Wang G, Li X, Guo ZY *et al.* SMEC: Scene mining for E-commerce. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(1): 192-210 Jan. 2024. DOI: 10.1007/s11390-021-1277-0

SMEC: Scene Mining for E-Commerce

Gang Wang[1] (王 罡), Xiang Li[2] (李 翔), *Member, CCF, ACM, IEEE*, Zi-Yi Guo[3] (郭子义)
Da-Wei Yin[4] (殷大伟), *Member, ACM, IEEE*, and Shuai Ma[1,*] (马 帅), *Member, CCF, ACM, IEEE*

[1] *State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China*

[2] *School of Data Science and Engineering, East China Normal University, Shanghai 200062, China*

[3] *JD.com, Inc., Beijing 100176, China*

[4] *Baidu, Inc., Beijing 100085, China*

E-mail: iegwang@buaa.edu.cn; xiangli@dase.ecnu.edu.cn; guoziyi@jd.com; yindawei@acm.org; mashuai@buaa.edu.cn

Received January 8, 2021; accepted October 24, 2021.

Abstract Scene-based recommendation has proven its usefulness in E-commerce, by recommending commodities based on a given scene. However, scenes are typically unknown in advance, which necessitates scene discovery for E-commerce. In this article, we study scene discovery for E-commerce systems. We first formalize a scene as a set of commodity categories that occur simultaneously and frequently in real-world situations, and model an E-commerce platform as a heterogeneous information network (HIN), whose nodes and links represent different types of objects and different types of relationships between objects, respectively. We then formulate the scene mining problem for E-commerce as an unsupervised learning problem that finds the overlapping clusters of commodity categories in the HIN. To solve the problem, we propose a non-negative matrix factorization based method SMEC (Scene Mining for E-Commerce), and theoretically prove its convergence. Using six real-world E-commerce datasets, we finally conduct an extensive experimental study to evaluate SMEC against 13 other methods, and show that SMEC consistently outperforms its competitors with regard to various evaluation measures.

Keywords graph clustering, E-commerce, heterogeneous information network (HIN), scene mining

1 Introduction

The recent development of E-commerce platforms has witnessed the fast advancement of recommendation. Most conventional recommendation methods in E-commerce recommend substitutable or complementary commodities to users. However, the seemingly irrelevant commodities sometimes form a commodity set that is meaningful under a specific scene. For example, printers, trash bins and humidifiers seem irrelevant commodities, but they can be recommended to users simultaneously under a scene called "Daily Office". Such a recommendation method is called scene-based recommendation, which is of great use in E-commerce platforms. By incorporating scenes into recommendation systems, the recommendation diversity may stimulate more potential purchase intention and alleviate uncomfortable user experiences (e.g., recommending a user who has recently purchased a printer with even cheaper and better printers, which could make the user feel bored).

Recommendation in E-commerce has been well studied[1–3], but there are few studies on the scene-based recommendation due to the difficulty of obtaining scenes. For example, Shop the Look[4, 5] and Complete the Look[6] recommend substitutable or complementary products of a specific category that visually match the given scene in the form of images. However, in these tasks, scenes are given as input, which may not be available in the E-commerce scenarios. Moreover, scene mining is a non-trivial task and is costly by human labors. To facilitate the downstream

Regular Paper

The work was supported by the National Key Research and Development Program of China under Grant No. 2018AAA0102301 and the National Natural Science Foundation of China under Grant No. 61925203.

*Corresponding Author

©Institute of Computing Technology, Chinese Academy of Sciences 2024

recommendation tasks, an effective scene mining method for E-commerce is needed, where a scene is defined as a set of elements that occur simultaneously and frequently in a real-world situation. The elements of the scenes should achieve a balance between the expressive ability of the situations and the adaptability of the E-commerce platforms. Since concrete commodities are frequently updated in E-commerce platforms, the scenes consisting of commodities have a huge volume, and need to be updated within a short period. To avoid the above troubles, the elements of scenes are defined as a set of categories, where a category represents a group of products with the same function. When a commodity updates or stops selling in a group, the meaning of its corresponding category does not change, which insulates the frequent changes of scenes. This also makes scenes more abstractive and provides enough information with a small amount of categories.

An E-commerce platform contains rich and complex information, which can be modeled as a heterogeneous information network (HIN). An HIN is a network whose nodes and edges represent different types of objects and different types of relationships between objects, respectively. Based on user behaviors and commodity information, a constructed E-commerce HIN contains various types of entities like users, commodities, brands, and categories. Different relationships exist between these entities. For example, users can browse, search, and purchase commodities. The E-commerce HIN also needs to include the commodity category as an object type to constitute the scenes to be discovered.

The challenges of scene mining in large E-commerce platforms are fourfold.

1) *Interpretability.* A scene represents a real-life situation, and thus the derived set of commodity categories should be practically interpretable as an indicator of a specific life situation. In contrast, a set of categories that fails to indicate any real-life situations is not a scene. For example, a set of routers, switches, cables, and 3G/4G devices strongly indicates the scene of "Networking" while a set of routers, switches, and hats cannot be interpreted as a scene.

2) *Generality.* E-commerce platforms can be modeled as HINs. Even though meta-paths[7] and meta-structures[8, 9] show their usefulness in a variety of data mining tasks in HINs[10, 11], most of these HINs are simple, owing to the fact that meta-paths and meta-structures are generally given by human experts, and it is not practically feasible to generate them in large and complex HINs. Therefore, a scene mining method that can be generalized to HINs with varying complexities is desired.

3) *Multiplicity.* In E-commerce, a commodity category can exist in multiple scenes and a scene can be represented by multiple sets of categories. For example, trash bins can occur in both scenes "Daily Office" and "Family Life"; scene "Daily Office" can be represented by either a set of printers, trash bins and humidifiers or a set of computers, pens and printers.

4) *Learnability.* Scenes in E-commerce are usually unknown and the supervision information to guide the process of finding scenes is typically missing. Therefore, mining scenes is better treated as an unsupervised learning problem.

In this article, we study scene mining for E-commerce to address the challenges mentioned above. We first construct an E-commerce HIN based on user behaviors and commodity attributes, and formulate scene mining for E-commerce as an overlapping graph clustering problem in HINs. As non-negative matrix factorization (NMF) has shown to generate superior clusters that are easy to interpret[12], NMF-based methods are easier to discover the scenes which correspond to real-life situations. Moreover, some existing network embedding methods have advanced performance to abstract information from the networks by performing matrix factorization implicitly[13]. Hence, we propose a non-negative matrix factorization based method SMEC (Scene Mining for E-Commerce) to aggregate these advantages. It jointly factorizes a set of weight matrices derived from the input HIN. The derived factorization matrix of the commodity categories is taken as the clustering membership matrix to discover scenes. In the end, our objective is to shed light on how to discover scenes in E-commerce and provide valuable scenes for the downstream scene-based recommendation. Our main contributions can be summarized as follows.

1) To the best of our knowledge, we are among the first to study scene mining for E-commerce. We formally define a scene as a set of commodity categories, model E-commerce platforms as HINs, and formulate scene mining as an unsupervised learning problem that finds a set of overlapping clusters of the commodity categories in HINs.

2) We propose a matrix factorization based method SMEC and theoretically prove its convergence. Since a commodity category only exists in a

small number of scenes in practice, we add an $\ell_{2,1}$-norm to regularize the category representation to avoid the overfitting caused by the sparsity of scenes and post-processing.

3) We conduct extensive experiments to evaluate the performance of SMEC against 13 other methods. We find that our method SMEC is effective. Specifically, SMEC on average improves the three metrics, average $F1$, NMI, Omega index, over the baselines by 5.2%, 22.4%, and 189.5% on four JD datasets, respectively. Moreover, the results on two Amazon datasets show that scenes discovered by SMEC are practically meaningful.

The rest of this article is organized as follows. Section 2 introduces related work. Section 3 formally defines scenes and the scene mining for E-commerce. Section 4 discusses the construction of an E-commerce HIN, and presents our SMEC method and its optimization. Section 5 shows the experimental results, followed by conclusions in Section 6.

2 Related Work

Recommendation is a fundamental task in E-commerce and it has been widely studied[1–3, 14, 15], where most existing methods focus on recommending substitutable or complementary commodities. Recently, scene-based recommendation has received great attention, and aims to recommend commodities that co-occur in a scene. For example, a method was proposed to model the interaction between users and items to predict user preference with the help of the scenes[16]. However, the model assumes that the scene has been given as input, which is not always available. To bridge this gap, we study the scene mining problem to automatically extract recommendation scenes, and formulate it as a clustering problem in an E-commerce HIN.

As E-commerce scenes are regarded as overlapping clusters that are composed of commodity categories, the scene mining can be formulated as an overlapping graph clustering task. An overview on the overlapping community detection in networks can be found in [17]. The traditional overlapping clustering method Fuzzy C-Means (FCM)[18] is a soft version of k-means that allows each object to be assigned to multiple membership. There also exist several non-negative matrix factorization (NMF)[19, 20] based methods[21–24] to detect overlapping clusters. Most of them design the model for the specific concrete matrices, such as an adjacency matrix or a combination of a node attribute matrix and an adjacency matrix. They are unsuitable for the data features of scene mining that involves complex networks with multiple node types and edge types. Our method represents the algorithm by more abstract element type classes and could mine scenes on the datasets with different components. SMEC has good scalability for the networks in E-commerce platforms compared with them. Another type of clustering methods[25–28] builds generative models based on the assumption that each edge in the network is generated with a probability with respect to the clustering membership vectors of two node objects. Moreover, there are methods to find overlapping clusters in HINs[29–32], most of which use meta-paths/meta-structures, and can hardly be applied to complex HINs, as the generation of meta-paths/meta-structures is an obstacle. Also, a number of NMF-based clustering methods were proposed for HINs. MultiNMF[12] factorizes the matrices jointly which involve the objective elements, and minimizes the difference between the common consensus matrix and each objective element's matrix factor to extract disjoint clusters. It ignores those matrices that are not directly related to the objective elements but contain deeper-level information, especially in the E-commerce platforms. The non-objective elements could also appear in multiple matrices, where they need to be integrated in the factorization. Our method aggregates the information from multiple perspectives and keeps the interactions between the related objects by unifying the factorization matrix of the objects (not only limited to objective elements) to take full advantage of E-commerce HINs. Hence, our method not only suits for scene mining, but also has good scalability for clustering applications that contain complex networks with different representations or views.

Graph partition is the reduction of a graph to smaller graphs by partitioning its set of nodes into mutually exclusive groups for simplifying graph analysis, where its objective is minimizing the number of edges between groups[33]. Scene mining aims to put E-commerce elements with the similar characteristics into overlapping clusters, and it focuses on nodes rather than edges. In the task of scene mining, the number of edges is not the only factor to affect clustering, because the number of edges between nodes could not fully reflect their relationships, especially in complex networks.

Topic detection is a process from the concrete to the abstract, which extracts a set of words to generalize the main meaning of the articles or news[34], while scene mining discovers the corresponding real-life situation represented by E-commerce elements as comprehensive as possible. Specifically, the change of words and their meanings is far less frequent than that of commodity information[35]. Topic types are essentially weighted sets of keywords, where weights represent the importance of the keywords, but each commodity in a scene has an equal contribution.

Our work is also related to graph embedding (network representation)[36–40], which has attracted great attentions recently. Graph embedding represents objects with low-dimensional embedding vectors to encode the structural connectivity of objects in the network. These embedding vectors can then be fed into the downstream data mining tasks, such as node classification and link prediction. For homogeneous information networks, DeepWalk[41] and node2vec[42] generate random walks to capture the neighborhood information to learn node feature vectors. HIN2vec[43] and metapath2vec[44] are two representative methods that embed node objects in HINs, and they capture the specific semantic features via predefined meta-paths. There are also graph neural network based methods to learn the network representation, such as GCN[45] and GraphSAGE[46]. Their general idea is that the current representation of a node can be updated by aggregating representations of its neighbors. In the scene mining, cluster analyses can be performed on the embedding vectors via a post-processing step, such as FCM[18].

3 Problem Formulation

In this section, we formally introduce basic concepts and the problem to be studied.

Definition 1 (Heterogeneous Information Network, HIN)[47]. *Let* $\mathcal{T}=\{T_1,T_2,...,T_m\}$ *be a set of* m *object types and* $\mathcal{R}=\{R_1,R_2,...,R_k\}$ *be a set of* k *edge types. For each type* $T_i\in\mathcal{T}$, *let* V_i *be the set of objects of type* T_i. *For each type* $R_j\in\mathcal{R}$, *let* E_j *be the set of edges of type* R_j. *An information network is a graph* $G=(V,E)$ *with a node type mapping* $\varphi: V\rightarrow\mathcal{T}$ *and an edge type mapping* $\psi: E\rightarrow\mathcal{R}$, *where* $V=\bigcup_{i=1}^{m}V_i$ *is the set of nodes,* $E=\bigcup_{j=1}^{k}E_j$ *is the set of edges, and* $E_j=\{e_{pq}^{(j)}|x_p,x_q\in V\}$. *Each* $e_{pq}^{(j)}$ *represents a binary relation for type* R_j *between two objects* x_p *and* x_q *in* V. *When* $|\mathcal{T}|>1$ *or* $|\mathcal{R}|>1$, G *is called an HIN*[48].

A toy E-commerce HIN example is illustrated in Fig.1, which includes three object types $\mathcal{T}=\{Commodity, Category, Brand\}$. There are also five relations in $\mathcal{R}$ that are highlighted in different lines, e.g., the relation between *Commodity* and *Brand* carries the information "which commodity has which brand".

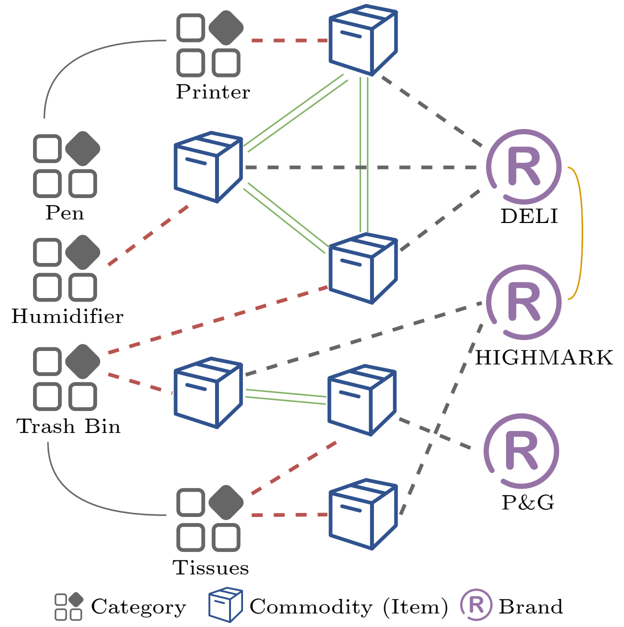


Fig.1. E-commerce HIN example, where both commodity attributes (e.g., category and brand) and user behaviors (reflected by different links between commodities) are incorporated.

Definition 2 (Scene). *An E-commerce scene is a set of commodity categories that occur simultaneously and frequently in a real-life situation, denoted as* $S=\{x_1,x_2,...,x_{|S|}|x_i\in V_c, 1\leqslant i\leqslant |S|\}$, *where* V_c *is a set of commodity categories and* $|S|\geqslant 1$. *Let* $\mathcal{S}=\{S_1,S_2,...,S_{|\mathcal{S}|}\}$ *be a set of scenes,* $\mathcal{L}$ *be a set of real-life situations, and* $\phi:\mathcal{S}\rightarrow\mathcal{L}\bigcup\emptyset$ *be a mapping function that maps a scene into a real-life situation. An E-commerce scene satisfies the following properties: 1) if a scene* S *cannot indicate any real-life situation,* $\phi(S)=\emptyset$, *and 2) two scenes* S_1 *and* S_2 *are the same scene if and only if* $\phi(S_1)=\phi(S_2)$.

In Fig.1, printers, trash bins and humidifiers are commonly seen in daily working offices; thus they form a scene S called "Daily Office". A scene is flexible in that different sets of categories can be used to represent the same real-life situation. We can add pens to S, and S still represents "Daily Office". However, adding/deleting a category in a scene may lead to a completely different scene. For example, deleting printers from S obtains a new scene called "Family Life".

Definition 3 (Scene Mining for E-Commerce). *Given an E-commerce HIN* $G=(V,E)$ *and the number of scenes* r, *the problem of scene mining in* G *is to par-*

tition the commodity categories in G into a set of overlapping nonempty clusters $\hat{\mathcal{S}}=\{S_1,S_2,...,S_r\}$.

The notations used are summarized in Table 1.

Table 1. Summary of Symbols

Symbol	Description
$\mathcal{T}$	Set of node types $\{T_i\}$
$\mathcal{R}$	Set of edge types
V, E	Set of nodes and edges, respectively
$\mathcal{A}$	Set of commodity attributes $\{A_i\}$
$\mathcal{B}$	Set of user behaviors $\{B_i\}$
$\mathcal{L}$	Set of real-life situations
$\mathcal{S}$	Set of scenes $\{S_i\}$
G_{AA}	Attribute-attribute network of a given attribute A
G_{BB}	Item-item network of a given behavior B
G_{IA}	Item-attribute network of a given attribute A
φ,ψ	Node and edge type mapping, respectively
ϕ	Mapping function that maps a scene into a real-life situation
$\boldsymbol{W}_{pq}$	Weight matrix between objects of types T_p and T_q in a sub-network
$\boldsymbol{H}_p$	Low-dimensional factorization matrix for objects of type T_p
n_p	Number of objects of type T_p
r	Number of scenes
α,β	Hyperparameters
$(\boldsymbol{H}_p)_{ij},h$	The (i,j)-th entry of $\boldsymbol{H}_p$
h^t,h^{t+1}	Entry h in the t-th iteration and the $(t+1)$-th iteration, respectively

4 Scene Mining for E-Commerce

In this section, we present our method SMEC (Scene Mining for E-Commerce). We first introduce how to construct an E-commerce HIN based on user behaviors and commodity attributes. We then introduce the details of SMEC, together with its optimization technique and theoretical convergence analysis.

4.1 Construction of an E-Commerce HIN

We construct an E-commerce HIN using user behaviors and commodity attributes. For a better illustration, we split the HIN into different sub-networks that can be divided into three categories.

Item-Item Network. This kind of networks describes the relationships between commodities (items). We interchangeably use commodity and item in this article. In E-commerce, there are various user behaviors that can be utilized to capture relationships between items, such as clicking, purchasing, and commenting. For example, items can co-occur in different user behaviors, such as "users who clicked item x_1 also clicked x_2" and "users who purchased item x_1 also purchased x_2". Let $\mathcal{B}=\{B_1,B_2,...,B_{|\mathcal{B}|}\}$ be a set of user behaviors. Given a behavior B, an item-item network is defined as $G_{BB}=(V_B,E_{BB})$, where V_B is a set of items and E_{BB} is a set of edges between items. Each edge represents a co-occurrence relation of two linked items in behavior B. Moreover, an adjacency matrix $\boldsymbol{W}_{BB}$ is used to measure the relevance between items.

Item-Attribute Network. This network depicts the relationships between items and attributes. In E-commerce, each item has a set of attributes, such as *brand* and *category*. We model these attributes as objects in an E-commerce HIN. Let $\mathcal{A}=\{A_1,A_2,...,A_{|\mathcal{A}|}\}$ be a set of commodity attributes, where each attribute A_i is considered as an object type. Given an attribute A, an item-attribute network can be represented by a bipartite graph $G_{IA}=(V_I\cup V_A,E_{IA})$, where V_I is a set of items, V_A is a set of attribute nodes, and E_{IA} is a set of links that are binary relations between items and attributes. Each item-attribute network can also be represented by using an adjacency matrix $\boldsymbol{W}_{IA}$.

Attribute-Attribute Network. A network in this kind describes the relations between commodity attributes. For example, the shoe brand "NIKE" is highly relevant to "ADIDAS" but not to the luxury brand "CHANEL". To enrich the E-commerce HIN, we construct the attribute-attribute network. Given an attribute A, an attribute-attribute network is defined as $G_{AA}=(V_A,E_{AA})$, where V_A is a set of attribute nodes and E_{AA} is a set of links between attributes. Moreover, we use an adjacency matrix $\boldsymbol{W}_{AA}$ to measure the relevance between attributes.

Given a set of user behaviors $\mathcal{B}$ and a set of commodity attributes $\mathcal{A}$, we construct an E-commerce HIN that comprises a set of sub-networks of the three categories above. It ought to include the commodity category as an object type that forms scenes to be discovered. Fig.2 shows an HIN from multiple user behaviors and commodity attributes for scene mining where the commodity category is treated as a special attribute.

4.2 SMEC Framework

We propose a non-negative matrix factorization-based method SMEC to discover E-commerce scenes. Since each scene consists of a set of commodity cate-

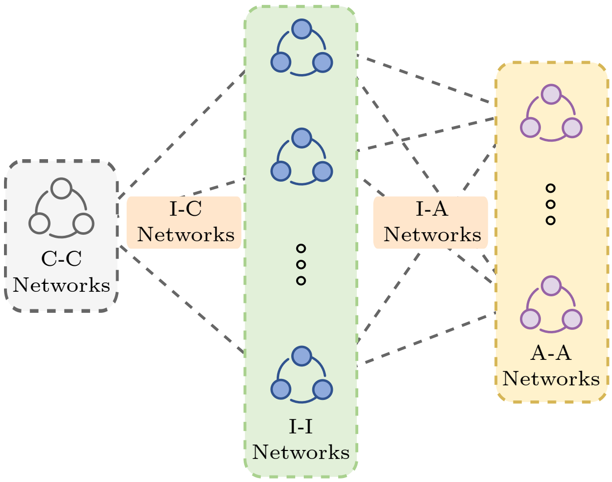


Fig.2. Illustrative HIN constructed from commodity attributes and user behaviors can be split into attribute-attribute (A-A), item-attribute (I-A), item-item (I-I), item-category (I-C) and category-category (C-C) networks, where commodity categories as an object type is essential for scene mining.

gories, we formulate scene mining as an unsupervised learning problem that clusters scenes in an HIN. SMEC is based on the NMF framework. On the one hand, NMF has been shown to be effective for a variety of data mining and machine learning tasks including clustering[49], recommendation[50], and representation learning[13]. We can use NMF to generate low-dimensional representations for commodity categories. Categories that are close in the low-dimensional space are more likely to form a scene. On the other hand, it is theoretically shown that some existing network representation methods are implicitly performing matrix factorization[13], which further motivates us to put forward a matrix factorization based method.

An E-commerce HIN contains various object types that are correlated with each other. To discover scenes, we should consider the whole HIN rather than only the sub-networks related to commodity categories. Given an HIN, we first split it into a set of sub-networks and then jointly factorize the adjacency matrices associated with these networks. In this way, the interaction between objects of different types can be well captured, and each object is embedded into a shared low-dimensional space. Moreover, in E-commerce platforms, each commodity category usually exists in only a few E-commerce scenes. To encode such sparsity information, we regularize the low-dimensional representation of commodity categories by an $\ell_{2,1}$-norm, which ensures a sparse representation for each commodity category. The low-dimensional representation matrix is considered as the clustering membership matrix for commodity categories, and each dimension in the matrix is regarded as a scene. We set a threshold ϵ to filter scenes. For the dimension where all the category objects have a value smaller than ϵ, it corresponds to an invalid scene that is discarded. Only valid scenes are returned. Therefore, one advantage of the sparse representation is to enable SMEC to directly encode the scene information in embedding vectors. Moreover, these sparse representation vectors can also be taken as objects' feature vectors, and a post-processing step like Fuzzy C-Means[18] can be used to discover scenes.

Given an HIN $G = (V, E)$, we can derive a set of item-item networks, item-attribute networks, and attribute-attribute networks based on various user behaviors and commodity attributes. For notation convenience, let $\boldsymbol{W}_{pq}$ denote the adjacency matrix between objects of types T_p and T_q in a sub-network. Since a kind of vertices is always involved in multiple sub-networks, we design an NMF-based method to incorporate the information of the same elements (items or attributes) from different types of relations into the corresponding embedding vectors. The proposed method jointly factorizes the weight matrices associated with these sub-networks and the objective function is given as:

$$\mathcal{O} = \frac{1}{2} \sum_{T_p,\ T_q \in \mathcal{T}} \left\| \boldsymbol{W}_{pq} - \boldsymbol{H}_p \boldsymbol{H}_q^{\mathrm{T}} \right\|_F^2, \tag{1}$$

where $\boldsymbol{H}_p$ is the low-dimensional factorization matrix for objects of type T_p, and $\|\cdot\|_F$ denotes Frobenius-norm. By unifying the factorization matrix of the same objects among different adjacency matrices of sub-networks, our method aggregates the information from multiple perspectives and keeps the interaction between related objects. In order to avoid the inaccuracy caused by the extra post-processing step to obtain the final results, the objective should also take generating scenes directly into consideration. Since each commodity category usually exists in a very small number of scenes in practice, an $\ell_{2,1}$-norm is added to regularize the category representation $\boldsymbol{H}_c$. To prevent overfitting, we further add the following regularization term to (1):

$$\mathcal{O}_{\mathcal{R}} = \frac{\alpha}{2} \sum_{T_p \in \mathcal{T}} \|\boldsymbol{H}_p\|_F^2 + \beta \|\boldsymbol{H}_c\|_{2,1}, \tag{2}$$

where α and β are the coefficients that control the two regularization terms. $\|\cdot\|_{2,1}$ is the $\ell_{2,1}$-norm that regularizes the commodity category representation $\boldsymbol{H}_c$ to keep it sparse, which can achieve the goal of directly obtaining the scenes. Finally, we minimize the objective function $\mathcal{O}$ subject to the constraints $\boldsymbol{H}_p \geqslant 0$.

4.3 Optimization and Convergence

Our objective is to find the low-dimensional representation matrices $\{\boldsymbol{H}_p\}_{p=1}^{|\mathcal{T}|}$ that minimize the objective function $\mathcal{O}$. Inspired by [19], SMEC learns these parameters by using the multiplicative update rules. Based on the method of Lagrange multipliers and the Karush-Kuhn-Tucher (KKT) condition[51], we get the following update rules:

$$(\boldsymbol{H}_p)_{ij} \leftarrow (\boldsymbol{H}_p)_{ij} \frac{(\boldsymbol{W}_{pp}^{\mathrm{T}}\boldsymbol{H}_p + \boldsymbol{W}_{pp}\boldsymbol{H}_p + \sum\limits_{T_q \in \mathcal{T_A}} \boldsymbol{W}_{pq}\boldsymbol{H}_q)_{ij}}{(2\boldsymbol{H}_p\boldsymbol{H}_p^{\mathrm{T}}\boldsymbol{H}_p + \sum\limits_{T_q \in \mathcal{T_A}} \boldsymbol{H}_p\boldsymbol{H}_q^{\mathrm{T}}\boldsymbol{H}_q + \alpha\boldsymbol{H}_p)_{ij}}, \quad T_p \in \mathcal{T_B}, \tag{3}$$

$$(\boldsymbol{H}_p)_{ij} \leftarrow (\boldsymbol{H}_p)_{ij} \frac{(\boldsymbol{W}_{pp}^{\mathrm{T}}\boldsymbol{H}_p + \boldsymbol{W}_{pp}\boldsymbol{H}_p + \sum\limits_{T_q \in \mathcal{T_B}} \boldsymbol{W}_{qp}^{\mathrm{T}}\boldsymbol{H}_q)_{ij}}{(2\boldsymbol{H}_p\boldsymbol{H}_p^{\mathrm{T}}\boldsymbol{H}_p + \sum\limits_{T_q \in \mathcal{T_B}} \boldsymbol{H}_p\boldsymbol{H}_q^{\mathrm{T}}\boldsymbol{H}_q + \alpha\boldsymbol{H}_p)_{ij}}, \quad T_p \in \mathcal{T_A} \backslash T_c, \tag{4}$$

$$(\boldsymbol{H}_p)_{ij} \leftarrow (\boldsymbol{H}_p)_{ij} \frac{(\boldsymbol{W}_{pp}^{\mathrm{T}}\boldsymbol{H}_p + \boldsymbol{W}_{pp}\boldsymbol{H}_p + \sum\limits_{T_q \in \mathcal{T_B}} \boldsymbol{W}_{qp}^{\mathrm{T}}\boldsymbol{H}_q)_{ij}}{(2\boldsymbol{H}_p\boldsymbol{H}_p^{\mathrm{T}}\boldsymbol{H}_p + \sum\limits_{T_q \in \mathcal{T_B}} \boldsymbol{H}_p\boldsymbol{H}_q^{\mathrm{T}}\boldsymbol{H}_q + \alpha\boldsymbol{H}_p + \beta\boldsymbol{H}_p\boldsymbol{U})_{ij}}, \quad T_p = T_c, \tag{5}$$

where $\mathcal{T_B}$ is a set of item types corresponding to different user behaviors, $\mathcal{T_A}$ is a set of attribute types, and $T_c \in \mathcal{T_A}$ represents the commodity category. Let $(\boldsymbol{H}_p)_{ij}$ denote the (i,j)-th entry of $\boldsymbol{H}_p$ and $\boldsymbol{U}$ denote a diagonal matrix, where the j-th element on the diagonal $(\boldsymbol{U})_{jj} = 1/\|(\boldsymbol{H}_c)_{\cdot j}\|_2$ and $(\boldsymbol{H}_c)_{\cdot j}$ is the j-th column of $\boldsymbol{H}_c$.

Theorem 1. *The objective function $\mathcal{O}$ is non-increasing under the update rules* (3), (4), *and* (5).

Inspired by prior work[32], we provide the proof for Theorem 1 as follows.

Proof. For rule (3), let h denotes the (i,j)--th entry in $\boldsymbol{H}_p$ where $T_p \in \mathcal{T_B}$.. We use F to denote the part of $\mathcal{O}$ which is only relevant to h.. Since our update is essentially element-wise, it is sufficient to show that each F is non-increasing under the update rule.

We first construct an auxiliary function $G(h^t,h)$ for F as

$$G(h^t,h) = F(h^t) + F'(h^t)(h-h^t) + \frac{1}{2}K(h^t)(h-h^t)^2,$$

where h^t is the (i,j)-th entry of $\boldsymbol{H}_p$ in the t-th iteration, F' is the derivative of F, and $K(h^t) = (2\boldsymbol{H}_p{\boldsymbol{H}_p}^{\mathrm{T}}\boldsymbol{H}_p + \sum_{T_q \in \mathcal{T_A}} \boldsymbol{H}_p{\boldsymbol{H}_q}^{\mathrm{T}}\boldsymbol{H}_q + \alpha\boldsymbol{H}_p)_{ij}/h^t$. Then it is easy to show that $G(h^t,h)$ satisfies 1) $G(h^t,h^t) = F(h^t)$ and 2) $G(h^t,h) \geqslant F(h)$.

Let $h^{t+1} = \arg\min_h G(h^t,h)$. Since $G(h^t,h)$ is convex with respect to h, the minimum can be achieved by setting $(\partial G(h^t,h))/(\partial h) = 0$, that is

$$\frac{\partial}{\partial h}G(h^t,h) = F'(h^t) + K(h^t)(h-h^t) = 0,$$
$$h^{t+1} \leftarrow h = h^t - \frac{F'(h^t)}{K(h^t)},$$

where h^{t+1} exactly equals the righthand side in rule (3). Based on properties (1), (2), and h^{t+1}, we have

$$F(h^t) = G(h^t,h^t) \geqslant G(h^t,h^{t+1}) \geqslant F(h^{t+1}).$$

It follows that setting h^t to h^{t+1} does not increase the objective function $\mathcal{O}$ under the update rule (3). Along the same lines, we can also prove that $\mathcal{O}$ is non-increasing under update rules (4) and (5). □

The convergence of SMEC can be proved based on Theorem 1. SMEC alternatively updates each $\boldsymbol{H}_p$ with the other matrices fixed, and the value of the objective function $\mathcal{O}$ is non-increasing in each update. Moreover, since the value of the function $\mathcal{O}$ is bounded by 0, the convergence of SMEC is thus guaranteed.

4.4 Complete Algorithm

Finally, algorithm SMEC is summarized in Algorithm 1. The algorithm mainly consists of two parts: factorizing the adjacency matrices of the sub-networks jointly to obtain a factorization matrix that denotes membership between item categories and scenes, and selecting the candidate categories to generate the final scenes. Given an E-commerce HIN G and the number of scenes r, SMEC first initializes the representation matrices $\{\boldsymbol{H}_p\}_{p=1}^{|\mathcal{T}|}$ randomly and sets each scene in $\{S_j\}_{j=1}^{r}$ to be an empty set. Then SMEC iteratively updates $\{\boldsymbol{H}_p\}_{p=1}^{|\mathcal{T}|}$ based on the update rules (3), (4), and (5) (lines 5–9). After the above updating, the loss is computed and stored to judge convergence. When the difference between the successive values of the loss is smaller than a predetermined value θ (set to 10^{-5} in this work), the process can be regarded as reaching convergence. The loss value can be calculated every few iterations instead of every iteration. In fact, we can set the maximum number of iterations to achieve the trade-off between effectiveness and efficiency. Since each sub-network in G is generally sparse, the time complexity of updating $T_p \in \mathcal{T}_{\mathcal{B}}$ in each iteration is $O(d_{pp}n_pr+\sum_{q=1}^{|\mathcal{T}_{\mathcal{A}}|} d_{pq}n_qr + \sum_{q=1}^{|\mathcal{T}_{\mathcal{A}}|}(n_p+n_q)r^2)$, where n_p and n_q are the numbers of objects in T_p and T_q, respectively, and d_{pp} and d_{pq} are the average numbers of nonzero entries in each row of $\boldsymbol{W}_{pp}$ and $\boldsymbol{W}_{pq}$, respectively. Similarly, the computational cost of updating $T_p \in \mathcal{T}_{\mathcal{A}}$ in each iteration is $O(d_{pp}n_pr + \sum_{q=1}^{|\mathcal{T}_{\mathcal{B}}|} d_{pq}n_qr + \sum_{q=1}^{|\mathcal{T}_{\mathcal{B}}|} \times (n_p+n_q)r^2)$. Finally, in lines 14–19, SMEC constructs scenes from the category representation matrix $\boldsymbol{H}_c$ with a threshold ϵ.

Algorithm 1. SMEC

Input: $G, \mathcal{T}_{\mathcal{A}}, \mathcal{T}_{\mathcal{B}}, r, \epsilon, \theta$
Output: the set of scenes $\hat{\mathcal{S}}$

```
1: Initialize representation matrices {H_p}_{p=1}^{|T|};
2: Let each set S_j = ∅, S_j ∈ {S_j}_{j=1}^r, Ŝ = ∅;
3: num_iter = 0; last_loss = +∞;
4: while num_iter < max_iter do
5:     foreach T_p ∈ T_B do
6:         Update H_p by rule (3);
7:     foreach T_p ∈ T_A\T_c do
8:         Update H_p by rule (4);
9:     Update H_c by rule (5);
10:    Compute loss by (1) and (2);
11:    if |last_loss − loss| < θ then
12:        break;
13:    last_loss=loss; num_iter++;
14: for j = 1 to r do
15:     for i = 1 to n_c do
16:         if (H_c)_{ij} > ε then
17:             S_j ← S_j ∪ {x_i};
18:     if S_j ≠ ∅ then
19:         Ŝ ← Ŝ ∪ S_j;
20: return Ŝ.
```

The total time complexity of SMEC is $O(h(dknr+knr^2)+n_cr)$, where d is the average number of nonzero entries for each row in all weights matrices, n is the number of objects in the HIN G, $k=\max\{|\mathcal{T}_{\mathcal{A}}|, |\mathcal{T}_{\mathcal{B}}|\}$, and h is the number of iterations.

Remarks. 1) Interpretability. SMEC utilities the sets of commodity categories to represent scenes, so that the obtained scenes can be displayed intuitively and easy to be corresponded with the real-life situations. That is, SMEC obtains scenes that are practically interpretable. 2) Generality. SMEC is based on more abstract data types for handling the variety of HINs, by treating elements as different types. Networks with different element types are divided based on the types, and brought to the corresponding update rules for scene mining. Hence, SMEC can potentially be generalized to HINs with varying complexities. 3) Multiplicity. SMEC allows the overlapping of the commodity categories among different scenes so that the multiplicity is well preserved. 4) Learnability. SMEC formulates the scene mining problem for E-commerce in an unsupervised manner, and makes use of a matrix factorization based method. Hence, SMEC avoids the issue that the supervision scene information is generally missing.

5 Experimental Study

In this section, we evaluate SMEC using six real-world E-commerce datasets. We first introduce the datasets and evaluation metrics. Then we describe baseline methods and experimental settings. Finally, we present the experimental results and case studies to verify the effectiveness of the proposed method. We have also released the codes and datasets①.

5.1 Datasets

To the best of our knowledge, there are no public datasets that are available for scene mining tasks. 1) To evaluate our method quantitatively, we build four datasets from JD.com, one of the largest B2C E-commerce platform in China, namely Baby & Toy, Electronics, Fashion, and Food & Drink, where HINs are constructed from online logs and commodity information, and scene ground truth is generated manually by experts. User behaviors could vary considerably on different commodities. For example, users are usually more serious if they would like to purchase electronic products by checking item specifications than fast-moving consumer goods such as foods and drinks, which directly affects the HIN construction. Therefore, we build datasets on the four distinct commodity domains. 2) Amazon-product② is widely used for applications in E-commerce[52], but is lack of scene ground truths. Therefore, we choose its domains Home & Kitchen and Sports & Outdoors as another two datasets for case studies only, by extracting items that include diverse types and relations. Statistics of the six datasets (four from JD and two from Amazon) are shown in Table 2 and Table 3 analyzes the number of scenes on four JD datasets, and we next dis-

①https://github.com/e09b47e1/Scene_Mining_Data, Oct. 2021.
②http://jmcauley.ucsd.edu/data/amazon/links.html, Oct. 2021.

Table 2. Statistics of JD and Amazon Datasets

Dataset	Item-Brand	Item-Category	Item-Item (Co-Viewed)	Item-Item (Co-Bought)	Category-Category
Baby & Toy (JD)	38 749-4 306 (38 749)	38 749-103 (38 749)	38 749-38 749 (2 392 364)	38 749-38 749 (722 598)	103-103 (1 791)
Electronics (JD)	32 006-2 255 (32 006)	32 006-78 (32 006)	32 006-32 006 (2 125 348)	32 006-32 006 (162 571)	78-78 (825)
Fashion (JD)	39 316-4 985 (39 316)	39 316-91 (39 316)	39 316-39 316 (2 417 100)	39 316-39 316 (109 316)	91-91 (1 058)
Food & Drink (JD)	43 982-7 075 (43 982)	43 982-105 (43 982)	43 982-43 982 (2 870 748)	43 982-43 982 (794 791)	105-105 (1 628)
Home & Kitchen (Amazon)	66 011-7 795 (66 011)	66 011-989 (290 795)	66 011-66 011 (969 203)	66 011-66 011 (725 080)	-
Sports & Outdoors (Amazon)	72 626-8 263 (72 626)	72 626-1 659 (324 555)	72 626-72 626 (869 401)	72 626-72 626 (967 838)	-

Note: The statistics of each relation A-B has three parts: the number of A–the number of B (the number of A-B).

Table 3. Number of Scenes on JD Datasets

Dataset	Number of Scenes
Baby & Toy	323
Electronics	54
Fashion	438
Food & Drink	136

cuss more details on the former four datasets.

Users perform various behaviors in real-world E-commerce systems, such as "view", "purchase", "comment", and "save". In this work, we consider two types of user behaviors, "view" and "purchase", to construct item-item networks. A view session is a sequence of items viewed by a user within a period of time, and two items that are frequently co-viewed should be relevant. In the item-item view network, two items are connected if they are co-viewed by a user within the same session where the weight is the sum of the co-occurrence frequency in two months. Similarly, items purchased together within a time window are supposed to be related to each other. We construct an item-item purchase network by connecting two items if they are bought by the same user during five days and the weight is summed in six months. All the time length and time window mentioned above are empirically set based on the trade-off between the dataset size and co-viewed/co-bought relevance between items. The edge weights in both view and purchase networks are normalized between 0 and 1 using the min-max normalization. To be specific, the edge weights of the relations item-item (co-viewed) and item-item (co-bought) are on the range $[0, 1]$.

In this article, we consider two types of commodity attributes: "category" and "brand". The item-attribute network depicts binary links between items and different attributes, that is, the values of the edges in the relations item-category and item-brand are 0 or 1. For each type of commodity attributes, we continue to construct an attribute-attribute network by linking attribute nodes relevant to each other. To do this, each pair of attribute nodes, e.g., different commodity brands, is weighted by the sum of the co-view frequency in six months, and is labeled as 0 or 1 from consensus decision-making by three data labeling engineers to indicate the relevance or not. We reserve binary links without any normalization in the relation category-category.

The scene ground truths are manually generated by human experts. Specifically, this procedure consists of two steps: candidate generation and scene determination.

1) *Candidate Generation*. An E-commerce expert team (about 10 operation staff members) edits the original scene candidates based on the corresponding domain knowledge. They first remove the candidate sets that have fewer members than a preset value and merge the sets that have similar members. Then, the team checks each reserved candidate scene manually to adjust their elements or to eliminate the sets with poor diversity, and further divides them into two quality levels.

2) *Scene Determination*. A data labeling team with three engineers refines the generated scenes based on the criteria on whether a scene reasonably reflects a real-life situation. For the candidates labeled high quality, the team double-checks the rationality of the scenes' members and most of them would be put into the final scene sets. For the candidates labeled low quality, the experts first check them independently and do consensus decision-making to determine whether they are retained or not. The scenes considered reasonable by all experts would be reserved, while the others without consensus would be eliminated after full discussion.

To sum up, we build an E-commerce HIN, which has different types of nodes (i.e., item, category, and brand) and two relations (i.e., view and purchase) connecting item-item sub-networks from multiple dimensions, and has the same structure as shown in Subsection 4.1.

5.2 Evaluation Metrics

Given a set of ground truth scenes $\overline{S}$ and a set of predicted scenes $\hat{S}$, we use the average $F1$ score, Omega index, and normalized mutual information as metrics for evaluation. In general, the average $F1$ score and normalized mutual information are calculated to measure how well predictions match the ground truth, while the Omega index measures the quality of each commodity category assignment itself.

1) Average $F1$ score[53] computes the average $F1$ score between predicted scenes and ground truth scenes. For each predicted scene, we find its best matching ground truth scene with the largest $F1$ score. For each ground truth scene, we also search for its best matching predicted scene. The average $F1$ score is defined as:

$$\frac{1}{2}\left(\frac{1}{|\hat{S}|}\sum_{\hat{S}_i\in\hat{S}} F1(\overline{S}_{\overline{g}(i)},\hat{S}_i)+\frac{1}{|\overline{S}|}\sum_{\overline{S}_i\in\overline{S}} F1(\overline{S}_i,\hat{S}_{\hat{g}(i)})\right),$$

where $F1(\cdot,\cdot)$ is the harmonic mean of precision and recall values, and the matching functions $\overline{g}(i)$ and $\hat{g}(i)$ are defined as $\overline{g}(i)=\arg\max_j F1(\hat{S}_i,\overline{S}_j)$ and $\hat{g}(i)=\arg\max_j F1(\hat{S}_j,\overline{S}_i)$, respectively.

2) Normalized mutual information (NMI) adopts the criterion used in information theory (i.e., normalization of the mutual information) to compare the mined and ground truth scenes. NMI has been proposed as an effective metric for overlapping clustering and community detection (more details can be found in [54]).

3) Omega index[55] computes the accuracy of the same number of scenes that each commodity category shares as follows:

$$\frac{1}{|V_c|^2}\sum_{u,v\in V_c,u\neq v}\mathbf{1}\{|\overline{S}_{uv}|=|\hat{S}_{uv}|\},$$

where $\overline{S}_{uv}$ is the set of ground truth scenes that two categories u and v share, $\hat{S}_{uv}$ is the shared set of prediction scenes, and V_c is the set of commodity categories. Note that the case $|\overline{S}_{uv}|=|\hat{S}_{uv}|=0$ is not considered as a successful prediction.

All the above evaluation metrics are in the range between 0 and 1 such that higher values mean better performance, and 1 indicates perfect matching between the predicted and ground truth scenes.

5.3 Comparison Methods

SMEC learns a sparse vector representation for each commodity category in an E-commerce HIN, therefore we compare our method with a variety of network representation learning algorithms. SMEC is also compared with overlapping clustering and community detection methods because they can be naturally utilized to generate scene clusters. Table 4 shows the utilized sub-networks of the compared methods. These methods are described as follows.

1) *Fuzzy C-Means* (*FCM*)[18]. It is a soft version of k-means, and allows overlapping clusters, where each object can be assigned to one or more clusters with different degrees. We apply FCM to the adjacency matrix associated with the category-category sub-network to generate scenes, i.e., it only utilizes the relations between commodity categories.

2) *Non-Negative Matrix Factorization* (*NMF*)[19].

Table 4. Statistics of Sub-Networks Utilized by Compared Methods

Method	Category-Category	Item-Category	Item-Brand	Item-Item (Co-Viewed)	Item-Item (Co-Bought)
DeepWalk[41]	✓				
node2vec[42]	✓				
metapath2vec[44]	✓	✓	✓	✓	✓
GraphSAGE[46]	✓				
BigClam[22]	✓				
NMF[19]	✓				
FCM[18]	✓				
HMFClus-S[29]	✓	✓	✓	✓	✓
SMEC	✓	✓	✓	✓	✓
SMEC-non	✓	✓	✓	✓	✓
SMEC-FCM	✓	✓	✓	✓	✓
SMEC-II		✓		✓	✓
SMEC-AA	✓				
SMEC-IA		✓			

Note: The check mark indicates that the method utilizes the corresponding sub-network.

It is an effective matrix factorization method based on the part-based representation. We apply NMF to the adjacency matrix of the category-category sub-network, and use the factorization matrix as the low-dimensional representation of categories.

3) *DeepWalk*[41]. It adopts the truncated random walk and skip-gram model to learn low dimensional feature representations of network nodes. We perform DeepWalk on the category-category sub-network to learn a vector representation for each category, where the default setting in [41] is used.

4) *node2vec*[42]. It is also a random walk based embedding method that combines the BFS (Breath First Search) and DFS (Depth First Search) to control the exploration of first-order and second-order neighbors. We run node2vec on the category-category sub-network to learn node representation, use its default setting, and perform a grid search for hyperparameters $p, q \in \{0.25,\ 0.50, 1, 2, 4\}$.

5) *metapath2vec*[44]. It leverages the random walk guided by the predefined meta-path scheme for network representation learning in an HIN. To mitigate the bias induced by a single meta-path scheme, we design multiple category to category meta-paths, i.e., CC (category-category), CIBIC (category-item-brand-item-category), CIIC (category-item-item-category, item in the user view dimension) and CIIC (category-item-item-category, item in the user purchase dimension). We run metapath2vec on each meta-path scheme and concatenate each embedding vector to generate the final representation.

6) *GraphSAGE*[46]. It is a state-of-the-art graph neural network method, where the representation of a node can be updated by aggregating the representation of its neighbors in an inductive manner. We use the unsupervised version of GraphSAGE on the category-category sub-network, where the neighbor set is generated via random walks with the default settings in [46].

7) *BigClam*[22]. It is an overlapping community detection method on a homogeneous network. This method follows the observation that there exist dense connections between overlapping clusters, i.e., the probability of existing an edge between two nodes increases with the number of their shared communities. We apply BigClam to the category-category sub-network.

8) *HMFClus-S*[29]. It is a state-of-the-art overlapping clustering algorithm for HINs. HMFClus-S simultaneously generates clusters for all types of objects based on the similarity of the same type of nodes. It applies non-negative matrix tri-factorization on the similarity matrices derived from meta-paths to learn a consensus latent representation. We use the same meta-paths as metapath2vec to extract similarity matrices from the input HIN.

9) *SMEC-non*. It is a variant of SMEC by removing the $\ell_{2,1}$-norm regularization. As a result, SMEC-non learns a dense vector representation from the HIN.

10) *SMEC-FCM*. It is a variant of SMEC such that we apply FCM to generate overlapping clusters instead of setting a threshold.

11) *SMEC-II*. It is a variant of SMEC to evaluate the influence of user behaviors based on the item-item sub-network. Besides the item-item sub-network, the item-category network is used as an intermediary to help the method generate scenes consisting of categories.

12) *SMEC-AA*. It is a variant of SMEC to observe the contribution of item attributes based on the attribute-attribute sub-network. SMEC-AA only uses the attribute-attribute sub-network as input.

13) *SMEC-IA*. It is a variant of SMEC and it only uses the item-attribute sub-network to mine scenes.

For the comparison methods that learn dense vector representation, we further apply Fuzzy C-Means to generate the overlapping scene clusters. For fair comparison, the representation dimension, which indicates the number of scene clusters, is 64, and the threshold ϵ is 0.1 in SMEC. More discussion about parameter selection is in Subsection 5.4.2. The hyperparameters α, β are chosen by a grid search over $\alpha, \beta \in \{0.1, 0.5, 1, 5, 10\}$.

5.4 Experimental Results

We evaluate the effectiveness of SMEC from three perspectives: 1) whether SMEC achieves better overlapping clustering performance than the baselines, 2) whether key components and parameters are essential to the performance of SMEC, and 3) whether SMEC generates reasonable scenes matching real-life situations.

5.4.1 Performance Comparison with Baselines

We summarize the comparative results in Table 5 and Table 6. From the results, we have the following observations.

Table 5. Scene Mining Experimental Results on Two JD Datasets Baby & Toy and Electronics

Method	Baby & Toy			Electronics		
	Average $F1$	NMI	Omega Index	Average $F1$	NMI	Omega Index
DeepWalk	0.511	0.218	0.001	0.471	0.230	0.002
node2vec	0.536	0.248	0.005	0.536	0.309	0.006
metapath2vec	0.549	0.276	0.009	0.538	0.313	0.006
GraphSAGE	0.502	0.204	0.003	0.510	0.269	0.004
BigClam	0.516	0.244	0.020	0.473	0.179	0.025
NMF	0.544	0.289	0.001	0.486	0.274	0.002
FCM	0.539	0.272	0.004	0.474	0.247	0.002
HMFClus-S	0.553	0.288	0.003	0.525	0.285	0.006
SMEC	**0.590**	**0.363**	**0.036**	**0.579**	**0.414**	**0.111**
SMEC-II	0.579	0.328	0.002	0.557	0.387	0.005
SMEC-AA	0.291	0.168	0.035	0.347	0.172	0.091
SMEC-IA	0.482	0.236	0.000	0.474	0.293	0.001

Table 6. Scene Mining Experimental Results on Two JD Datasets Fashion and Food & Drink

Method	Fashion			Food & Drink		
	Average $F1$	NMI	Omega Index	Average $F1$	NMI	Omega Index
DeepWalk	0.545	0.310	0.001	0.482	0.153	0.004
node2vec	0.562	0.319	0.003	0.565	0.296	0.008
metapath2vec	0.570	0.332	0.012	0.535	0.244	0.002
GraphSAGE	0.549	0.284	0.004	0.482	0.180	0.010
BigClam	0.543	0.266	0.020	0.475	0.194	0.026
NMF	0.556	0.291	0.010	0.509	0.214	0.012
FCM	0.547	0.335	0.001	0.516	0.226	0.007
HMFClus-S	0.588	0.345	0.006	0.548	0.278	0.011
SMEC	**0.605**	**0.394**	**0.066**	**0.586**	**0.348**	**0.053**
SMEC-II	0.590	0.365	0.003	0.576	0.319	0.006
SMEC-AA	0.322	0.181	0.044	0.313	0.154	0.034
SMEC-IA	0.500	0.299	0.000	0.447	0.120	0.000

1) The methods applied to the category-category sub-network are only outperformed by metapath2vec in three out of the four datasets on average $F1$, which indicates that it is important to capture information from different object types. In contrast, the methods considering interactions between categories only ignore the information from different user behaviors and commodity attributes.

2) On the Food & Drink dataset, metapath2vec does not have better results on average $F1$ compared with the baselines based on the homogeneous network. It is possibly because metapath2vec relies on the design of meta-path schemes, and meta-paths used in this work may not be informative for Food & Drink. For example, users are usually less sensitive to Food & Drink brands compared with electronic products and fashion clothes; thus meta-path "CIBIC" may bring less information.

3) HMFClus-S outperforms metapath2vec in three out of the four datasets on average $F1$ and NMI. There are two main reasons. First, although these two methods adopt the same set of meta-paths, HMFClus-S explicitly extracts the similarity between the same type of nodes, which could be more precise than the random sampling from random walks. Second, the non-negative matrix tri-factorization helps to improve the representation of each node. From Table 5 and Table 6, we see that SMEC outperforms HMFClus-S and metapath2vec on all four datasets. This means that SMEC is able to extract more comprehensive information that may be ignored by the bias induced from the design of meta-paths.

4) On metric NMI, although there are no obvious dominant methods in all baselines, we find that the results of metapath2vec and HMFClus-S are consistently better than the median of the baselines which only utilize the information from the category-category sub-network. Compared with the single information source, the complex information from the interrelated networks is easier to help the methods perform better.

5) When considering Omega index, it is observed that BigClam outperforms all the other baselines on all four datasets. BigClam focuses on capturing the

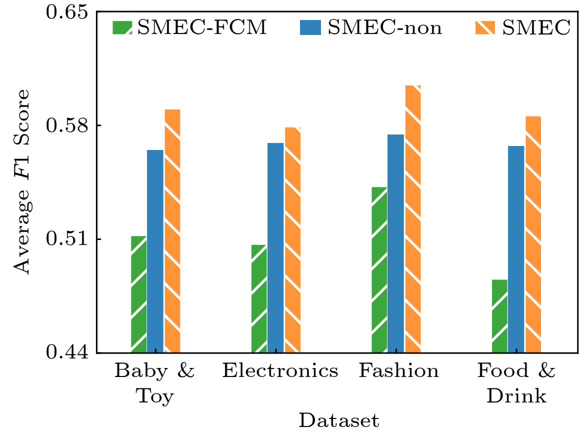
sub-network where there exist dense connections between different nodes, and BigClam tends to assign those nodes with dense connections into multiple clusters. As a result, BigClam achieves higher Omega index values because commodity categories frequently related to other categories are easier to be assigned into multiple scene clusters. For example, “mobile phone” may appear in multiple scenes as phones are related to many other categories of phone accessories or digital devices.

6) The proposed method SMEC obtains best overall performance using different evaluation metrics. Specifically, SMEC boosts 6.7%, 7.6%, 2.9%, and 3.7% for the average $F1$ scores, 25.6%, 32.3%, 14.2%, and 17.6% for NMI, and 80.0%, 344.0%, 230.0%, and 103.8% for Omega index on datasets Baby & Toy, Electronics, Fashion, and Food & Drink, compared with the best baseline. There are several main reasons. First, SMEC leverages comprehensive information of diverse commodity attributes and user behaviors in the elaborate E-commerce HIN. In contrast, the homogeneous category-category sub-network cannot provide rich multi-aspect information. Second, SMEC utilizes matrix factorization to integrate the global information from the input HIN, which is essentially different from meta-path based HIN embedding methods. Moreover, the design of meta-path highly relies on the domain knowledge and may not be informative to specific datasets. Third, the sparse representation enables SMEC to directly encode the scene information from the input HIN, while baselines rely on the dense vector representation with additional noises, and thus the specific commodity category could be incorrectly clustered into irrelevant scene clusters.

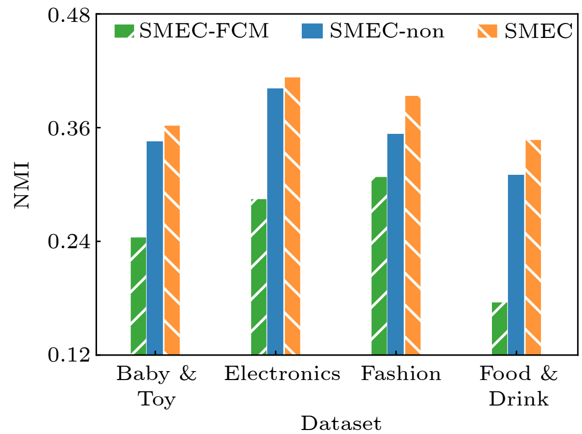
5.4.2 *Component Analyses*

We study how different components or key parameters affect SMEC's performance.

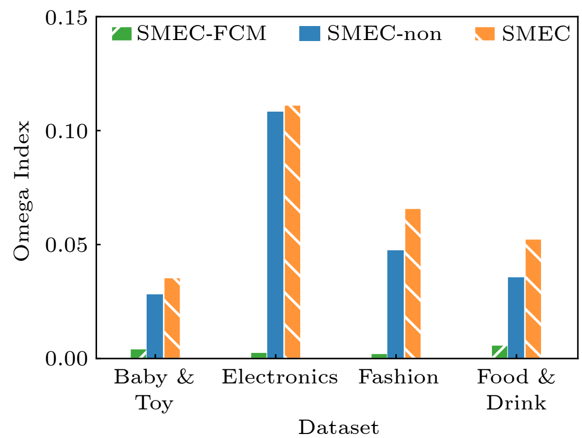
Fig.3 shows the performance comparisons between SMEC and its two variants. It is observed that SMEC obtains better results than SMEC-non and SMEC-FCM in terms of all three evaluation metrics. SMEC performs better than SMEC-non, and this confirms that the sparse representation helps the method to capture scene-based co-relations between the commodity categories. In contrast, the dense representation derived from SMEC-non brings additional noises that may further affect the overlapping performance.



(a)



(b)



(c)

Fig.3. Performance of SMEC against its variants on four JD datasets. (a) Average $F1$ score. (b) Normalized mutual information. (c) Omega index.

SMEC performs better than SMEC-FCM, which suggests that setting a threshold is more compatible to encode the scene information from the sparse representation.

We conduct sub-network analyses and report the results in Table 5 and Table 6. The effectiveness of the sub-networks item-item, attribute-attribute, and item-attribute is evaluated by the variants of SMEC: SMEC-II, SMEC-AA, and SMEC-IA. From Table 5 and Table 6, we have following observations. 1) SMEC-IA only utilizes item-attribute sub-networks, and its performance is not better than the best baseline on all metrics. 2) Compared with SMEC-IA, SMEC-II increases the use of the item-item sub-network and gets better performance than other baselines on average $F1$ and NMI. This means that the user behaviors benefit the proposed method on the two metrics. 3) Compared with SMEC-II, SMEC increases the use of the attribute-attribute sub-network. SMEC outperforms SMEC-II on all metrics and SMEC-AA achieves the best results compared with other baselines on Omega index, which shows that the item attributes promote the scene mining task. 4) To incorporate user behaviors and item attributes into a model, the item-attribute sub-network is essential for providing connection information between them. 5) Each sub-network of the HIN makes contributions from different aspects. The results of the variants also verify the design of SMEC: extracting information of user behaviors and item attributes from item-item and attribute-attribute sub-networks to mine scenes with the help of systematic combination based on the connection of item-attribute sub-networks.

Fig.4 reports the performance on the four datasets with respect to the number of scenes r. In particular, we set r to four different values $\{32, 64, 128, 256\}$ with the threshold ϵ set to 0.1. It is observed that SMEC keeps good performance on most values of r, which shows the robustness of SMEC. Although the values of all metrics do not drop sharply when r is 32, SMEC does not perform so well as the other cases. The main reason is that SMEC assigns too many irrelevant categories into the same clusters and cannot generate enough valid scenes when r is too small.

We continue to conduct parameter analyses on the threshold ϵ and report the results in Fig.5. The threshold ϵ is evaluated between 0 and 1 with every 0.02 interval. It reveals that the performance first increases and then decreases after the threshold ϵ is larger than 0.1, and this means that the threshold ϵ is essential to the performance. The method does not perform well when the threshold ϵ is too small because a given category could be incorrectly predicted into irrelevant scenes, and it also does not perform well when the threshold ϵ is too large, as a specific scene may include no commodity categories, i.e., an invalid empty scene.

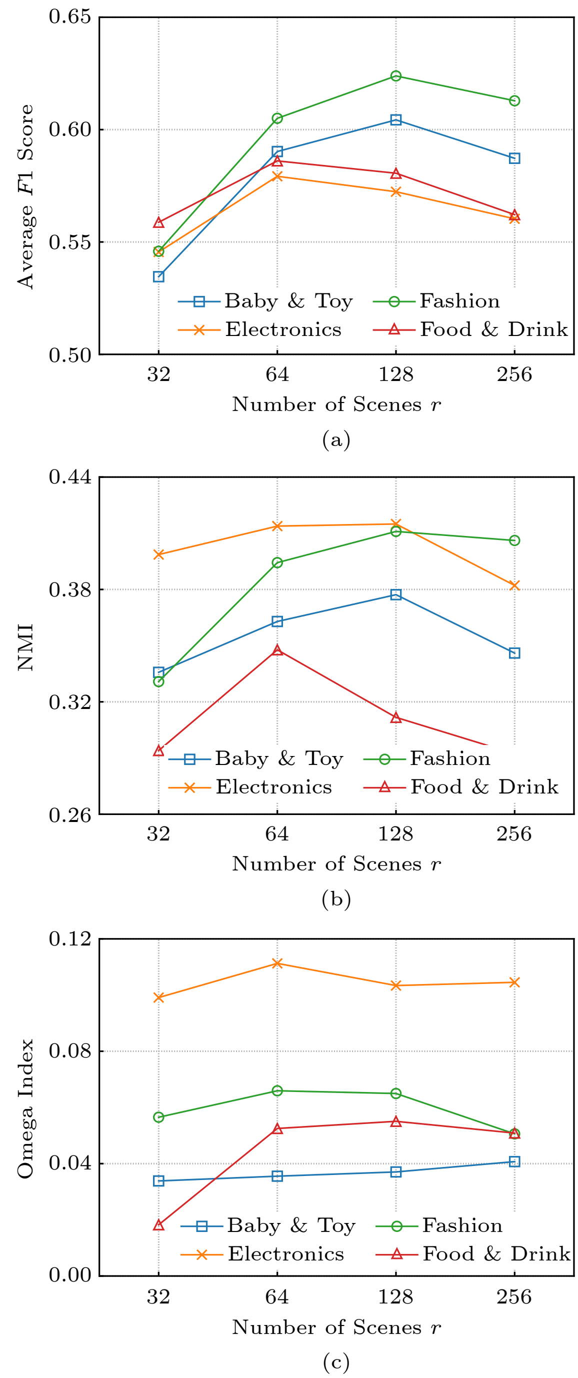
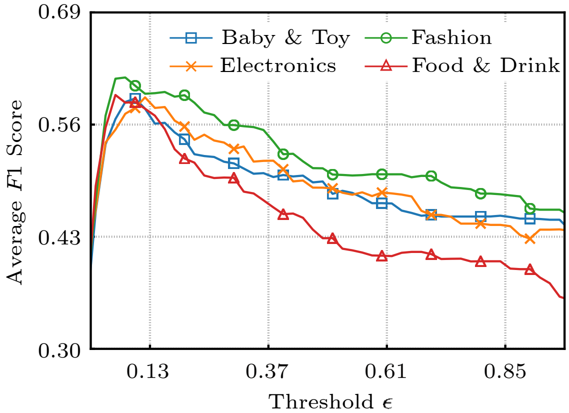


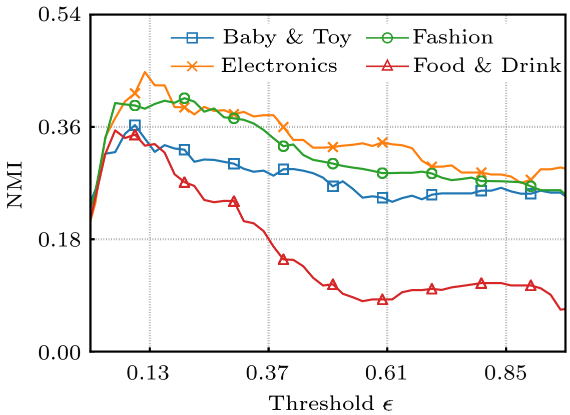
Fig.4. Parameter analyses on four JD datasets: the number of scenes r. (a) Average $F1$ score. (b) Normalized mutual information. (c) Omega index.

5.4.3 *Convergence Analyses*

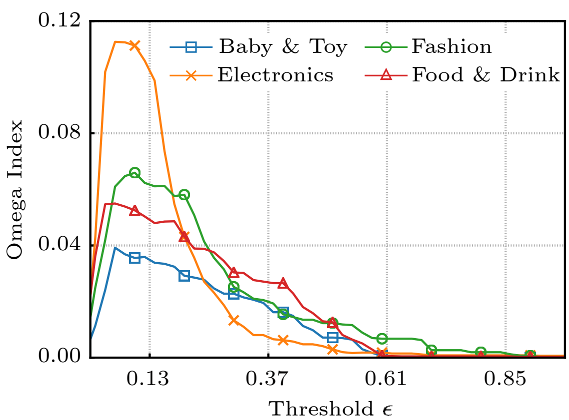
Fig.6 shows the convergence results of the objective function $\mathcal{O}$ in SMEC on all the four JD datasets. The iterations stop when the difference between the objective function values of two successive iterations is less than 10^{-5}. We see that the loss curves decrease monotonously, and it shows the convergence of



(a)



(b)



(c)

Fig.5. Parameter analyses on four JD datasets: the prediction threshold ϵ. (a) Average $F1$ score. (b) Normalized mutual information. (c) Omega index.

SMEC experimentally. Specifically, it is easy to find that SMEC converged before the maximum number of iterations (set to 500 here) is reached on all the four JD datasets.

5.4.4 Case Studies

We further present several real-world cases generated by SMEC on two Amazon datasets Home & Kitchen and Sports & Outdoors in Table 7.

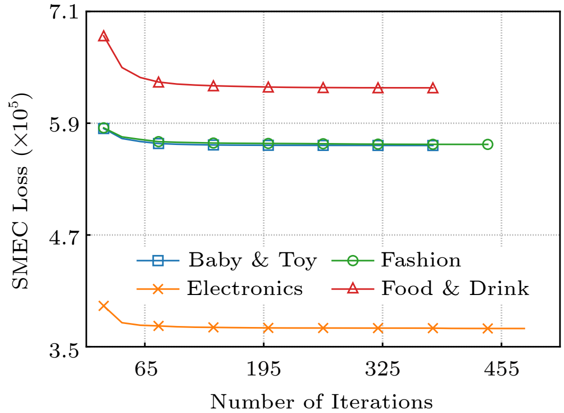


Fig.6. Convergence of SMEC on four JD datasets.

We have the following observations from these results. 1) The scenes discovered by SMEC in an unsupervised way are practically meaningful such that each scene indicates a real-life situation. For example, the scene "Baking" could be applied to the situation where the user has already bought a bakeware and thus complementary commodities in the corresponding categories, e.g., Cake Toppers and Baking Cups, are recommended. 2) A given commodity category could belong to multiple scenes. For example, the category "Storage & Organization" appears in both scene "Family Storage" and scene "Picnic", which confirms that multiplicity is the inherent property of scene mining that can be successfully achieved by an overlapping clustering problem. 3) SMEC helps to capture complementary relations between certain commodity categories in multiple scenes. For example, SMEC assigns the categories "Bakeware" and "Candy Making Supplies" into the scenes "Baking" and "Picnic". 4) Commodity categories that are not directly relevant may appear in the same scene. For example, there is a weak relevance between "Camp Stoves" and "Lanterns", but they co-occur in the scene "Camp". 5) SMEC can also obtain scenes from different HINs without any extra change of the model structure. As shown in Table 2, compared with the JD datasets, there is no category-category sub-network data on the Amazon datasets, but it does not hinder our method to discover scenes.

There are failure cases in the experiments. Since these mined scenes most include categories with a high similarity, the lack of diversity cannot support them to reflect real-life situations, for example, "Writing & Correction", "Supplies", "Pens & Refills", "Ballpoint Pens", and "Copy & Multipurpose Paper".

Table 7. Scenes Mined by SMEC on Amazon Datasets Home & Kitchen and Sports & Outdoors

Scene	Commodity Category
Baking	Bakeware, Candy Making Supplies, Decorating Tools, Cake Toppers, Baking Tools & Accessories, Cookie Cutters, Cake Pans, Specialty & Novelty Cake Pans, Sculpting & Modeling Tools, Icing Dispensers & Tips, Baking Cups
Family storage	Storage & Organization, Clothing & Closet Storage, Cabinet Organizers, Baskets & Bins
Home brewing	Wine Making, Beer Brewing, Kitchen Knives & Cutlery Accessories, Kegs & Kegging
Picnic	Bakeware, Candy Making Supplies, Storage & Organization, Lunch Boxes & Bags, Candy Making Molds
Camp	Outdoor Gear, Camping & Hiking, Knives & Tools, Personal Care, Camp Kitchen, Coolers, Camp Stoves, Lights & Lanterns, Lanterns, Safety & Survival, Stove Accessories, Camping Furniture, Tents, Backpacks & Bags
Fishing	Hunting & Fishing, Fishing, Terminal Tackle, Rod & Reel Combos, Spinning Combos, Lures, Baits & Attractants

There are a number of similar categories in some commodity domains. Compared with all the other categories, they are often very close to each other in the embedding vector space. It is difficult to take one of them separately for composing reasonable scenes with all the other categories. Mixing the different domains together to increase variety might solve this problem, since the user behaviors and commodity attributes from different domains might make the embedding vectors more dispersed. It is our future work.

5.4.5 Summary

From these tests, we find the followings.

1) Our method SMEC is effective for scene mining. SMEC's three metrics, average $F1$, NMI, Omega index, are improved on average 5.2%, 22.4%, and 189.5% compared with the best baseline on all datasets, respectively.

2) The strategy of constructing sparse representation by adding $\ell_{2,1}$-norm to the objective function encodes the scene information into the embedding vectors directly. Since no post-processing is needed, wrong decisions are avoided to achieve better results.

3) SMEC performs well on most values of the number of scenes, which indicates the robustness of SMEC. Moreover, the setting of the threshold ϵ is also flexible in a range of $[0.05, 0.25]$ for our method.

4) Besides the theoretical proof of convergence, we have experimentally shown the convergence of SMEC.

5) The scenes discovered by our method SMEC are practically meaningful where each scene corresponds to a specified real-life situation, and they also have multiple attributes of scenes at the same time.

6 Conclusions

In this work, we studied a novel scene mining problem for E-commerce systems, where a scene is defined as a set of commodity categories. We proposed a method SMEC to discover the practically meaningful scenes with multiple attributes by implementing non-negative matrix factorization on the E-commerce HIN, which effectively extracts information from multiple aspects. Moreover, we constructed sparse representation by directly encoding the scene information into the embedding vector to avoid wrong decisions caused by additional post-processing. The extensive experiments showed the effectiveness of our proposed method SMEC for scene mining and the robustness of SMEC to the HINs with different components in E-commerce platforms. Compared with the traditional method[18, 19, 22, 29, 41, 42, 44, 46], SMEC on average improves the three metrics, average $F1$ score, NMI, Omega index, by 5.2%, 22.4%, and 189.5% on four JD datasets, respectively. SMEC also discovers the scenes normally on JD and Amazon datasets, which have different sub-network structures.

Scenes in E-commerce have many potential applications. One example is that the scene can be regarded as a special kind of side information, which may assist relational learning and graph representation to build item profiling or user profiling in E-commerce platforms. Another possible future topic is to feed a scene into the personalized recommendation for E-commerce, which is an interesting problem on recommending items in a given scene based on a user's preference.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. In *Proc. the 10th International Conference on World Wide Web*, May 2001, pp.285–295. DOI: 10.1145/371920.372071.

[2] Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recom-

mendation. In *Proc. the 19th International Conference on World Wide Web*, Apr. 2010, pp.811–820. DOI: 10.1145/1772690.1772773.

[3] Zhou G R, Zhu X Q, Song C R, Fan Y, Zhu H, Ma X, Yan Y H, Jin J Q, Li H, Gai K. Deep interest network for click-through rate prediction. In *Proc. the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Jul. 2018, pp.1059–1068. DOI: 10.1145/3219819.3219823.

[4] Kiapour M H, Yamaguchi K, Berg A C, Berg T L. Hipster wars: Discovering elements of fashion styles. In *Proc. the 13th European Conference on Computer Vision*, Sept. 2014, pp.472–488. DOI: 10.1007/978-3-319-10590-1_31.

[5] Liu Z W, Luo P, Qiu S, Wang X G, Tang X O. DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp.1096–1104. DOI: 10.1109/CVPR.2016.124.

[6] Kang W C, Kim E, Leskovec J, Rosenberg C, McAuley J. Complete the look: Scene-based complementary product recommendation. In *Proc. the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp.10524–10533. DOI: 10.1109/CVPR.2019.01078.

[7] Sun Y Z, Han J W, Yan X F, Yu P S, Wu T Y. PathSim: Meta path-based top-K similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 2011, 4(11): 992–1003. DOI: 10.14778/3402707.3402736.

[8] Huang Z P, Zheng Y D, Cheng R, Sun Y Z, Mamoulis N, Li X. Meta structure: Computing relevance in large heterogeneous information networks. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp.1595–1604. DOI: 10.1145/2939672.2939815.

[9] Fang Y, Lin W Q, Zheng V W, Wu M, Chang K C C, Li X L. Semantic proximity search on graphs with metagraph-based learning. In *Proc. the 32nd IEEE International Conference on Data Engineering*, May 2016, pp.277–288. DOI: 10.1109/ICDE.2016.7498247.

[10] Zhao H, Yao Q M, Li J D, Song Y Q, Lee D L. Metagraph based recommendation fusion over heterogeneous information networks. In *Proc. the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2017, pp.635–644. DOI: 10.1145/3097983.3098063.

[11] Wang X, Ji H Y, Shi C, Wang B, Ye Y F, Cui P, Yu P S. Heterogeneous graph attention network. In *Proc. the 28th World Wide Web Conference*, May 2019, pp.2022–2032. DOI: 10.1145/3308558.3313562.

[12] Liu J L, Wang C, Gao J, Han J W. Multi-view clustering via joint nonnegative matrix factorization. In *Proc. the 13th SIAM International Conference on Data Mining*, Dec. 2013, pp.252–260. DOI: 10.1137/1.9781611972832.28.

[13] Qiu J Z, Dong Y X, Ma H, Li J, Wang K S, Tang J. Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and node2vec. In *Proc. the 11th ACM International Conference on Web Search and Data Mining*, Feb. 2018, pp.459–467. DOI: 10.1145/3159652.3159706.

[14] Linden G, Smith B, York J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 2003, 7(1): 76–80. DOI: 10.1109/MIC.2003.1167344.

[15] Zhang H, Chen X, Ma S. Dynamic news recommendation with hierarchical attention network. In *Proc. the 19th IEEE International Conference on Data Mining*, Nov. 2019, pp.1456–1461. DOI: 10.1109/ICDM.2019.00190.

[16] Wang G, Guo Z Y, Li X, Yin D W, Ma S. SceneRec: Scene-based graph neural networks for recommender systems. In *Proc. the 24th International Conference on Extending Database Technology*, Mar. 2021, pp.397–402. DOI: 10.5441/002/EDBT.2021.41.

[17] Xie J R, Kelley S, Szymanski B K. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys*, 2013, 45(4): Article No. 43. DOI: 10.1145/2501654.2501657.

[18] Dunn J C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 1973, 3(3): 32–57. DOI: 10.1080/01969727308546046.

[19] Lee D D, Seung H S. Algorithms for non-negative matrix factorization. In *Proc. the 13th International Conference on Neural Information Processing Systems*, Jan. 2000, pp.556–562. DOI: 10.5555/3008751.3008829.

[20] Duan L, Ma S, Aggarwal C, Ma T J, Huai J P. An ensemble approach to link prediction. *IEEE Trans. Knowledge and Data Engineering*, 2017, 29(11): 2402–2416. DOI: 10.1109/TKDE.2017.2730207.

[21] Wang F, Li T, Wang X, Zhu S H, Ding C. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 2011, 22(3): 493–521. DOI: 10.1007/S10618-010-0181-Y.

[22] Yang J, Leskovec J. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *Proc. the 6th ACM International Conference on Web Search and Data Mining*, Feb. 2013, pp.587–596. DOI: 10.1145/2433396.2433471.

[23] Wang X, Jin D, Cao X C, Yang L, Zhang W X. Semantic community identification in large attribute networks. In *Proc. the 30th AAAI Conference on Artificial Intelligence*, Feb. 2016, pp.265–271. DOI: 10.5555/3015812.3015851.

[24] Kuang D, Park H, Ding C H Q. Symmetric nonnegative matrix factorization for graph clustering. In *Proc. the 12th SIAM International Conference on Data Mining*, Apr. 2012, pp.106–117. DOI: 10.1137/1.9781611972825.10.

[25] Yang J, McAuley J, Leskovec J. Community detection in networks with node attributes. In *Proc. the 13th IEEE International Conference on Data Mining*, Dec. 2013, pp.1151–1156. DOI: 10.1109/ICDM.2013.167.

[26] Liu Y, Niculescu-Mizil A, Gryc W. Topic-link LDA: Joint models of topic and author community. In *Proc. the 26th Annual International Conference on Machine Learning*, Jun. 2009, pp.665–672. DOI: 10.1145/1553374.1553460.

[27] McAuley J, Leskovec J. Learning to discover social circles in ego networks. In *Proc. the 25th International Conference on Neural Information Processing Systems*, Dec. 2012, pp.539–547. DOI: 10.5555/2999134.2999195.

[28] Coscia M, Rossetti G, Giannotti F, Pedreschi D. DE-

MON: A local-first discovery method for overlapping communities. In *Proc. the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2012, pp.615–623. DOI: 10.1145/2339530.2339630.

[29] Zhang X C, Li H X, Liang W X, Luo J B. Multi-type co-clustering of general heterogeneous information networks via nonnegative matrix tri-factorization. In *Proc. the 16th IEEE International Conference on Data Mining*, Dec. 2016, pp.1353–1358. DOI: 10.1109/ICDM.2016.0185.

[30] Zhou Y, Liu L, Buttler D. Integrating vertex-centric clustering with edge-centric clustering for meta path graph analysis. In *Proc. the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2015, pp.1563–1572. DOI: 10.1145/2783258.2783328.

[31] Sun Y Z, Norick B, Han J W, Yan X F, Yu P S, Yu X. Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Trans. Knowledge Discovery from Data*, 2013, 7(3): Article No. 11. DOI: 10.1145/2500492.

[32] Shi Y, He X W, Zhang N J, Yang C, Han J W. User-guided clustering in heterogeneous information networks via motif-based comprehensive transcription. In *Proc. the 2019 European Conference on Machine Learning and Knowledge Discovery in Databases*, Sept. 2019, pp.361–377. DOI: 10.1007/978-3-030-46150-8_22.

[33] Elsner U. Graph partitioning: A survey. Technical Report, SFB 393/97-27, Technische Universität Chemnitz, Chemnitz, Germany, 1997. https://core.ac.uk/download/pdf/153227811.pdf, Jan. 2024.

[34] Tumasjan A, Sprenger T O, Sandner P G, Welpe I M. Predicting elections with Twitter: What 140 characters reveal about political sentiment. In *Proc. the 4th International Conference on Weblogs and Social Media*, May 2010, pp.178–152. DOI: 10.1609/icwsm.v4i1.14009.

[35] Azarbonyad H, Dehghani M, Beelen K, Arkut A, Marx M, Kamps J. Words are malleable: Computing semantic shifts in political and media discourse. In *Proc. the 2017 ACM on Conference on Information and Knowledge Management*, Nov. 2017, pp.1509–1518. DOI: 10.1145/3132847.3132878.

[36] Cui P, Wang X, Pei J, Zhu W W. A survey on network embedding. *IEEE Trans. Knowledge and Data Engineering*, 2019, 31(5): 833–852. DOI: 10.1109/TKDE.2018.2849727.

[37] Lu Y F, Shi C, Hu L M, Liu Z Y. Relation structure-aware heterogeneous information network embedding. In *Proc. the 33rd AAAI Conference on Artificial Intelligence*, Jul. 2019, pp.4456–4463. DOI: 10.1609/AAAI.V33I01.33014456.

[38] Tang J, Qu M, Wang M Z, Zhang M, Yan J, Mei Q Z. LINE: Large-scale information network embedding. In *Proc. the 24th International Conference on World Wide Web*, May 2015, pp.1067–1077. DOI: 10.1145/2736277.2741093.

[39] Wang D, Cui P, Zhu W. Structural deep network embedding. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp.1225–1234. DOI: 10.1145/2939672.2939753.

[40] Hu R J, Aggarwal C C, Ma S, Huai J. An embedding approach to anomaly detection. In *Proc. the 32nd IEEE International Conference on Data Engineering*, May 2016, pp.385–396. DOI: 10.1109/ICDE.2016.7498256.

[41] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online learning of social representations. In *Proc. the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2014, pp.701–710. DOI: 10.1145/2623330.2623732.

[42] Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp.855–864. DOI: 10.1145/2939672.2939754.

[43] Fu T Y, Lee W C, Lei Z. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proc. the 2017 ACM on Conference on Information and Knowledge Management*, Nov. 2017, pp.1797–1806. DOI: 10.1145/3132847.3132953.

[44] Dong Y X, Chawla N V, Swami A. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proc. the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2017, pp.135–144. DOI: 10.1145/3097983.3098036.

[45] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In *Proc. the 15th International Conference on Learning Representations*, Apr. 2017.

[46] Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs. In *Proc. the 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp.1024–1034. DOI: 10.5555/3294771.3294869.

[47] Sun Y Z, Han J W. Mining Heterogeneous Information Networks: Principles and Methodologies. Springer, 2012: 1–159. DOI: 10.1007/978-3-031-01902-9.

[48] Sun Y Z, Han J W. Mining heterogeneous information networks: A structural analysis approach. *ACM SIGKDD Explorations Newsletter*, 2012, 14(2): 20–28. DOI: 10.1145/2481244.2481248.

[49] Xu W, Liu X, Gong Y H. Document clustering based on non-negative matrix factorization. In *Proc. the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2003, pp.267–273. DOI: 10.1145/860435.860485.

[50] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. *Computer*, 2009, 42(8): 30–37. DOI: 10.1109/MC.2009.263.

[51] Boyd S, Vandenberghe L. Convex Optimization. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511804441.

[52] McAuley J, Targett C, Shi Q F, Van Den Hengel A. Image-based recommendations on styles and substitutes. In *Proc. the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Aug. 2015, pp.43–52. DOI: 10.1145/2766462.2767755.

[53] Yang J, Leskovec J. Community-affiliation graph model for overlapping network community detection. In *Proc. the 12th IEEE International Conference on Data Mining*, Dec. 2012, pp.1170–1175. DOI: 10.1109/ICDM.2012.139.

[54] Fortunato S. Community detection in graphs. *Physics Re-*

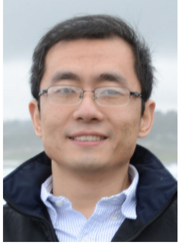
ports, 2010, 486(3/4/5): 75–174. DOI: 10.1016/J.PHYSREP.2009.11.002.

[55] Gregory S. Fuzzy overlapping communities in networks. arXiv: 1010.1523, 2010. https://arxiv.org/abs/1010.1523, Jan. 2024.

Gang Wang is now a Ph.D. candidate in computer science and technology at State Key Laboratory of Software Development Environment, the School of Computer Science and Engineering, Beihang University, Beijing. He received his B.S. degree in computer science and technology from Zhengzhou University, Zhengzhou, in 2016. His current research interests include recommender system and data mining.

Xiang Li is currently a young researcher with the title of Zijiang Young Scholar in the School of Data Science and Engineering at East China Normal University, Shanghai. Before that, he received his Ph.D. degree in computer science from The University of Hong Kong, Hong Kong, in 2018, M.S. degree in applied computer technology from University of Science and Technology of China, Hefei, in 2014, and B.S. degree in computer science from Nanchang University, Nanchang, in 2011. Previously, he worked as a research scientist in the Data Science Lab at JD.com from Aug 2018 to July 2019 and a postdoc researcher in The University of Hong Kong, Hong Kong, from July 2019 to Dec. 2020. His current research interests include data mining, graph mining, and machine learning.

Zi-Yi Guo is a research scientist at JD.com, Beijing. He obtained his Ph.D. degree in computer science from Lehigh University, Bethlehem, in 2017, and B.S. degree in software engineering from Northwestern Polytechnical University, Xi'an, in 2012. His research interests include recommender system and data mining.



Da-Wei Yin is an engineering director at Baidu, Inc., Beijing. He is managing the search science team at Baidu, Inc., Beijing. Previously, he was a senior director, managing the recommendation engineering team at JD.com from 2016 to 2020. Prior to JD.com, he was senior research manager at Yahoo Labs. He obtained his Ph.D. degree from Lehigh University, Bethlehem, in 2013, M.S. degree from Lehigh University, Bethlehem, in 2010, and B.S. degree from Shandong University, Jinan, in 2006, all in computer science. From 2007 to 2008, he was an M.Phil. student in The University of Hong Kong, Hong Kong. He is a recipients of WSDM 2016 Best Paper Award, KDD 2016 Best Paper Award, WSDM 2018 Best Student Paper Award, and ICHI 2019 Best Paper Honorable Mention. His research interests include data mining, applied machine learning, and information retrieval and recommender system.

Shuai Ma is a professor at State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing. He received his Ph.D. degrees in computer science from University of Edinburgh, Edinburgh, in 2010, and Peking University, Beijing, in 2004, respectively. He is a recipient of the Best Paper Award for VLDB 2010 and Best Paper Candidate for ICDM 2019. He has been an associate editor of VLDB Journal since 2017, IEEE Transactions on Big Data since 2020, and Knowledge and Information Systems since 2020. His research interests include database theory and systems, and big data.