relationship between users is modeled based on the co-participation of events. The more events they attended, the closer their relationship is. Additionally, they also modeled the online relationship between users based on the co-participation of online social groups. Similarly, the more groups that they participated, the closer they became. Because the author employed the matrix factorization model for recommendation, the influence of social factor is modeled as the social regularization term of the objective function, which assumes that the preference of a user is close to the weighted average preference of his friends. Additionally, the strength of online and offline social relationships is used to compute the above weight.

Macedo et al. [23] incorporated the influence of groups, which promotes the events. They considered two kinds of relations: user-group relations; group-event relations. User-group relations denote the interactions between users and the groups they are affiliated to. Additionally, group-event relations denote the interactions between groups and events created by them. By utilizing the two kinds of relations conjointly, users that are affiliated to the same or similar groups are prone to attend the same events created by these groups. Additionally, the Multi-Relational Factorization with Bayesian Personalized Ranking method is employed in order to model these relations. Once the model is learned, users and events will be represented by latent vectors that encode the influence of the two relations.

To help users simultaneously find their interested events and suitable partners, Yin et al. [22] utilized the potential friendship. That is, if two users have attended the same event, they may be friends. Additionally, the weight on the edge linking the two user nodes in the user-user graph is proportional to the number of their commonly attended events. To make the learned users' vectors capture the social dimension, the social user-user graph is embedded into the same latent space with user-event, event-word, event-location, and event-time graphs.

Liao et al. [9] utilized the potential trust relationship of a user to obtain the user's preference for the unexperienced events. The trust value of a user holds for another user depends on the similarity between them and the social status of the trusted user. Based on the trust relationships among users, random walks are performed in order to simulate the process of a user consulting with his/her friends to obtain the opinions of their friends for the unexperienced events.

### 3.1.5. Summary of Contextual Factors

We summarize the main contextual factors and extraction or representation approaches, as shown in Table 1. There are other contextual factors that are considered in the literature, such as the number of events attended by the user in the past; the number of similar events the user has attended with the candidate event; the number of RSVPs for the candidate event [14]; the social impacts of events [29]; social network features, such as the link information and degree distributions [20,30]; participant influence [31]; relations of different types of entities [32,33]; image content [34]; surrounding environment (such as brightness, noise, and obstacles) [35,36]; event capacities, spatio-temporal conflict, and travel expenditure [37]; and, the participation lower bound [38].

In a summary, there are many contextual factors that are considered in the literature. However, most of these contexts are predefined and static ones, i.e., representational contexts, few works investigate interactional contexts that are complex, partially observable, and dynamic. Recent research has shown that it is possible to model implicit contexts, such as the user's intention. In [39], an attentional intention-aware recommender system is proposed in order to predict category-wise future user intention based on the recurrent neural network. Other recent works [40–42] have shown that considering sequential contextual information can improve recommender performance, because sequences encode the long and short-term preferences of the user. Besides, the problems, such as evaluating the relevance of contextual information to the user preferences, modeling the correlation, and mutual influence among various contextual factors, needs further investigation.

**Table 1.** Summary of contextual factors used in Context-Aware Recommender Systems (CARS).

| Contextual Factors | Extraction or Representation Approaches | Representative References |
|---|---|---|
| Text content factor | LDA | [17,19,20,29,43,44] |
| | TF-IDF | [14,23] |
| | event-word bipartite graph, graph-based embedding algorithm | [22] |
| | Glove | [16] |
| | CNN | [21] |
| Temporal factor | the day of the week; the hour of day | [17] |
| | 7 × 24 dimensional vector in the space of all possible days of the week and hours of the day | [23] |
| | 7 × 4 dimensional vector in the space of all possible days of the week and time regions of the day | [14] |
| | 33 time slots include 24 hour slots, 7 day slots, and 2 time slots which indicate weekday and weekend. | [22] |
| | temporal patterns such as weekday-hour, month-weekday-hour, day-hour, and month-day-hour patterns | [24] |
| | • a session node $s(u,d)$denotes that a user $u$ has joined an event in day $d$ of a week (e.g., Saturday);<br>• time duration (in days) between two successive events of each user | [25] |
| Spatial factor | the distance between the user's location and the event's location | [17] |
| | $k$-means algorithm to obtain $k$ regions | [28] |
| | obtain a set of discrete regions by using DBSCAN | [22] |
| | split the city into even grid cells according to coordinates | [24] |
| | a kernel-based density estimation approach is used to model the mobility patterns of users as distributions of geographic distances between the attended events | [23] |
| | each venue is associated with venue topics, which are represented by multinomial distributions over venues | [20] |
| | • textual reviews and tips of a venue are modeled as a vector by using Glove tool;<br>• venue category and services of a venue are represented by sparse one-hot vectors;<br>• multilayer perceptrons are used to obtain the venue representations | [16] |
| Social factor | two types of social relationships between the user and the host are extracted to compute the social similarity between two events: following relationship; preferring relationship | [17] |
| | • the offline relationship between users is modeled based on the co-participation of events;<br>• the online relationship between users based on the co-participation of online social groups | [14,28] |
| | • use-group relations denote the interactions between users and the groups they are affiliated to;<br>• group-event relations denote the interactions between groups and the events created by them | [23] |
| | potential friendship: the edge linking the two user nodes in the user-user graph, the weight on the edge is proportional to the number of their commonly attended events. | [22] |
| | potential trust relationship: computed by combining the similarity between two users with the social status value of the trusted user | [9] |

### *3.2. Computing Techniques about CARS in EBSNs*

In this section, we take a brief review of computing techniques about CARS in EBSNs. Figure 3 gives the list of the techniques used in the literature. They are broadly classified into six mutually disjunctive categories: Matrix Factorization (MF), Learning to Rank (LTR), Probabilistic Model (PM), Graph-based Model (GBM), Deep Learning (DL), and Heuristic-based Algorithm (HBA).

Among them, MF is a fundamental technique in linear algebra that factorizes a large matrix into a product of smaller matrices with specific properties. LTR takes the recommendation as a ranking problem and employ learning algorithms to learn a ranking model, which will be finally applied in order to sort the candidate items according to their relevance to users. PM addresses the recommendation task based on probability and statistics theory. It uncovers the hidden patterns in the data and uses them to summarize data and form predictions. GBM represents social networks as graphs, and it employs graph learning methods to convert the recommendation task into a problem of computing the convergency probabilities of nodes. DL is a sub field of machine learning that is based on learning multiple layers of representations, typically by using artificial neural networks. Through the layer hierarchy of a DL model, the higher-level concepts are defined from the lower-level concepts. Finally, HBA represents the algorithm that is used to address NP-hard problem in the recommendation task.

In the following Sections 3.2.1–3.2.6, we will describe these techniques in more detail. Additionally, in Section 3.2.7, we summarize the principal characteristics of these techniques.

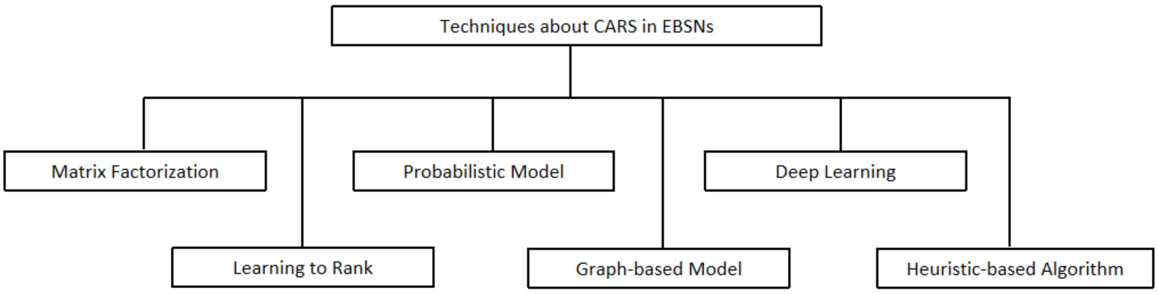

**Figure 3.** Layout of the techniques used in the literature.

#### 3.2.1. Matrix Factorization

Matrix Factorization (MF) is a Latent Factor Model (LFM), which is generally effective at estimating an overall structure hidden in the observations. The basic idea of matrix factorization is decomposing a large matrix into smaller matrices with specific properties, with the goal of the original matrix being retained when multiplying the smaller matrices. Representative MF methods include Singular Value Decomposition (SVD) [45], Non-negative Matrix Factorization (NMF) [46], Probabilistic Matrix Factorization (PMF) [47], Collective Matrix Factorization (CMF) [48], etc.

MF techniques are widely used in the CARS of movie, music, tourism, and restaurant, etc. However, because recommendation in EBSNs is faced with serious cold-start problem, the classic MF approaches, whose recommendation performance depends heavily on the sufficient historical interactions between users and items, does not work well. To tackle this problem, approaches that extend the classic MF are proposed. This section takes an overview of the usage of MF techniques about CARS in EBSNs.

Qiao et al. [28] proposed an extended matrix factorization model to combine three types of information, i.e., heterogeneous social relationships, geographical features of events, and implicit rating data. Specifically, given a user $u_i$ and a candidate event $e_j$, then the predicted ratings of $u_i$ on $e_j$ is computed, as follows:

$$\hat{r}(u_{i,}e_j) = \lambda\hat{r}_1(u_{i,}e_j) + (1-\lambda)\hat{r}_2(u_{i,}e_j), \tag{1}$$

where $\lambda$ is a hyperparameter to control the contribution of the two parts. Additionally, $\hat{r}_1(u_i,e_j)$ is obtained by using the classic matrix factorization model, as follows:

$$\hat{r}_1(u_i,e_j) = \boldsymbol{u}_i^T \boldsymbol{e}_j, \tag{2}$$

where $\boldsymbol{u}_i$, $\boldsymbol{e}_j$ are the low-dimensional latent factor vectors of $u_i$ and $e_j$, respectively. And $\hat{r}_2(u_i,e_j)$ is the geographical preference rating of $u_i$ on $e_j$, which is approximated as follows:

$$\hat{r}_2(u_i,e_j) = \sum_{t=1}^{k} \tilde{\boldsymbol{u}}_i^T \boldsymbol{g}_t p_{jt}, \tag{3}$$

where $k$ is the number of regions where the events were held, $\tilde{\boldsymbol{u}}$ is a geography related low-dimensional latent factor vector of $u_i$, $\boldsymbol{g}_t$ is a low-dimensional latent column vector of region $d_t$, and $p_{jt}$ is the probability of $e_j$ that belongs to $d_t$.

Additionally, the heterogeneous social relationship are modeled as a social regularization term of the objective function, which is based on the Bayesian Personalized Ranking (BPR) [49] optimization criterion.

Du et al. [17] integrated the SVD model with a multi-factor neighborhood method to predict a user's rating for a candidate event. Figure 4 shows the framework of the model, which consists of three components: the attendance matrix construction, the neighborhood discovering, and the event attendance prediction. The attendance matrix construction is based on the current attendance of the candidate event and the historical attendance of all the users. Neighborhood discovering considers multiple contextual factors, including content, temporal, spatial, and social factors. Additionally, the user's attendance to the candidate event is predicted by combing the MF model with the neighborhood based prediction model. The predicted rating of $u_i$ on $e_j$ is computed, as follows:

$$\hat{r}(u_i,e_j) = \boldsymbol{u}_i^T \boldsymbol{e}_j + |\mathcal{N}(u_i,e_j;k)|^{-\frac{1}{2}} \sum_{t \in \mathcal{N}(u_i,e_j;k)} \alpha_{ik}(r(u_i,e_t) - \bar{r}_{u_i}), \tag{4}$$

where $\bar{r}_{u_i}$ is the mean of ratings of $u_i$, $\alpha_{ik}$ is a parameter to be learned together with the MF model parameters, $\mathcal{N}(u_i,e_j;k)$ is the set of $k$-nearest neighbors of $e_j$. The selection of neighbors are determined by the similarity between $e_j$ and each past event attended by $u_i$. The computation of the similarity considers the impact of five contextual features. Given a past event $e_t$ that was attended by $u_i$, the similarity of $e_t$ with the candidate event $e_j$ is computed, as follows:

$$Similary_{u_i}(e_j,e_t) = \lambda_1 C_{u_i}(e_j,e_t) + \lambda_2 W_{u_i}(e_j,e_t) + \lambda_3 H_{u_i}(e_j,e_t) + \lambda_4 L_{u_i}(e_j,e_t) + \lambda_5 S_{u_i}(e_j,e_t), \tag{5}$$

where $C_{u_i}(e_j,e_t)$, $W_{u_i}(e_j,e_t)$, $H_{u_i}(e_j,e_t)$, $L_{u_i}(e_j,e_t)$, and $S_{u_i}(e_j,e_t)$ denote the similarity of between $e_t$ and $e_j$ with regard to event content, day of the week, hour of the day, event location, and the relationship between the user and event host. $\lambda_1, \ldots, \lambda_5$ represent the weights of above features, which are learned in advance by using decision tree on the event attendance prediction issue.

The parameters of the proposed model are learned by minimizing the objective function, as follows:

$$\sum_{(i,j,r(u_i,e_j)) \in \mathcal{D}} (r(u_i,e_j)\text{-}\hat{r}(u_i,e_j))^2 + \lambda \parallel \boldsymbol{u}_i \parallel + \parallel \boldsymbol{e}_j \parallel, \tag{6}$$

where $\mathcal{D}$ is the observation dataset and $\lambda$ is the regularization parameter.

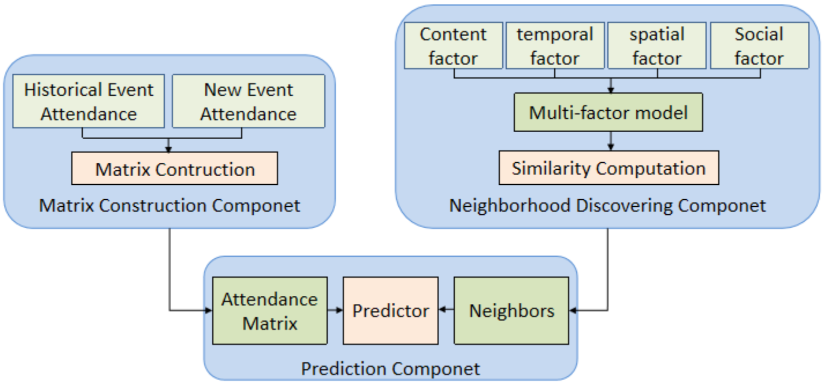

**Figure 4.** Framework of SVD-MFN model.

Gu et al. [14] proposed a unified model that linearly combines the classic MF model with the linear contextual features model. Specifically, the MF model is used to model implicit feedback of users on events. Meanwhile, the linear contextual features model is used to model explicit contextual features, like semantic, spatial, temporal, group, and social features. The final contextual feature between a user and an event is represented as a feature vector by concatenating above contextual features together. To recommend events for a given user, the rating of the user to each candidate event is computed, and a sorted list of recommended events is generated by ranking the ratings.

All the approaches mentioned above obtained an improvement in the commendation performance. However, because they are based on the classic MF model that performs poor when the interaction data is sparse, the improvement on the performance is limited.

### 3.2.2. Learning to Rank

Instead of focusing on the prediction as a stepping stone to make recommendations, Learning to Rank (LTR) takes the recommendation as a ranking problem and optimizes a model using a ranking function. LTR is a kind of supervised machine learning technique that trains model in the ranking tasks [50]. LTR techniques are categorized into three types [51]:

- Pointwise approach: it takes the positive and negative examples as the input and regards ranking as a binary classification or regression problem;
- Pairwise approach: it cares about the relative order between users' preferences on two items. A loss function is defined on pairwise items, with the goal of minimizing the number of miss-classified pairs; and,
- Listwise approach: it optimizes the ranking of the whole list to generate the optimal ordering.

In recommender systems, LTR is usually used as an optimization framework for other models, such as MF, Bayesian latent factor model, neural networks, and graph-based model, etc.

Figure 5 shows the process of LTR-based recommendation. The training set includes the user-event interaction records. Subsequently, a specific learning algorithm is utilized on the training set to learn a ranking function $f(u, e)$, where $u$, $e$ denote a user and an event, respectively. In the test phase, given a target user $u_s$, the ranking system employs the learned rank function to sort the candidate events $e_1, e_2, \ldots, e_s$ and generates the recommendation list to $u_s$. From above illustration, we can see that LTR is a kind of supervised learning model.

In above three types of LTR techniques, a pairwise approach, such as BPR, is the most often used one. Just as we have mentioned in Section 3.2.1, Qiao et al. [28] combined the MF model with BPR for event recommendation. Next, we illustrate how BPR is combined with MF in [28].

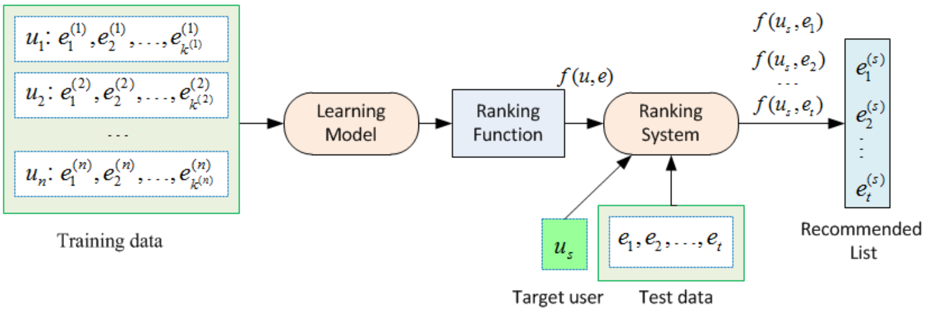

**Figure 5.** Process of Learning to Rank (LTR)-based recommendation.

According to the BPR optimization criterion, the generic optimization criterion for personalized ranking is as follows:

$$\max_{\Theta} \prod_{(u_i,e_j,e_k)\in(\mathcal{U},\mathcal{E}^+_{u_i},\mathcal{E}^-_{u_i})} P(\hat{r}(u_i,e_j) > \hat{r}(u_i,e_k)|\Theta), \tag{7}$$

where $\Theta$ is the parameter set of MF model, $\mathcal{E}^+_{u_i}$ is the positive event set, including the events attended by $u_{i,}$, and the negative event set of the remaining events is denoted as $\mathcal{E}^-_{u_i}$. In order to incorporate the online and offline social relationships into the model, a social regularization term is used, as follows:

$$s(\boldsymbol{u}_i|C^{on}, C^{off}, W, \sigma) \propto N(\boldsymbol{u}_i - \sum_{j=1}^{n} \beta_{ij}\boldsymbol{u}_i|\mathbf{0}, \sigma^2 I), \tag{8}$$

where $C^{on}, C^{off}$ are the predefined online social relationship matrix and offline social relationship matrix, respectively. $W$ is a predefined weight matrix that represents the confident weights in heterogenous social relationships, in which the element $\beta_{ij}$ indicates the strenghth of heterogenous relationship between $u_i$ and $u_j$. $N$ denotes a Gaussian distribution with zero mean and variance-covariance matrix $\sigma^2 I$, where $\sigma$ is the prior parameter of the distribution and $I$ is the identity matrix.

Finally, when considering the influence of the heterogeneous social relationship, the above generic optimization criterion could be extended, as follows:

$$\begin{aligned} &\max_{\Theta} \prod_{(u_i,e_j,e_k)\in(U,\mathcal{E}^+_{u_i},\mathcal{E}^-_{u_i})} P(\hat{r}(u_i,e_j) > \hat{r}(u_i,e_k)|\Theta) \cdot \prod_i s(\boldsymbol{u}_i|C^{on}, C^{off}, W, \sigma) \\ &\cdot \prod_i P(\boldsymbol{u}_i|\sigma_{u_i}) P(\tilde{\boldsymbol{u}}_i|\sigma_{\tilde{u}_i}) \cdot \prod_j P(\boldsymbol{e}_j|\sigma_{e_j}) \cdot \prod_t P(\boldsymbol{g}_t|\sigma_{g_t}), \end{aligned} \tag{9}$$

where $\sigma_{u_i}$, $\sigma_{\tilde{u}_i}$, $\sigma_{e_j}$, and $\sigma_{g_t}$ are the prior parameters of Gaussian distributions which are enforced on the latent factor vectors $\boldsymbol{u}_i$, $\tilde{\boldsymbol{u}}_i$, $\boldsymbol{e}_j$, and $\boldsymbol{g}_t$, respectively. The parameters are learned by using the stochastic gradient descent (SGD) algorithm [52]. Once the parameters are learned, Equation (1) is used to predict the score of $u_i$ for the new events. Additionally, the recommendation list can be generated according to the scores.

Lu et al. [53] also combined a MF model with BPR with the aim of recommending groups to users to join. The proposed model can jointly formulate three types of data: geographical information, implicate user rating, and user behavior. Additionally, the BPR framework is used to define the likelihood function. Li et al. [54] combined the Bayesian latent factor model with BPR to make friend recommendation in EBSNs.

Tran et al. [33] proposed Medley of Sub-Attention Networks (MoSAN), a neural architecture for the group recommendation task. MoSAN leverages BPR in order to optimize the pair-wise ranking between the positive and negative events.

Liao et al. [31] proposed an event recommendation model that was based on Poisson factorization and BPR optimization criterion when considering the participant influence.

Du et al. [55] proposed a group event recommendation framework GERF based on learning-to-rank technique. They proposed a novel learning-to-rank algorithm, called Bayesian Group Ranking (BGR), for the group event recommendation task. Particularly, they defined the ranking model of each group as a linear function of the feature vector of each group-event pair. The parameters of ranking model are estimated based on the optimization criterion of Bayesian personalized ranking.

Instead of using the pairwise ranking approach, Macedo et al. [23] used a listwise learning algorithm, which directly optimizes the Normalized Discounted Cumulative Gain (NDCG), which is a ranking evaluation metric to rank candidate events. Besides content factor, factors like social, geographic, and temporal ones are taken as the input features of the algorithm.

As we can see from above works, BPR is more often used in CARS, because it considers the relative order between events in the learning processes. However, pairwise approaches have the limitation of ignoring the global structure of ranking. The listwise approach takes ranking lists as instances in both learning and prediction, so the global structure of ranking is maintained and the ranking evaluation measures can be more directly incorporated into the loss functions in learning. However, the training complexities of some listwise ranking algorithms are high, because the evaluation of their loss functions is permutation based. Therefore, more efficient learning algorithms are needed in order to make the listwise approach more practical.

### 3.2.3. Probabilistic Model

The probabilistic model is a kind of mathematical model, which is used to describe the probability relationship among multiple random variables and it is usually expressed as a set of probability distribution functions. The recommendation methods that are based on probabilistic model employ statistical learning methods to learn probability distributions from sample data, and then these probability distributions are used in order to generate the recommended items. The core of probability generation algorithm is how to learn and obtain the parameters of probability distribution function. Representative probabilistic models include Naive Bayes (NB), Gaussian Mixture Model (GMM) [56], Hidden Markov model (HMM), LDA, Poisson Factorization [57], etc.

A simple probabilistic generative model for collaborative filtering is proposed in [58]. Let $\mathcal{U}$ and $\mathcal{I}$ be the user set and item set, respectively. A latent topic set $\mathcal{T}$ is assumed to capture the users' latent interests and the item profiles. The process of a user $u \in \mathcal{U}$ accessing an item $i \in \mathcal{I}$ is assumed, as follows: user $u$ selects a topic $t \in \mathcal{T}$ according to his/her probability distribution over topics, and the topic $t$ generates an item $i$ according to its probability distribution over items. Assuming that users are independent of items given the chosen topic, the joint probability distribution over $u$, $t$, and $i$ can be computed, as follows:

$$P(u,t,i) = P(u)P(t|u)P(i|t) = P(t)P(u|t)P(i|t). \tag{10}$$

The joint distribution over $u$ and $i$ is:

$$P(u,i) = \sum_{t\in\mathcal{T}} P(u,t,i) = \sum_{t\in\mathcal{T}} P(t)P(u|t)P(i|t), \tag{11}$$

where $P(t)$, $P(u|t)$m and $P(i|t)$ are parameters to be learned, which are denoted as $\Theta$. Additionally, the objective function is as follows:

$$\max_{\Theta} \sum_{(u,i)\in\mathcal{D}} \log(P(u,i|\Theta)), \tag{12}$$

where $\mathcal{D}$ is the set of observed user-item pairs.

After the parameters are learned, items can be ranked according to $P(i|u)$, which is computed as follows:

$$P(i|u) = P(u,i)/P(u) \propto P(u,i). \tag{13}$$

For CARS in EBSNs, Yuan et al. [19] proposed a probabilistic generative model, called COM, based on LDA to model the generative process of group activities and make recommendations for a group of users. The authors assumed that users join a group, because of different group topics and they select events based on the group topic or personal consideration of contextual factors, such as content of events. When making recommendations, COM estimates the preferences of a group to an event by weighted sum of the preferences of the group members. Figure 6 shows the graphical model, where the observed variables are shown as shaded circles, and latent variables are shown as unshaded circles.

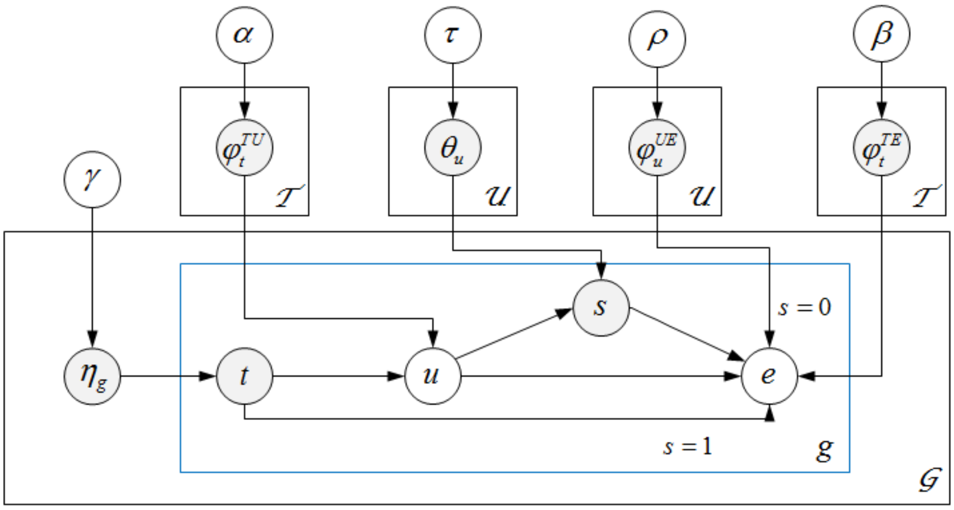

**Figure 6.** Graphical Model of COM.

Specifically, a multinomial distribution $\eta_g$ over latent topics is used to model the topic preferences of group $g$. Additionally, each topic $t$ has a multinomial distribution $\varphi_t^{TU}$ over users, which represents the relevance of users to topic $t$. Additionally, each topic $t$ also has a multinomial distribution $\varphi_t^{TE}$ over events, which represents the relevance of events to the topic $t$. For each member in group $g$, a topic $t$ is drawn from the topic distribution $\eta_g$, and then a user $u$ is sampled according to $\varphi_t^{TU}$. To model the intuition that a user in a group may select events either based on the group topics or his/her personal consideration in content factors, such as the geographical distance, a switch $s$ is used to model the event selection process for a user. If $s = 1$, the event is sampled based on the topic-specific multinomial distribution over events $\varphi_t^{TE}$; if $s = 0$, the event is sampled from the user-specific multinomial distribution over events $\varphi_u^{UE}$. Switch $s$ is sampled from a user-specific Bernoulli distribution with parameter $\theta_u$. Algorithm 1 shows the process of generating the group events.

To estimate the model parameters, the goal is to maximize the likelihood of the group participation data, which is, as follows:

$$\begin{aligned} P(u,t,s,e|\gamma,\alpha,\rho,\beta,\tau) = \int P(s|\theta)P(\theta|\tau)d\theta \cdot \int P(t|\eta)P(\eta|\gamma)d\eta \cdot \int P(u|t,\varphi^{TU})P(\varphi^{TU}|\alpha)d\varphi^{TU} \\ \cdot \iint P(e|u,t,s,\varphi^{UE},\varphi^{TE})P(\varphi^{UE}|\rho)P(\varphi^{TE}|\beta)d\varphi^{UE}d\varphi^{TE}. \end{aligned} \tag{14}$$

A two-step Gibbs sampling algorithm is employed in order to maximize the above likelihood and obtain the unknown parameters$\{\theta, \varphi^{TU}, \varphi^{UE}, \varphi^{TE}\}$.

Du et al. [20] presented a Bayesian probability generative model, called Content-Venue-aware Topic Model (CVTM), in order to extract groups' venue preferences and content preferences. They noticed that there exists high correlation between an organizer and the content of an event, i.e., the events sharing same organizer have more similar content than the events that are held by distinct organizers. The correlation between organizer and textual content is modeled in CVTM to alleviate the sparsity of textual content, where some events are described with very few words.

**Algorithm 1:** Probabilistic process of generating group events in COM.

```
for each topic $t$ do
    Draw $\varphi_t^{TU} \sim Dirichlet(\alpha)$;
    Draw $\varphi_t^{TE} \sim Dirichlet(\beta)$;
end
for each user $u$ do
    Draw $\varphi_u^{UE} \sim Dirichlet(\rho)$;
    Draw $\theta_u \sim Beta(\tau)$, where $\tau = \{\tau_1, \tau_2\}$;
end
for each group $g$ do
    Draw $\eta_g \sim Dirichlet(\gamma)$;
    for each group member $u$ do
        Draw a topic $t \sim Multinomial(\eta_g)$;
        Draw a user $u \sim Multinomial(\varphi_t^{TU})$;
        Draw switch $s \sim Bernoulli(\theta_u)$;
        If $s = 0$
            Draw $e \sim Multinomial(\varphi_u^{UE})$;
        If $s = 1$
            Draw $e \sim Multinomial(\varphi_t^{TE})$;
    end
end
```

Ji et al. [59] proposed a topic-based probabilistic model, snamed GIST, jointly considering individual members' choices and subgroups' choices for group recommendations. Purushotham et al. [60] proposed the Collaborative Filtering-based Bayesian model to capture the location or event semantics and group dynamics, such as user interactions, user group membership, or user influence for group recommendations. Yin et al. [61] proposed a location-aware probabilistic generative model that considers user home locations and event locations. Liao et al. [31] proposed a Poisson factorization model that considers the participant influence for event recommendation. Zhang and Wang [62] proposed a Collective Bayesian Poisson Factorization (CBPF) model for handling cold-start problem in EBSNs. In the model, user preferences to events, social relation, and content text are separately modeled by the Bayesian Poisson factorization. Subsequently, these preferences are further jointly connected by the CMF model. Moreover, event textual content, organizer, and location information are utilized in order to learn the representations of cold-start events.

The probability model makes full use of the advantages of the Bayesian theory and knowledge reasoning, which makes the recommendation system have a good theoretical basis and high recommendation accuracy. The disadvantages of probability model are that there are many parameters to be learned, so that the model training is not very efficient and it cannot support real-time recommendation on large datasets.

### 3.2.4. Graph-Based Model

Graph is the most natural and direct representation of EBSNs. The graph-based models construct graphs consisted of various entities (such as users, groups, and events) and relations of entities in EBSNs. In this section, we introduce how a context-aware recommendation can be generated based on graph learning approaches, such as Random Walk (RW), Random Walk with Restart (RWR), and Multivariate Markov Chain (MMC), etc.

Liao et al. [9] proposed a two-phase group event recommendation (2PGER) model to help satisfy the groups' query for attending unexperienced events. Information, such as online social behaviors, users' event participation records, and topological structures of EBSNs, is first utilized to establish a global trust network among users and an egotrust network for each user. Subsequently, random walks are performed on the egotrust network for each user to acquire the user's preferences on unexperienced events. Finally, the RWR method is used to aggregate users' preferences and top-*N* events are generated to be recommended to the target group.

Because RWR is developed for homogeneous graphs, it cannot explicitly model the influences between different entities in a heterogeneous graph. Therefore, MMC is developed in order to address above problem.

Liu et al. [32] proposed an event recommendation model by using the MMC method on a hybrid graph. Figure 7 shows the hybrid graph construction. There are five kinds of nodes in the graph: user node $u$, event node $e$, group node $g$, host node $h$, and subject node $s$. Explicit relations are used to obtain the explicit edges between two nodes. For example, if user $u_1$ joins an online group $g_1$, then an undirected edge is built to link $u_1$ and $g_1$. Implicit relations between events are represented as the directed dashed lines between events in the figure. Additionally, the implicit relations between events are built according to their cosine similarity of their attribute vectors. The event attributes include event time, event location, event cost, and event type.

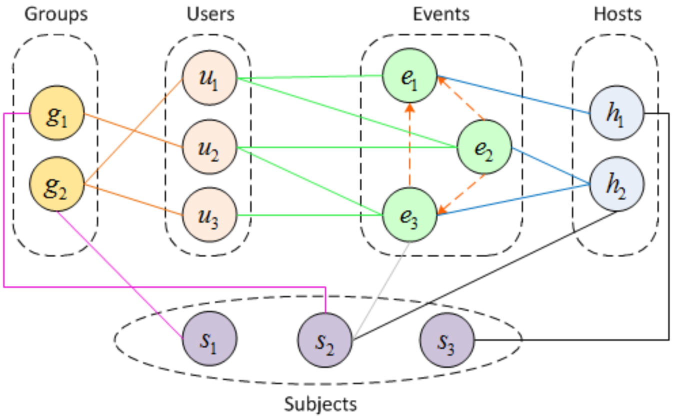

**Figure 7.** Illustration of hybrid graph construction.

A multivariate Markov chain is used in order to transform the event recommendation task into a problem of computing the node convergency probability. In order to obtain the convergency probabilities, the following equations are iteratively computed:

$$\boldsymbol{u}^{(t+1)} = \lambda_{eu}\boldsymbol{e}^{(t)}P_{eu} + \lambda_{gu}\boldsymbol{g}^{(t)}P_{gu} + (1-\lambda_{eu}-\lambda_{gu})\boldsymbol{q}_u, \tag{15}$$

$$\boldsymbol{e}^{(t+1)} = \lambda_{ue}\boldsymbol{u}^{(t)}P_{ue} + \lambda_{he}\boldsymbol{h}^{(t)}P_{he} + \lambda_{se}\boldsymbol{s}^{(t)}P_{se} + (1-\lambda_{ue}-\lambda_{he}-\lambda_{se})\boldsymbol{e}^{(t)}P_{ee}, \tag{16}$$

$$\boldsymbol{h}^{(t+1)} = \lambda_{eh}\boldsymbol{e}^{(t)}P_{eh} + (1-\lambda_{eh})\boldsymbol{s}^{(t)}P_{sh}, \tag{17}$$

$$\boldsymbol{g}^{(t+1)} = \lambda_{ug}\boldsymbol{u}^{(t)}P_{ug} + (1-\lambda_{ug})\boldsymbol{s}^{(t)}P_{sg}, \tag{18}$$

$$\boldsymbol{s}^{(t+1)} = \lambda_{gs}\boldsymbol{g}^{(t)}P_{gs} + \lambda_{hs}\boldsymbol{h}^{(t)}P_{hs} + (1-\lambda_{gs}-\lambda_{hs})\boldsymbol{e}^{(t)}P_{es}, \tag{19}$$

where $\boldsymbol{q}_u$ is the user query vector, $\boldsymbol{u}^{(t)}$, $\boldsymbol{h}^{(t)}$, $\boldsymbol{s}^{(t)}$, $\boldsymbol{e}^{(t)}$, and $\boldsymbol{g}^{(t)}$ are distribution probability vectors that represent the probabilities that users, hosts, subjects, events, and groups are visited at time $t$, respectively. $\lambda_{xy}$ $(x, y \in \{u, e, h, s, g\})$ denotes the transition weight from nodes of type $x$ to nodes of type $y$. $P_{xy}$ is the transition matrix that is obtained by normalizing the matrix $A_{xy}$ by rows, where $A_{xy}$ is the adjacency matrix that represents the relations of nodes of type $x$ and nodes of type $y$.

The iteration terminates until the difference between two iteration probability vectors is smaller than a predefined threshold. When the iteration terminates, a vector of event convergency probabilities is obtained for $u$, which is denoted as $score_{grahp}(u, e)$.

When considering that the graph might ignored, the history preference of individual user to events when the transition weights are not individually set and trained for each user, the history preference of user $u$ on event $e$ is incorporated into the final score computation:

$$score(u, e) = score_{graph}(u, e) \times score_{hist}(u, e), e \in \mathcal{E}_{cand}, \tag{20}$$

where $\mathcal{E}_{cand}$ is the set of candidate events, $score_{hist}(u, e)$ is computed by the cosine similarity between the vector of $e$, and the vector of $u$'s history preference. Subsequently, the recommended event list is generated based on the $score(u, e)$ in descending order.

Pham et al. [25] also employed MMC techniques to construct a model, called HeteRS, which can handle multiple recommendation problems in EBSNs (such as recommending groups to users, tags to groups, and events to users). To incorporate the influence of temporal factors, a new type of nodes, named session node, is introduced to the graph, in which there are already five types of nodes, including users, events, groups, tags, and venues. Besides, a decay function is additionally used to define the importance weight of an event.

The graph embedding technique has been recently proposed in order to solve the problem of high computation and space cost. It converts a graph into a low dimensional space, in which the graph information is preserved. Graph algorithms can be computed efficiently by representing a graph as a set of low dimensional vectors. As one kind of classical social networks, EBSNs are usually denoted as social graphs, which enable them to be a perfect scenario to be applied with graph embedding technique.

Yin et al. [22] proposed a graph-based embedding model to recommend an event-partner pair for a given user. The model collectively embeds all of the observed relations among users, events, locations, time, and text content in a shared low-dimension space. Subsequentlys, the cold-start events are represented as vectors that are learned from their associated contextual information captured by event-word, event-location, and event-time bipartite graphs. Additionally, the users' vectors that capture users' preferences are learned from the user-event relation graph. The success probability of a recommended event-partner pair for a given user considers not only the target user's preference, but also the partner's preference and social proximity between the target user and recommended partner.

The recommendation that is based on graph model has high flexibility and scalability, and it can make relatively high-quality recommendations when the amount of data is insufficient. However, random walk based approaches face a common problem that, when the graph becomes large, the computation is expensive. Although approximation algorithms are proposed, they sacrifice the accuracy. Utilizing graph embedding to study CAR is a new research area. Additionally, problemsm such as how to interpret the recommendation results, how to characterize the evolution of networks, and how to preserve the graph structure, need further exploration.

### 3.2.5. Deep Learning

In recent years, deep learning and representation learning techniques have attracted considerable attention in the recommendation research community, and they are also widely applied in CARS. By now, the related CARS models in EBSNs cover a wide range of deep learning techniques, including the convolutional neural network (CNN), Recurrent Neural Networks (RNN), Autoencoders, attention mechanism, Restricted Boltzmann Machines (RBM), etc. In this section, we will review related works and analyze their advantages and short comings.

Wang et al. [21] proposed an event recommendation model, named DUMER, by combining the CNN technique with PMF model. The architecture of the model is shown in Figure 8. DUMER first employs CNN with word embedding to capture the textual features of events attended by users.

Subsequently, the convoluted features are used to form the user latent factors for PMF. The PMF approximate the attendance matrix . Additionally, the recommendation list is finally generated according to the attendance matrix. The detailed process is as follows.

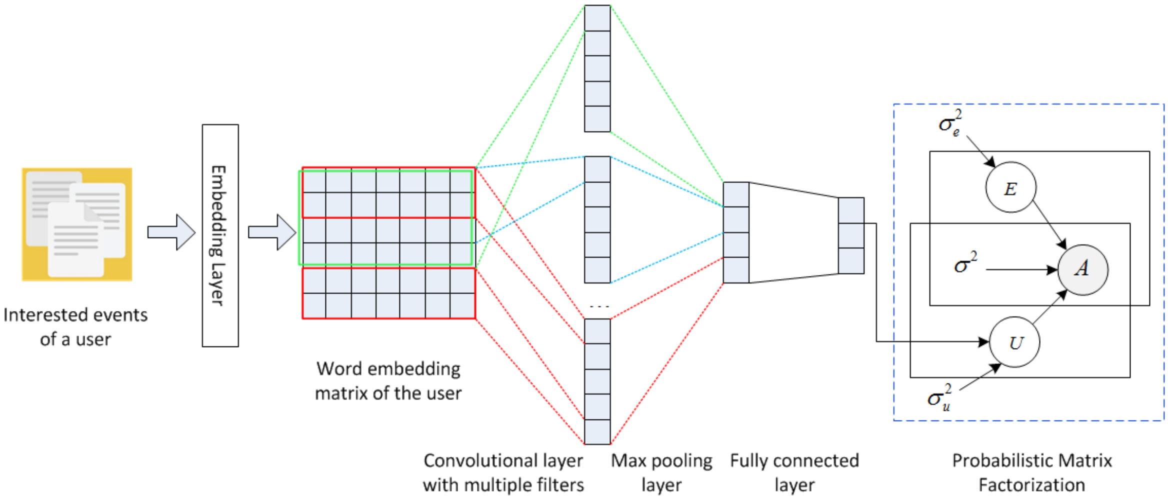
Interested events of a user
Embedding Layer
Word embedding matrix of the user
Convolutional layer with multiple filters
Max pooling layer
Fully connected layer
Probabilistic Matrix Factorization
$\sigma_e^2$
$E$
$\sigma^2$
$A$
$U$
$\sigma_u^2$

**Figure 8.** Architecture of DUMER.

The event recommendation problem is cast as an attendance matrix approximation problem. Let $A \in \mathbb{R}^{n \times m}$ stand for the attendance matrix, where $n$ and $m$ are the numbers of users and that of events, respectively. The attendance matrix approximation problem can be formulated as maximizing the posterior probability, as follows:

$$P(A|U, E, \sigma^2) = \prod_i^n \prod_j^m P(A_{ij}|U_i E_j, \sigma^2)^{I_{ij}}, \tag{21}$$

where $U_i$, $E_j$ are the latent factors of user $u_i$ and event $e_j$, respectively. $P(A_{ij}|U_i E_j, \sigma^2)$ is the probability density function of Gaussian distribution, whose mean is $U_i E_j$ and variance is $\sigma^2$. $I$ is the indicator matrix, the element of which, $I_{ij}$ is 1 when user $u_i$ attended event $e_j$, otherwise 0.

A word embedding layer is firstly used to transform each event description of the user's past events into a word embedding matrix in order to capture the contextual information of a user's interested events. Additionally, all of the embedding matrices are concatenated together to form the user's word embedding matrix. Subsequently, CNN is utilized to characterize the preference of the user on events. There are three layers in DUMER: the convolutional layer, the pooling layer, and the fully-connected layer.

In the convolutional layer, multiple convolution filters are used to transform the input matrix into multiple feature maps. Let $B_i$ be the word embedding matrix of user $u_i$ and the convoluted feature vector of $B_i$ with the $j$th convolutional filter is as follows:

$$\boldsymbol{l}_i^j = \mathrm{ReLU}(W_1^j \odot B_i), \tag{22}$$

where $W_1^j \in \mathbb{R}^{q \times s}$ is the $j$th convolutional filter ($q$ is the window size and $s$ is the dimension of word embedding) and $\odot$ stands for the convolution operation.

In the pooling layer, one-max-pooling operation [63] is utilized to select the largest number from each feature map. The feature vector at the pooling layer for user $u_i$ is represented, as follows:

$$\boldsymbol{r}_i = [\max(\boldsymbol{l}_i^1), \max(\boldsymbol{l}_i^2), \ldots, \max(\boldsymbol{l}_i^k)], \tag{23}$$

where $k$ is the number of filters.

In the fully connected layer, the output of the pooling layers is transformed to the user latent factor for $u_i$, which can be formulated, as follows:

$$\boldsymbol{x}_i = softmax(W_2 \otimes \boldsymbol{r}_i), \tag{24}$$

where $W_2$ is the weight matrix of the multi-layer perceptron. $\boldsymbol{x}_i$ is the final feature vector of user $u_i$.

All of the vectors of users are concatenated to form the feature matrix $X$. Additionally, then it is taken as the user latent factor for PMF.

To better fit the attendance matrix, a Gaussian noise variable is added to the user feature matrix $X$, and then the user latent matrix is:

$$U = X + Y, \tag{25}$$

where $Y$ is a Gaussian noise matrix, whose entries follow Gaussian distribution (whose mean is 0 and variance is $\sigma_u^2$). Additionally, the event latent factor is initialized as following Gaussian distribution whose variance is $\sigma_e^2$. To conduct PMF, the goal is to maximize the posterior probability, as follows:

$$\max_{U,E,W_1,W_2} P(U, E, W_1, W_2 | A, \mathcal{Z}, \sigma^2, \sigma_u^2, \sigma_e^2) = \max_{U,E,W_1,W_2} [P(A|U, E, \sigma^2) P(E|\sigma_e^2) P(U|W_1, W_2, \mathcal{Z}, \sigma_u^2)], \tag{26}$$

where $\mathcal{Z}$ is the set of all event descriptions.

Wu et al. [64] proposed a three-level hierarchical Long Short-Term Memory (LSTM) architecture in order to predict event attendance of each user, in which users' evolving preferences are explicitly modeled. Specifically, the users preferences are modeled from three dimensions: sequential preferences, which are denoted as a sequence of events attended by the user; contextual preferences, which are the spatial and temporal features of the events in above sequences; exclusive preferences, which represent the implicit influences between events. THe above preferences are encoded by three LSTM encoder respectively, and the Multilayer Perceptron (MLP) component is utilized in order to derive the attendance probability.

Tran et al. [33] proposed an attentive neural model for the group recommendation task. The model first captures a user's preference with respect to all other group members by a sub-attention module. Subsequently, a medley of sub-attention modules is used to obtain the group's final preferences. Because a user's preference may be highly influenced by the other members in the group, modeling the interactions among group members is crucial. In this paper, the user-user interactions are modeled based on the sub-attention networks.

Jhamb et al. [65] proposed a contextual recommendation model that is based on a denoising autoencoder neural network. The contextual attributes such as online groups and event venues are encoded by an attention mechanism to obtain the hidden representation of the users' preferences.

Wang and Tang [24] proposed an embedding method, named Event2Vec, in order to encode events in a low-dimension latent space that integrates the spatial and temporal influence. Specifically, embedding layers and fully connected layers are utilized to learn representations for three factors: the event, the location, and the time based on the event sequential data attended by users. Multitask learning settings are proposed to model and predict user's preference on three factors.

Pramanik et al. [16] presented venue recommendation system DeepVenue, which provides venue recommendations for the Meetup event-hosts to host their events. The authors argue that an event $e$ can be hosted by a group $g$ at a particular venue $v$ successfully only if one of the following situations is true: (a) $v$ is similar to the venues where events similar to $e$ have been recently hosted successfully, (b) $e$ is similar to the events that have been hosted at venue $v$ successfully, and (c) $g$ is similar to the groups recently hosted events at $v$ successfully. Three modules are constructed based on embedding and LSTM techniques to compute the suitability of the candidate venue from above three perspectives. Additionally, all of the suitability values are concatenated to pass through a dense layer to obtain the final prediction score that indicates the possibility of venue to host event organized by the group.

Li et al. [66] incorporated multiple RBMs and a conditional RBM to solve the cold-start problem of group event recommendation. RBMs are used to extracts high latent group preference from user feedback and group feedback, and the conditional RBM obtains latent event features that are based on contextual information, such as location and organizer of events.

Deep learning techniques automatically extract features, effectively capture latent relationships, and represent complex abstractions in higher layers. However, the hidden layers in most deep neural networks do not possess understandable meanings. Therefore, the explainability of the deep models needs further exploration.

### 3.2.6. Heuristic-Based Algorithms

One important task of EBSN recommender systems is recommending personalized event arrangement (or planning) for each user. When recommending event arrangement, users need to consider many factors like spatio-temporal conflicts of different events, travel expenditure, capacities of events, etc. The event arrangement problem is usually proved to be NP-hard, and heuristic algorithms are the widely used solutions [67]. We overview the related work, as follows.

Li et al. [68] proposed the social event organization (SEO) problem, which is to assign a set of events for a group of users by maximizing the overall innate and social affinities. However, this work only considers the similarity of attributes and social friendship among users, without explicitly modeling the spatial influence between events and users. Tong et al. [69] defined a new problem of Bottleneck-aware Social Event Arrangement (BSEA). Three factors are considered in the problem: distances between events and users, attribute similarities between events and users, and friend relationships among users. Given a set of events $\mathcal{E}$ and a set of users $\mathcal{U}$, the utility that the user $u \in \mathcal{U}$ is assigned to an event $e \in \mathcal{E}$ is measured, as follows:

$$f(u,e) = \lambda \cdot (1 - \frac{d(l_u, l_e)}{\max\limits_{u,e} d(l_u, l_e)}) + (1-\lambda) \cdot g(u,e), \tag{27}$$

where $\lambda \in [0,1]$ is a parameter that is used to balance the two terms in the equation, $l_u$, $l_e$ represent the geographic coordinates of user $u$ and event $e$, respectively. $g(u,e)$ represents the attribute similarity between $u$ and $e$.

Additionally, given an event $e$, a set of users $\mathcal{U}$, and an arrangement $\mathcal{R}$, the normalized utility of event $e$ is represented by the following equation:

$$f_{\mathcal{R}}(e) = \frac{\sum\limits_{u \in \mathcal{U}, (u,a) \in \mathcal{R}} f(u,e)}{\gamma_e}, \tag{28}$$

where $\gamma_e$ is the capacity of the event $e$.

The goal of BSEA problem is to find an arrangement $\mathcal{R}$ to maximize $\min_{e \in \mathcal{R}}\{f_{\mathcal{R}}(e)\}$, such that the capacity and social friendship constraints are satisfied.

Because BSEA is proven to be NP-hard, the authors devised two greedy-based heuristic algorithms to approximately solve the BSEA problem, i.e., Greedy and Random+Greedy. Additionally, the Random+Greedy algorithm is verified to be faster and more effective than the Greedy algorithm in most cases in the experiments. Algorithm 2 illustrates the procedure of the Random+Greedy algorithm. In the algorithm, $F$ represents a social network graph, where each vertex is a user, and any two users (vertices) are connected by an edge if and only if they are mutual friends. $\theta$ is a predefined threshold to indicate when the procedure should be stopped. $C(e_i)$ is an ordered list, which stores a list of users that have been visited before, and users are stored in the order that they are visited. Any two users in $C(e_i)$ are not friends mutually. $\mathcal{R}(e_i)$ is the arrangement of users for event $e_i$.

**Algorithm 2:** Random+Greedy.

```
Input: $\mathcal{E}, \mathcal{U}, F$
Output: $\mathcal{R}$
initialize $\mathcal{R}$;
$\mathcal{E}' \leftarrow$ a shuffle-ordered list of $\mathcal{E}$;
for each $e_i \in \mathcal{E}'$ do
    calculate $\theta$;
    $\mathcal{L} \leftarrow$ sorted list of unassigned $u \in \mathcal{U}$ in non-ascending order of utility $f(u, e_i)$;
    for $u_j \in \mathcal{L}$ do
        if $u_j$ has friends in $\mathcal{R}(e_i)$ then
            add $u_j$ to $\mathcal{R}(e_i)$;
            add friends of $u_j$ from $C(e_j)$to $\mathcal{R}(e_i)$ if not exceeding $\theta$ and $\gamma_{e_i}$;
        else if $u_j$ has friends in $C(e_j)$ then
            add $u_j$ and its friends from $C(e_j)$ to $\mathcal{R}(e_i)$ if not exceeding $\theta$ and $\gamma_{e_i}$;
        else
            append $u_j$ to the tail of $C(e_j)$;
        if $|\mathcal{R}(e_i)| \geq \min\{\theta, \gamma_{e_i}\}$ then
            Break;
    end
end
return $\mathcal{R}$.
```

She et al. [37] defined a problem of Utility-aware Social Event-participant Planning (USEP) considering event capacities, spatio-temporal conflict, and travel expenditure constraints in order to maximize the overall satisfaction of event participants. The total utility score of the planning $\mathcal{R}$ towards both event organizers and users is defined, as follows:

$$\Omega(\mathcal{R}) = \sum_{u} \sum_{e \in \mathcal{E}_u} \mu(e, u), \tag{29}$$

where $\mu(e, u)$ represents the preference of $u$ towards event $e$, $\mathcal{E}_u$ is the schedule of arranged events in increasing time order for user $u$.

The authors proved that this problem is NP-hard, and devised a greedy-based heuristic algorithm that performs fast but has no approximation guarantee. Subsequently, they presented a two-step approximation framework, which guarantees an approximation ratio and includes optimization techniques to improve its space and time efficiency. In [70], they also studied a new event-participant arrangement strategy for online EBSNs that can learn the satisfaction of users towards the arrangement through their feedback on accepting or rejecting the events.

Cheng et al. [38] defined an extension of the Global Event Planning problem [37], called Global Event Planning with Constraints (GEPC). GEPC not only considers the participation upper bound, but also the participation lower bound for each event. Subsequently, a two-step framework solution with approximate guarantees is proposed. Xin et al. [71] made improvements based on [38] and proposed a heuristic dynamic programming strategy that can consider constraints asynchronously.

Kou et al. [72] proposed an interaction-aware global event-participant arrangement strategy that not only considers the interests of users, but also considers the potential interactions among participants. Li et al. [73] proposed an incremental bilateral preference stable planning problem, while taking user preferences and the needs of event organizer into account. Liang [74] proposed the event scheduling problem in online scenes, while considering the needs of users, events, and organizers.

The heuristic-based approaches obtained suitable recommendation results by considering various conditions. In addition to the above conditions, more conditions, such as the user's free time, the category diversity, and time balance requirements of event arrangement, could be further incorporated into the model. In addition, the dataset crawled from EBSNs platform does not usually contain some necessary constraint information; therefore, synthetic data need to be generated for evaluation.

#### 3.2.7. Summary of Computing Techniques

Each technique described above has its advantages and limitations for context-aware recommendation in EBSNs. We present a brief overview of them, as shown in Table 2.

In recent years, as compared with other techniques, DL has achieved greater success in the area of CARS; however, it has the limitation of poor explainability, i.e., we do not fully understand why a context plays a more important role than others and how an item is recommended out of the other items. Therefore, an important task is to make the deep model itself explainable for the recommendation. Additionally, heuristic based algorithms, e.g., Bio-inspired algorithms, such as artificial Algae algorithm, bat algorithm, need to be further explored in this area.

Next, we discuss the match problem of the context representation with computing techniques from four aspects, as follows: (1) when considering the representation of the text content contextual factor, the natural language processing method (e.g., Glove, CBOW) is the most widely used one. Additionally, most of the computing technologies will get a good performance by adopting this method. However, for the graph-based embedding model, the content document of an event is presented as a set of word nodes in a bipartite graph. Additionally, when the computing resources are available, deep learning techniques such as CNN are a good choice for most computing techniques since they have been proved to be effective in capturing the contextual information from event descriptions. (2) When considering the temporal contextual factor, the weekday-hour pattern has been validated to be able to achieve the best performance compared with other temporal patterns. This pattern is applicable to most of the computing techniques, except GBM, in which the temporal factor is modeled as nodes in a graph. (3) When considering the spatial contextual factor, the continuous spatial information is usually transformed into a set of discrete regions by using clustering algorithms such as k-means, DBSCAN. Subsequently, the distance between the user's location and the event's location is computed based on these regions. Representing the spatial contextual factor as the distance between two locations is applicable to MF, LTR, and PM for obtaining the users' location preferences. However, for DL, representing the locations as feature vectors is more appropriate. Additionally, for GBM, it would be preferred to represent the regions as the region nodes in an event-location bipartite graph. (4) When considering the social contextual factor, for MF, the influence of social factor is modeled as the social regularization term of the objective function. Additionally, for GBM, the influence of social factor is represented by the weight on the edge between two user nodes.

**Table 2.** Overview of techniques used about CARS in EBSNs.

| Techniques | Advantages | Limitations | Applicable Situation | References |
|---|---|---|---|---|
| MF | • low complexity<br>• high scalability | • poor explainability<br>• linear modeling<br>• data sparseness problem | Dense rating matrix or abundant contexts | [14,17,28,53] |
| LTR | • Focus on item ranking | • high computational complexity | top-*N* recommendation | [23,25,28,31,33,53–55] |
| PM | • good theoretical basis | • low efficiency | non real-time recommendation | [19,20,31,59–62,75] |
| GBM | • high scalability<br>• utilizing graph topology | • high complexity | network scale is not large | [9,22,25,32,76] |
| DL | • nonlinear modeling<br>• high scalability | • poor explainability | large amount of data; complex feature engineering | [16,21,24,33,64–66] |
| HBA | • consider various constraints | • no guarantee on the degree of deviation between the feasible solution and the optimal solution | NP-hard problem | [37,38,68–70,77–79] |

## 4. Datasets and Evaluation Metrics

In this paper, we review the datasets and evaluation metrics used to measure the weakness and strength of recommendation methods in related literature.

### *4.1. Datasets*

There are two kinds of datasets used for the evaluation of CARS in EBSNs: real-world datasets; synthetic datasets. The real-world datasets are usually crawled from some well-known websites. Additionally, the synthetic datasets are created by simulating the contextual attributes, which are used to evaluate the recommendation algorithms.

Douban Event and Meetup are two well-known websites that provide APIs for users to crawl datasets. Douban Event is the largest online event-based social network in China, while Meetup found in New York in USA has users and groups worldwide. We take Meetup as an example to introduce the general use of an EBSN platform. Meetup consists of entities, like online groups, users, and events. In Meetup, users can create online groups (e.g., "Team-Art", "intersoccer") to share comments, photos, and event plans. Users can create events in online groups to attract other users to join by specifying when, where, and what the event is. Subsequently, the event creator can distribute the created social event to selected users or make it public. Other users may express their intentions to attend the event by making RSVPs online. As users prefer to participate the offline events nearby, Meetup organizes information by cities and provides users with the events which are located in the same city to attend. That is the reason why most approaches in CARS recommend events in the local city where users live. Additionally, large cities in the USA, like San Jose, Phoenix, and Chicago, are often selected for investigation, since they publish large number of events every day and users are active on the platform. Sometimes, the crawled data from websites do not contain the required contextual information, and researchers resort to create their own synthetic datasets by simulating the contextual attributes that they want to evaluate their algorithms on. For example, in [37], spatio-temporal conflicts, capacities, and travel budgets are all synthetically generated.

Existing literature in EBSNs rarely makes its CARS datasets public. Most of those synthesized datasets are privately owned. Although there are a few works, such as [25], which have released their datasets, the types of contextual factors in the datasets are very limited. Therefore, learning and implementing existing algorithms for CARS has become challenging for research communities.

### *4.2. Evaluation Metrics*

There are different evaluation approaches used to test the effectiveness of a recommendation algorithm. Usually, they can be categorized into two categories: online evaluation and offline evaluation. The online evaluation tests the recommendation performance in real-time with metrics, like Click-Through-Rate (CTR) and Bounce Rate. In off-line evaluation, users are not involved and it will be more efficient and economical to conduct the evaluation experiments on large scale datasets as compared to online evaluation. Our results indicate that the majority of studies on CARS in EBSNs have used off-line evaluation. Metrics in offline evaluation can be categorized into accuracy metrics and usefulness metrics [80,81]. Accuracy metrics measure a system's ability of predicting a user's ratings on items, and usefulness metrics measure the suitability of recommended results to users.

The results of this survey show that the majority of studies have used the accuracy metrics. And the accuracy metrics can be categorized into following three categories [81]:

- Predictive accuracy metrics: they measure how close the predicted ratings are to the true ratings. The Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) are the widely used metrics.
- Classification metrics: they measure how frequent the system can make correct or incorrect decisions regarding whether an item is of interest to the target user. The widely used metrics are Precision, Recall, F1-measure, Macro-F1, HitRate, Mean Average Precision (MAP), and Area Under the ROC Curve (AUC).
- Rank accuracy metrics: they measure the ability of a recommender system to rank the truly interested items higher in the recommendation list. The widely used metrics are Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), and Mean Inverse Rank (MIR).

We have also found that previous studies have used following three usefulness metrics in order to measure the suitability of their recommendations:

- Coverage metric: it measures the proportion of events recommended to users in the test events.
- Utility metric: it defines whether the recommendation results are interested to the users in their contexts.
- S-Pearson metric: it measures the ability of the system to recommend proper number of relevant users to the upcoming events.

Other metrics, such as the running time and the memory consumption, are also used for CARS in EBSNs. Table 3 shows the overview of evaluation metrics of the context-aware recommendation in EBSNs.

**Table 3.** Overview of evaluation metrics of the context-aware recommendation in EBSNs.

| Category | Metrics | | References |
|---|---|---|---|
| Accuracy metrics | Predictive accuracy metrics | RMSE | [21,60] |
| | Classification metrics | Precision | [9,14,21,24,25,28,29,31,32,53,54,59,60,62,65,82,83] |
| | | Recall | [9,14,16,19–21,24,25,32,60,65,66,82,83] |
| | | F1-measure | [21,60] |
| | | Macro-F1 | [21,60] |
| | | HitRate | [21,60] |
| | | MAP | [21,60] |
| | | AUC | [21,60] |
| | Rank accuracy metrics | MRR | [62] |
| | | NDCG | [9,19–21,23,31,62,65,66] |
| | | MIR | [16] |
| Usefulness metrics | Coverage metric | | [29,32,83] |
| | Utility metric | | [37,38] |
| | S-Pearson metric | | [83] |
| Other metrics | running time | | [22,37,38] |
| | memory consumption | | [37,38] |

## 5. Applications of Context-Aware Recommendation in EBSNs

Since the definition of an EBSN was first proposed by Liu et al. in 2012 [8], context-aware recommendation approaches have been applied in a variety of different application scenarios in EBSNs, including event recommendation, group recommendation, event-participant planning, participant predication, group-to-user recommendation, tag-to-group recommendation, venue-to-host recommendation, friend recommendation, and joint event-partner recommendation. We present a brief introduction of these applications, as follows:

- Event recommendation. It is the most widely used application scenario for context-aware recommendation approaches in EBSNs. The goal of event recommendation is recommending interesting or useful events for individual users. Based on users' historical participation records combining with various contextual factors, the techniques we have mentioned in this survey are employed in order to obtain the preferences of users.
- Group recommendation. The goal of group recommendation is recommending a list of events that a group of users may be interested in. The challenge of group recommendation lies in how to aggregate different preferences of group members in order to obtain a consistent group decision under various contexts.
- Event-participant planning. The goal of event-participant planning is to recommend a plan that assigns users to events, such that a predefined objective function is maximized, given sets of users and events, together with constraints like utility scores, travel budgets, and participation upper/lower bounds. The optimal solution of the objective function is usually obtained by heuristic algorithms.
- Participant predication. The goal of participant predication is to predict the possibility of a user attending the given event under various contexts. Its predication result could be used to discover potential participants for event hosts.
- Group-to-user recommendation. Users prefer to join social groups in which members share some common interests. The goal of this recommendation task is to find social groups that a user may be really interested to join.

- Tag-to-group recommendation. Online groups can specify some tags to represent common interests of group members, so this recommendation task takes groups and existing tags as input and returns the most likely tags that the groups may use.
- Venue-to-host recommendation. It recommends a ranked list of venues for hosting a target even. One of its challenges is how to mitigate the scarcity of venue related information in existing data.
- Friend recommendation. The friend recommendation problem is defined as ranking all candidate users for each user. Besides the implicit friendship that is indicated by information about whether a user follows other users, contextual factors, like the visited locations and event participation records, are considered.
- Joint event-partner recommendation. When considering that users prefer to find partners before attending social events, joint event-partner recommendation helps users to simultaneously find their interested events and suitable partners. That is, the goal of event-partner recommendation is to recommend top-*N* event-partner pairs to the target user, so that the user and the recommended partners would like to attend the recommended events together.

We give an overview of above applications, as shown in Table 4.

**Table 4.** Overview of applications of the context-aware recommendation in EBSNs.

| Applications | Techniques | Contextual factors | References |
|---|---|---|---|
| Event recommendation | MF, LTR,<br>probabilistic model,<br>graph-based model,<br>deep learning | text content factor,<br>spatial factor,<br>temporal factor,<br>social factor,<br>other contextual factor | [14,23,24,28,29,31,61,62],<br>[21,25,32,65,83–85] |
| Group recommendation | LTR,<br>probabilistic model,<br>graph-based model,<br>deep learning | text content factor,<br>spatial factor,<br>temporal factor,<br>social factor,<br>network features | [9,19,20,30,33,60,66] |
| Event-participant planning | heuristic-based algorithms | event capacities,<br>spatio-temporal conflicts,<br>travel expenditure,<br>participation lower bound,<br>participation upper bound,<br>utility scores | [37,38,68–74] |
| Participant predication | MF, LTR,<br>graph-based model,<br>deep learning | text content factor,<br>spatial factor,<br>temporal factor,<br>social factor | [17,64,82,86–91] |
| Group-to-user recommendation | MF,<br>graph-based model | spatial factor,<br>temporal factor,<br>social factor | [25,92] |
| Tag-to-group recommendation | graph-based model | temporal factor,<br>network features | [25] |
| Friend recommendation | MF, LTR,<br>probabilistic model | spatial factor,<br>social factor | [53,54] |
| Venue-to-host recommendation | deep learning | spatial factor | [16] |
| Joint event-partner recommendation | graph-based model | text content factor,<br>temporal factor,<br>spatial factor,<br>social factor | [22] |

## 6. Discussion & Future Research Directions

This paper aimed at presenting a survey of the literature of context-aware recommender systems in EBSNs. The results of our study lead us to propose new research directions on this area, as follows:

(1) Modeling the interactional contexts. While a substantial amount of research has already been performed, many existing approaches to CARS in EBSNs focus on the so-called "representational view" that incorporates predefined and static contextual factors (such as temporal and spatial factors) in the recommendation process. Therefore, interactional contexts which are assumed to be underlying and dynamic need to be further explored. This also implies the need of developing mechanisms that can identify contextual changes and incorporate them in the recommendation process in time.

(2) Measuring the influence of contextual factors. One of our review objectives is to identify contextual factors that have been used in CARS. And we have classified the used contextual factors into four main categories including text content factor, temporal factor, spatial factor, and social factor. Existing works focus on discovering different types of contexts and on incorporating them into the recommendation process, yet few of them pay attention to the fact that the influences of these contextual factors on users' preferences are different. Therefore, more efforts are needed on how to measure these influences in order to obtain a more accurate recommendation performance.

(3) Incorporating more types of contexts. With the rapid development of wireless networks and cloud computing service [93–95], increasing amounts of information sensing devices are deployed in cities, communities, and natural environments to help achieve intelligent identification, monitoring, and management. These smart sensing devices including cameras, fire/water sensors, radars, and wearable devices, producing EB-level data every day [96], contribute to the rapid growth of data which could be further exploited by recommender systems. However, these multisource data are heterogeneous and varying, which makes it more challenging to describe them than traditional Web resources, such as texts, images, or video clips. Another type of context worth mentioning is the one related to human psychology. Human psychological or emotional states, such as happiness, anger, and depression, are believed to be a powerful driver of a user's daily decisions by changing his/her judgment and choice. Therefore, during the recommendation process, more attention should be paid on users' psychological states.

(4) Handling the issue of privacy leakage. With the rapid advancements in communication social networks, new technologies and applications bring new security and privacy issues [97,98]. CARS provide users with great convenience by helping users quickly find their interested groups, events, and friends in the social network. However, at the same time, since the efficient recommendation needs to acquire a large amount of contextual information, including users' personal information, such as demographic information, user social relations, etc., users are threatened by the privacy leakage. Therefore, more efforts are needed in order to protect the users' private information while preserving the quality of the recommendations.

(5) Discovering the correlation and mutual influence among contextual factors. There may be correlation and mutual influence among various contextual factors. For example, a rainy day can influence the mood of a user to be depressed and gloomy. Subsequently, (s)he may look for sad music for listening. For another example, temporal factor may influence the semantic of the location. An urban park in the evening may have the topic of "language and culture", because, at that time, the park usually holds events like English Salon, while in the morning the park may have the topic of "sports and fitness". However, most existing studies pay less attention to these correlations and influences, which will in turn influences the accuracy of the prediction of users' preferences. Therefore, more works need to be conducted in this area in the future.

(6) Improving the explainability. Existing works mostly focused on developing complex models to find the most relevant results as efficiently and effectively as possible, and they have achieved important success in promoting the performance of recommendation. However, the explainability of the recommendation models was largely neglected. The lack of explainability for recommendation systems and algorithms, especially those based on deep neural models, makes the users unaware of

why such items are recommended. As a result, the system may not be able to persuade the users to accept the recommended results successfully, which, in turn, decreases the trust of the system. Therefore, research efforts are needed regarding how to make the model itself more explainable and jointly leverage various contextual factors for explainable recommendation.

## 7. Conclusions

This paper presents a literature review on the recommendation process and methods about CARS in EBSNs since the definition of EBSN was first formally given in 2012. Our goal is to help the researchers to understand how the contextual information is incorporated into the recommendation process in order to enhance the recommendation performance. We have identified many contextual factors utilized, among which the text content, temporal, spatial, and social factors are the four main factors to be considered. We also summarized techniques used and categorized them into six categories: MF, LTR, PM, GBM, DL, and HBA. When compared with other techniques, the DL method has gained more attention since its successful application in other fields, like image and natural language processing in recent years. As for the datasets used for the evaluation of recommendation algorithms, we found that the most widely used datasets are created by crawling from two well-known websites, i.e., Meetup and Douban Event. Although a few datasets have been released, like the one in [25], the contextual information in it is limited. Therefore, more efforts need to be paid on building public datasets that are specific to CARS. The widely used evaluation method for recommendation algorithms is conducted though off-line evaluation. Among the evaluation metrics, Precision and Recall are the most used ones. We also found that context-aware recommendation approaches have been applied in a variety of application scenarios, including event recommendation, group recommendation, and event-participant planning, etc. Finally, we tried to give insight into some of the future research directions in this research area.

**Author Contributions:** Conceptualization, G.L. and X.H.; writing–original draft preparation, X.H. and T.L.; Discuss and analyze this paper, and also organize the paper, and polish the writing, G.L., N.X. and A.V.V.; funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Natural Science Foundation of China, grant number 61772245.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lin, C.; Xiong, N.; Park, J.H.; Kim, T.H. Dynamic power management in new architecture of wireless sensor networks. *Int. J. Commun. Syst.* **2009**, *22*, 671–693. [CrossRef]
2. Zeng, Y.; Sreenan, C.J.; Xiong, N.; Yang, L.T.; Park, J.H. Connectivity and coverage maintenance in wireless sensor networks. *J. Supercomput.* **2010**, *52*, 23–46. [CrossRef]
3. Wan, Z.; Xiong, N.; Ghani, N.; Vasilakos, A.V.; Zhou, L. Adaptive unequal protection for wireless video transmission over IEEE 802.11e networks. *Multimed. Tools Appl.* **2014**, *71*, 541–571. [CrossRef]
4. Meetup.com. Available online: https://www.meetup.com/media/ (accessed on 27 September 2020).
5. Dey, A.K. Understanding and using context. *Pers. Ubiquit. Comput.* **2001**, *5*, 4–7. [CrossRef]
6. Dourish, P. What we talk about when we talk about context. *Pers. Ubiquit. Comput.* **2004**, *8*, 19–30. [CrossRef]
7. Adomavicius, G.; Tuzhilin, A. Context-aware recommender systems. In *Recommender Systems Handbook*; Ricci, F., Rokach, L., Shapira, B., Eds.; Springer: Boston, MA, USA, 2015; pp. 191–226.
8. Liu, X.; Qi, H.; Tian, Y.; Lee, W.C.; Mcpherson, J.; Han, J. Event-based social networks: Linking the online and offline social worlds. In Proceedings of the 18th SIGKDD ACM International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 1032–1040.
9. Liao, G.; Huang, X.; Mao, M.; Wan, C.; Liu, X.; Liu, D. Group event recommendation in event-based social networks considering unexperienced events. *IEEE Access* **2019**, *7*, 96650–96671. [CrossRef]
10. Zhang, Y.; Jin, R.; Zhou, Z.H. Understanding bag-of-words model: A statistical framework. *Int. J. Mach. Learn. Cybern.* **2010**, *1*, 43–52. [CrossRef]

11. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. In Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, Vancouver, BC, Canada, 3–8 December 2001; pp. 601–608.
12. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
13. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
14. Gu, Y.; Song, J.; Liu, W.; Zou, L.; Yao, Y. Context aware matrix factorization for event recommendation in event-based social networks. In Proceedings of the 15th IEEE/WIC/ACM International Conference on Web Intelligence (WI), Omaha, NE, USA, 13–16 October 2016; pp. 248–255.
15. Baroni, M.; Dinu, G.; Kruszewski, G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 23–25 June 2014, pp. 238–247.
16. Pramanik, S.; Haldar, R.; Kumar, A.; Pathak, S.; Mitra, B. Deep learning driven venue recommender for event-based social networks. *IEEE Trans. Knowl. Data Eng.* **2019**, [CrossRef]
17. Du, R.; Yu, Z.; Mei, T.; Wang, Z.; Wang, Z.; Guo, B. Predicting activity attendance in event-based social networks: Content, context and social influence. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Seattle, WA, USA, 13–17 September 2014; pp. 425–434.
18. Lin, J. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* **1991**, *37*,145–151. [CrossRef]
19. Yuan, Q.; Cong, G.; Lin, C.Y. COM: A generative model for group recommendation. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 163–172.
20. Du, Y.; Meng, X.; Zhang, Y. CVTM: A content-venue-aware topic model for group event recommendation. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 1290–1303. [CrossRef]
21. Wang, Z.; Zhang, Y.; Chen, H.; Li, Z.; Xia, F. Deep user modeling for content-based event recommendation in event-based social networks. In Proceedings of the 37th IEEE Conference on Computer Communications, Honolulu, HI, USA , 16–19 April 2018; pp. 1304–1312.
22. Yin, H.; Zou, L.; Nguyen, Q.; Huang, Z.; Zhou, X. Joint event-partner recommendation in event-based social networks. In Proceedings of the 34th IEEE International Conference on Data Engineering (ICDE), Paris, France, 16–19 April 2018; pp. 929–940.
23. Macedo, A.Q.; Marinho, L.B.; Santos, R.L.T. Context-aware event recommendation in event-based social networks. In Proceedings of the 9th ACM International Conference on Recommender Systems (RecSys 2015), Vienna, Austria, 16–20 September 2015; pp. 123–130.
24. Wang, Y.; Tang, J. Event2Vec: Learning event representations using spatial-temporal information for recommendation. In Proceedings of the 23rd Pacific-Asia Conference on Knowledge Discovery and Data Mining, Macau, China, 14–17 April 2019; pp. 314–326.
25. Pham, T.N.; Li, X.; Cong, G.; Zhang, Z. A general graph-based model for recommendation in event-based social networks. In Proceedings of the 31st IEEE International Conference on Data Engineering, Seoul, Korea, 13–17 April 2015; pp. 567–578.
26. Attique, M.; Afzal, M.; Ali, F.; Mehmood, I.; Ijaz, M.F.; Cho, H.J. Geo-social top-k and skyline keyword queries on road networks. *Sensors* **2020**, *20*, 3, 1–23. [CrossRef] [PubMed]
27. Attique, M.; Cho, H.J.; Chung, T.S. Efficient processing of moving top-spatial keyword queries in directed and dynamic road networks. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 1–19. [CrossRef]
28. Qiao, Z.; Zhang, P.; Cao, Y.; Zhou, C.; Guo, L.; Fang, B. Combining heterogenous social and geographical information for event recommendation. In Proceedings of the 28th AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014, pp. 145–151.
29. Xu, M.; Liu, S. Semantic-enhanced and context-aware hybrid collaborative filtering for event recommendation in event-based social networks. *IEEE Access* **2019**, *7*, 17493–17502. [CrossRef]
30. Liu, X.; Tian, Y.; Ye, M.; Lee, W.C. Exploring personal impact for group recommendation. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, HI, USA, 29 October 2012; pp. 674–683.

31. Liao, Y.; Lam, W.; Bing, L.; Shen, X. Joint modeling of participant influence and latent topics for recommendation in event-based social networks. *ACM Trans. Inf. Syst.* **2018**, *36*, 29:1–29:31. [CrossRef]
32. Liu, S.; Wang, B.; Xu, M. Event recommendation based on graph random walking and history preference reranking. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 861–864.
33. Tran, V.L.; Pham, N.T.A.; Tay, Y.; Liu, Y.; Cong, G.; Li, X. Interact and decide: Medley of sub-attention networks for effective group recommendation. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 255–264.
34. He, R.; Xiong, N.; Yang, L.T.; Park, J.H. Using multi-modal semantic association rules to fuse keywords and visual features automatically for web image retrieval. *Inf. Fusion* **2011**, *12*, 223–230. [CrossRef]
35. Shu, L.; Zhang, Y.; Yu, Z.; Yang, L.T.; Hauswirth, M.; Xiong, N. Context-aware cross-layer optimized video streaming in wireless multimedia sensor networks. *J. Supercomput.* **2010**, *54*, 94–121. [CrossRef]
36. Yang, Y.; Xiong, N.; Chong, N.Y.; Défago, X. A decentralized and adaptive flocking algorithm for autonomous mobile robots. In Proceedings of the 3rd International Conference on Grid and Pervasive Computing—Workshops, Kunming, China, 25–28 May 2008; pp. 262–268.
37. She, J.; Tong, Y.; Chen, L. Utility-aware social event-participant planning. In Proceedings of the 36th ACM SIGMOD International Conference on Management of Data, Melbourne, VIC, Australia, 31 May–4 June 2015; pp. 1629–1643.
38. Cheng, Y.; Yuan, Y.; Chen, L.; Giraud-Carrier, C.; Wang, G. Complex event-participant planning and its incremental variant. In Proceedings of the 33rd IEEE International Conference on Data Engineering (ICDE), San Diego, CA, USA, 19–22 April 2017; pp. 859–870.
39. Chen, T.; Yin, H.; Chen, H.; Yan, R.; Nguyen, Q.V.H.; Li, X. AIR: Attentional intention-aware recommender systems. In Proceedings of the 35th International Conference on Data Engineering, Macao, China, 8–11 April 2019; pp. 304–315.
40. Massimo, D.; Ricci, F. Harnessing a generalised user behaviour model for next-POI recommendation. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2–7 October 2018; pp. 402–406.
41. Smirnova, E.; Vasile, F. Contextual sequence modeling for recommendation with recurrent neural networks. In Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, Como, Italy, 27–31 August 2017; pp. 2–9.
42. Zhao, P.; Zhu, H.; Liu, Y.; Xu, J.; Li, Z.; Zhuang, F.; Sheng, V.; Zhou, X. Where to go next: A spatio-temporal gated network for next POI recommendation. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, Hawaii, USA, 27 January–1 February 2019; pp. 5877–5884.
43. Zhang, Y.; Wu, H.; Sorathia, V.; Prasanna, V. Event recommendation in social networks with linked data enablement. In Proceedings of the 15th International Conference on Enterprise Information Systems(ICEIS), Angers, France, 4–7 July 2013, pp. 371–379.
44. Li, M.; Huang, D.; Wei, B.; Wang, C.D. Event recommendation via collective matrix factorization with event-user neighborhood. In Proceedings of the 5th International Conference on Intelligent Science and Big Data Engineering, Dalian, China, 22–23 September 2017; pp. 676–686.
45. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30 –37. [CrossRef]
46. Lee, D.; Seung, H. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–91. [CrossRef]
47. Salakhutdinov, R.; Mnih, A. Probabilistic matrix factorization. In Proceedings of the 21th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; pp. 1257–1264.
48. Singh, A.P.; Gordon, G.J. Relational learning via collective matrix factorization. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 650–658.
49. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009; pp. 452–461.

50. Liu.; Tie-Yan. *Learning to Rank for Information Retrieval*; Springer: Berlin, Germany, 2011; pp. 226–245.
51. Amatriain, X.; Basilico, J. Recommender systems in industry: A Netflix case study. In *Recommender Systems Handbook*; Ricci, F., Rokach, L., Shapira, B., Eds.; Springer: Boston, MA, USA, 2015; pp. 385–420.
52. Pan, W.; Xiang, E.W.; Yang, Q. Transfer learning in collaborative filtering with uncertain ratings. In Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; pp. 662–668.
53. Lu, Y.; Qiao, Z.; Zhou, C.; Hu, Y.; Guo, L. Location-aware friend recommendation in event-based social networks: A Bayesian latent factor approach. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 1957–1960.
54. Li, S.; Cheng, X.; Su, S.; Jiang, L. Followee recommendation in event-based social networks. In Proceedings of the 21st International Conference on Database Systems for Advanced Applications(DASFAA), Dallas, TX, USA, 16–19 April 2016; pp. 27–42.
55. Du, Y.; Meng, X.; Zhang, Y.; Lv, P. GERF: A Group event recommendation framework based on learning-to-rank. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 674–687. [CrossRef]
56. Rasmussen, C.E. The infinite Gaussian mixture model. In Proceedings of the 12th International Conference on Neural Information Processing Systems, Denver, CO, USA, 30 November–2 December 1999; pp. 554–560.
57. Gopalan, P.; Charlin, L.; Blei, D.M. Content-based recommendations with poisson factorization. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–11 December 2014; pp. 3176–3184.
58. Hofmann, T.; Puzicha, J. Latent class models for collaborative filtering. In Proceedings of the 16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 31 July–6 August 1999; pp. 688–693.
59. Ji, K.; Chen, Z.; Sun, R.; Ma, K.; Yuan, Z.; Xu, G. GIST: A generative model with individual and subgroup-based topics for group recommendation. *Expert Syst. Appl.* **2018**, *94*, 81–93. [CrossRef]
60. Purushotham, S.; Kuo, C.C.J. Personalized group recommender systems for location- and event-based social networks. *ACM Trans. Spatial Algorithms Syst.* **2016**, *2*, 16:1–16:29. [CrossRef]
61. Yin, H.; Cui, B.; Chen, L.; Hu, Z.; Zhang, C. Modeling location-based user rating profiles for personalized recommendation. *ACM Trans. Knowl. Discov. Data* **2015**, *9*, 1–41. [CrossRef]
62. Zhang, W.; Wang, J. A collective Bayesian Poisson factorization model for cold-start local event recommendation. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 10–13 August 2015; pp. 1455–1464.
63. Boureau, Y.; Le Roux, N.; Bach, F.; Ponce, J.; LeCun, Y. Ask the locals: Multi-way local pooling for image recognition. In Proceedings of 13th International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2651–2658.
64. Wu, X.; Dong, Y.; Shi, B.; Swami, A.; Chawla, N.V. Who will attend this event together? event attendance prediction via deep LSTM networks. In Proceedings of the 18th IEEE International Conference on Data Mining, Singapore, 17–20 November 2018, pp. 180–188.
65. Jhamb, Y.; Ebesu, T.; Fang, Y. Attentive contextual denoising autoencoder for recommendation. In Proceedings of the 41st ACM SIGIR International Conference on Theory of Information Retrieval, Ann Arbor Michigan, MI, USA, 8–12 July 2018; pp. 27–34.
66. Li, R.; Zhu, H.; Fan, L.; Song, X. Hybrid deep framework for group event recommendation. *IEEE ACCESS* **2020**, *8*, 4775–4784. [CrossRef]
67. Long, F.; Xiong, N.; Vasilakos, A.V.; Yang, L.T.; Sun, F. A sustainable heuristic QoS routing algorithm for pervasive multi-layered satellite wireless networks. *Wirel. Netw.* **2010**, *16*, 1657–1673. [CrossRef]
68. Li, K.; Lu, W.; Bhagat, S.; Lakshmanan, L.V.; Yu, C. On social event organization. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 1206–1215.
69. Tong, Y.; She, J.; Meng, R. Bottleneck-aware arrangement over event-based social networks: The max-min approach. *World Wide Web* **2016**, *19*,1151–1177. [CrossRef]
70. She, J.; Tong, Y.; Chen, L.; Song, T. Feedback-aware social event-participant arrangement. In Proceedings of the 2017 ACM International Conference on Management of Data, Chicago, IL, USA, 14–19 May 2017; pp. 851–865.

71. Xin, J.; Li, M.; Xu, W.; Cai, Y.; Lu, M.; Wang, Z. Efficient complex social event-participant planning based on heuristic dynamic programming. In Proceedings of the 23rd International Conference on Database Systems for Advanced Applications (DASFAA), Gold Coast, QLD, Australia, 21–24 May 2018; pp. 264–279.
72. Kou, F.; Zhou, Z.; Cheng, H.; Du, J.; Shi, Y.; Xu, P. Interaction-aware arrangement for event-based social networks. In Proceedings of the 35th IEEE International Conference on Data Engineering (ICDE), Macao, China, 8–11 April 2019; pp. 1638–1641.
73. Li, B.; Cheng, Y.; Wang, G.; Sun, Y. Incremental bilateral preference stable planning over event based social networks. *Complexity* **2019**, *2019*, 1532013:1–1532013:12. [CrossRef]
74. Liang, Y. Complex dynamic event participant in an event-based social network: A three-dimensional matching. *IEEE Access* **2019**, *7*, 144188–144201. [CrossRef]
75. Wang, Y.; Vasilakos, A.V.; Ma, J.; Xiong, N. On studying the impact of uncertainty on behavior diffusion in social Networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 185–197. [CrossRef]
76. Zheng, H.; Guo, W.; Xiong, N. A kernel-based compressive sensing approach for mobile data gathering in wireless sensor network systems. *IIEEE Trans. Syst. Man, Cybern. Syst.* **2018**, *48*, 2315–2327. [CrossRef]
77. Li, J.; Xiong, N.; Park, J.H.; Liu, C.; Shihua, M.A.; Cho, S. Intelligent model design of cluster supply chain with horizontal cooperation. *J. Intell. Manuf.* **2012**, *23*, 917–931. [CrossRef]
78. Guo, W.; Xiong, N.; Vasilakos, A.V.; Chen, G.; Yu, C. Distributed k-connected fault-tolerant topology control algorithms with PSO in future autonomic sensor systems. *Int. J. Sens. Netw.* **2012**, *12*, 53–62. [CrossRef]
79. Fang, W.; Li, Y.; Zhang, H.; Xiong, N.; Lai, J.; Vasilakos, A.V. On the throughput-energy tradeoff for data transmission between cloud and mobile devices. *Inf. Sci.* **2014**, *283*, 79–93. [CrossRef]
80. Herlocker, J.L.; Konstan, J.A.; Terveen, L.G.; Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **2004**, *22*,5–53. [CrossRef]
81. Sassi, I.B.; Mellouli, S.; Yahia, S.B. Context-aware recommender systems in mobile environment: On the road of future research. *Inf. Syst.* **2017**, *72*, 27–61. [CrossRef]
82. Zhang, S.; Lv, Q. Hybrid EGU-based group event participation prediction in event-based social networks. *Knowl. Based Syst.* **2018**, *143*, 19–29. [CrossRef]
83. 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.* **2017**, *79*, 383–395. [CrossRef]
84. Boutsis, I.; Karanikolaou, S.; Kalogeraki, V. Personalized event recommendations using social networks. In Proceedings of the 16th IEEE International Conference on Mobile Data Management, Pittsburgh, PA, USA, 15–18 June 2015, pp. 84–93.
85. Wang, Z.; Zhang, Y.; Li, Y.; Wang, Q.; Xia, F. Exploiting social influence for context-aware event recommendation in event-based social networks. In Proceedings of the 36th IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
86. Jiang, J.Y.; Li, C.T. Who should I invite: Predicting event participants for a host user. *Knowl. Inf. Syst.* **2019**, *59*, 629–650. [CrossRef]
87. Yu, Z.; Du, R.; Guo, B.; Xu, H.; Gu, T.; Wang, Z.; Zhang, D. Who should I invite for my party? Combining user preference and influence maximization for social events. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Umeda, Osaka, Japan, 7–11 September 2015; pp. 879–883.
88. Zhang, X.; Zhao, J.; Cao, G. Who will attend?—Predicting event attendance in event-based social network. In Proceedings of the 16th IEEE International Conference on Mobile Data Management, Pittsburgh, PA, USA, 15–18 June 2015; pp. 74–83.
89. Ding, H.; Yu, C.; Li, G.; Liu, Y. Event participation recommendation in event-based social networks. In Proceedings of the 7th International Conference on Social Informatics, Beijing, China, 9–12 December 2016, pp. 361–375.
90. Lu, M.; Dai, Y.; Zhong, H.; Ye, D.; Yan, K. MFDAU: A multi-features event participation prediction model for users of different activity levels. *Phys. A* **2019**, *534*, 122244. [CrossRef]
91. Trinh, T.; Nguyen, N.T.; Wu, D.; Huang, J.; Emara, T. A new location-based topic model for event attendees recommendation. In Proceedings of the 13th IEEE-RIVF International Conference on Computing and Communication Technologies, Danang, Vietnam, 20–22 March 2019, pp. 1–6.

92. Zhang, W.; Wang, J.; Feng, W. Combining latent factor model with location features for event-based group recommendation. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 910–918.
93. Wang, Z.; Li, T.; Xiong, N.; Pan, Y. A novel dynamic network data replication scheme based on historical access record and proactive deletion. *J. Supercomput.* **2012**, *62*, 227–250. [CrossRef]
94. Xiong, N.; Vasilakos, A.V.; Wu, J.; Yang, Y.R.; Rindos, A.; Zhou, Y.; Song, W.Z.; Pan, Y. A self-tuning failure detection scheme for cloud computing service. In Proceedings of the 26th IEEE International Parallel and Distributed Processing Symposium, Shanghai, China, 21–25 May 2012; pp. 668–679.
95. Lin, C.; He, Y.X.; Xiong, N. An energy-efficient dynamic power management in wireless sensor networks. In Proceedings of the 5th International Symposium on Parallel and Distributed Computing, Timisoara, Romania, 6 July 2006; pp. 148–154.
96. Zhou, Y.; Zhang, D.; Xiong, N. Post-cloud computing paradigms: A survey and comparison. *Tsinghua Sci. Technol.* **2017**, *22*, 714–732. [CrossRef]
97. Sang, Y.; Shen, H.; Tan, Y.; Xiong, N. Efficient protocols for privacy preserving matching against distributed datasets. In Proceedings of the 8th International Conference on Information and Communications Security, Raleigh, NC, USA, 4–7 December 2006; pp. 210–227.
98. Liu, Y.; Ma, M.; Liu, X.; Xiong, N.N.; Liu, A.; Zhu, Y. Design and analysis of probing route to defense sink-hole attacks for Internet of Things security. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 356–372. [CrossRef]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).