applied sciences

Article

Incorporating a Topic Model into a Hypergraph Neural Network for Searching-Scenario Oriented Recommendations

Xin Huang * and Xiaojuan Liu

School of Government, Beijing Normal University, Beijing 100875, China; lxj_2007@bnu.edu.cn
* Correspondence: 11132020309@bnu.edu.cn

Abstract: The personalized recommendation system is a useful tool adopted by e-retailers to help consumers to find items in line with their preferences. Existing methods focus on learning user preferences from a user-item matrix or online reviews after purchasing, and they ignore the interactive features in the process of users' learning about product information through search queries before they make a purchase. To this end, this study develops a topic augmented hypergraph neural network framework to predict the user's purchase intention by connecting the latent topics embedded in a consumer's online queries to their click, purchase, and online review behavior, which aims at mining the connection information existing in the interaction graph domain. Meanwhile, in order to reduce the influence of text noise words by fusing topic information, we integrate the topic distribution and convolutional embedding to better represent each user and item, which can make up for the lack of topic information in traditional convolutional neural networks. Extensive empirical evaluations on real-world datasets demonstrate that the proposed framework improves the novelty of recommendation items as well as accuracy. From a managerial perspective, recommending diversified and novel items to consumers may increase the users' satisfaction, which is conducive to the sustainable development of e-commerce enterprises.

Keywords: personalized recommender system; online query sessions; user's preference modeling; topic model; hypergraph neural network

Citation: Huang, X.; Liu, X. Incorporating a Topic Model into a Hypergraph Neural Network for Searching-Scenario Oriented Recommendations. *Appl. Sci.* **2022**, *12*, 7387. https://doi.org/10.3390/app12157387

Academic Editor: Antonio Moreno

Received: 29 June 2022
Accepted: 21 July 2022
Published: 22 July 2022

1. Introduction

As mobile Internet and information technology has achieved great technological progress, consumers can browse products and make purchase through mobile devices from anywhere at any time [1]. The rapid development of mobile e-commerce has intensified competition between e-commerce companies. E-commerce enterprises maintain their competitive advantage by implementing product differentiation strategies, offering consumers more products and discounts, as well as exploiting intelligent information filtering systems to assist online users in quickly finding products that are in line with their preferences [2]. Online retailers need to provide customers with targeted goods and services according to their different needs to avoid homogeneous competition [3]. The recommender system is a classical type of information filtering system that attempts to recommend products to users that can conform to their different hobbies and personal experience [4,5]. However, the traditional recommendation method only uses ratings to reflect the user's overall preference for items, but it is difficult to depict users' relative preferences for multiple dimensions of product features [6]. E-commerce platforms hope to help consumers to quickly find the right product that satisfies the heterogeneous needs of consumers and delivers their business ideas and product information to these potential consumers in a targeted manner [7,8]. E-commerce enterprises are committed to developing a more powerful personalized recommender system to enhance shopping experience of online consumers.

Although traditional personalized recommendation methods (such as collaborative filtering and content-based recommendation algorithms) are widely used, they all have their own shortcomings. Therefore, hybrid recommender systems are proposed to deal with these shortcomings by combining different recommendation algorithms. Recently, the most widely used hybrid recommender systems are based on collaborative filtering algorithms and content-based algorithms, while other types of combinations have also been developed. The main idea of collaborative filtering is to use the preferences of user groups with similar tastes to the target user to predict what the target user might like. The data sparsity problem and the cold start problem are considered as two key problems faced by collaborative filtering techniques [9–12]. The data sparsity problem seriously restricts the performance of collaborative filtering. For large business websites, due to the large number of products and users, the user rating products generally do not exceed 1% of the total number of products. The cold start problem usually occurs when new users arrive. As there is no user behavior data when a new user enters the system, it is difficult to make effective recommendations. The basic idea of content-based filtering is to recommend other items similar to the items that the user liked in the past. The content-based filtering technology relies on user portraits. Therefore, even if the database does not contain user interests, it will not affect the accuracy of the recommendation results [13]. However, the content-based filtering technique depends on the item's metadata. That is to say, the system needs rich item content descriptions and complete user portraits. Hence, users can only get recommendations similar to items in their own profile, hardly getting diversified options. One of the ways to build a hybrid recommender system is to independently apply collaborative filtering, content-based and other algorithms, for recommendations by combining the recommendation results of two or more systems and using the linear combination of prediction scores to make recommendations. Some hybrid recommender systems are content-based collaborative filtering algorithms. That is, the similarity of users is calculated through content-based profiles, rather than the information of products that are rated together. This can overcome the sparsity problem in collaborative filtering systems. Another hybrid recommendation mechanism is utilizing multiple independent recommendation algorithms, each of which generates its own recommendation results, and fusing these recommendation results in the mixing stage to generate the final recommendation result. It can be seen from the above analysis that the above recommendation techniques predict consumer purchase intention based on product ratings of what other similar users have purchased or what they themselves have purchased. User preferences characterized by these methods are commonly presented based on user ratings of 1 to 5, which can capture a user's overall evaluation of the product. However, ratings data is too simple to capture consumers' multi-dimensional and fine-grained evaluation of product attributes. Unlike sparse consumer purchasing data, consumers conduct extensive online search queries before making a purchasing decision. Take the laptop as an example, customers formulate queries like "best laptop for programming" that directly reflect their content preferences for product features. Thus, it is important to understand the navigation keywords associated with product features in users' online query sessions. It is critical for e-commerce platforms to extract consumer content preferences from online search sessions [14].

Despite the importance of inferring user preferences from online query sessions, few studies have focused on this area. Roscoe et al. [15] revealed that online search queries focused on superficial product features rather than key knowledge. Information search behavior is an important factor that is assessed to identify differences in consumers regarding their purchasing patterns and preferences [16]. Kim et al. [17] verified that there is a significant relationship between new product diffusion and internet search volume. Internet search volume is an important indicator for predicting new product demand. Liu and Toubia [18] suggested that marketers should focus their efforts on keywords and queries that reflect content preferences that are well aligned with the content they are trying to promote. Codignola et al. [19] found that these browsing data can be saved with cookies and can be used to show customers potentially suitable items. Although numerous studies have been conducted to empirically verify that online queries can explicitly express consumers' content preference or can be used to predict product demand, quantitative studies that can estimate content preferences from online queries in an interpretable manner are lacking. Therefore, it is managerially important for sustainable e-retailers to develop intelligent recommendations based on learning dynamic customer preference from online query sessions [20].

In this study, we develop a topic augmented hypergraph neural network (Topic-HGNN) framework, which uses the hypergraph structure to capture the complex multivariate relationship among users, query topics, items, and item features. Besides, we incorporate topic models into the hypergraph neural network to more finely depict user preferences and product characteristics. To this end, we specifically propose an Aggregated Latent Dirichlet Allocation model to jointly extract users' content preference topics from queries and webpages, and apply the Latent Dirichlet Allocation [21] model to extract product feature topics from online reviews, which is useful to enhance feature interaction interpretability. In detail, the proposed Topic-HGNN framework involves: (1) adopting a hypergraph to model the multivariate relationship among users, query topics, items, and item features and applying the dual-embedding mechanism to handle complex and high-order correlations; (2) applying hyperedge corruption to generate a user-query hypergraph and an item-feature hypergraph and utilizing the hyperedge convolution layer to obtain user embedding and item embedding; (3) developing an Aggregated Latent Dirichlet Allocation model to jointly extract users' content preference topics from queries and webpages and applying the Latent Dirichlet Allocation model to extract product feature topics from online reviews; (4) integrating topic distribution and convolutional embedding to represent each user and item; and (5) using multilayer perceptron to calculate the soft match score between query entities and item entities.

We summarize the main contributions in the paper as follows:

- Despite the importance of inferring user preferences from online query sessions, very little research has focused on this area. In this paper, we propose a sustainable recommender system architecture based on inferring users' preferences from online query sessions, which can more accurately predict user purchase intentions;
- We develop an Aggregated Latent Dirichlet Allocation (ALDA) model, a novel topic model that can simultaneously learn user query topics and topics of corresponding clicked webpages. The ALDA model treats the joint topic distribution of queries and webpages as the topic distribution of user preferences. The data sparsity of online query data is avoided by aggregating corresponding webpages to assist in learning users' content preferences;
- To handle the complex multivariate relationship among users, query topics, items, and item features, we design a topic augmented hypergraph neural network (Topic-HGNN) framework to more accurately represent each user and item by integrating the convolution information and the topic information. The Topic-HGNN framework can significantly improve the accuracy and the novelty of recommended items;
- Extensive tests verify that our approach can better capture consumers' multi-dimensional preferences for product attributes and can better predict consumers' purchase intentions.

We organize the rest of this paper as follows. Section 2 summarizes the related works. Section 3 describes the proposed recommender system in details. Section 4 presents the extensive experiments designed to evaluate the effectiveness and the efficiency of the proposed framework. Section 5 summarizes the paper.

2. Related Work

From the above analysis, this paper aims to develop a novel personalized recommendation system based on learning dynamic customer preference from online query sessions. Thus, in this section, we briefly summarize related works from the following three aspects: traditional recommendation systems, online query sessions, and recommendations based on graph learning.

2.1. Traditional Recommendation Systems

The recommendation system is a widely used information filtering tool to provide customers with product information and suggestions to help users to decide which products they should purchase. Bobadilla et al. [22] classified recommendation methods into three categories: collaborative filtering, content-based filtering, and hybrid recommender systems.

Collaborative filtering is one of the earliest and the most successful techniques used in recommender systems. It generally uses the preference of user groups with similar tastes to the target user to predict the target user's preference for a specific product. Generally speaking, there are two types of collaborative filtering techniques. The first is user-based collaborative filtering [23,24], and the second is item-based collaborative filtering [25,26]. User-based collaborative filtering mainly considers the similarity between users. It predicts the target user's rating for a particular item based on the ratings of items liked by similar users. The basic idea of item-based filtering is to calculate the similarity between items based on the historical preference data of all users, and then to recommend items similar to the user's favorite item to the target user. Currently, a large number of scholars focus on utilizing machine learning models to improve the performance of collaborative filtering technique. Matrix factorization [27], neural network [28], and graphic models [29] are commonly used in combination with collaborative filtering. The most difficult challenge faced by the collaborative filtering technique is the cold start problem when a new user arrives. Since the recommender system does not have any data of new users, it cannot effectively recommend items for new users. In addition, collaborative filtering cannot understand different scenarios, which is unable to capture the specific consumption purpose of users at a specific moment.

Content-based filtering works by evaluating the similarity between items that the user has not seen and items that the user has liked in the past. To generate meaningful recommendation results, content-based filtering uses different models to find similarities between items. It typically uses a vector space model (e.g., term frequency inverse document frequency) or a probabilistic model (naive Bayes classifier, decision tree, and neural network) to model relationships between different items [30–32]. Content-based filtering technology does not need to refer to other user portraits because other user portraits will not affect the final recommendations. Moreover, content-based filtering technology can still adjust the recommendation results in a very short period of time if the user profile changes. The main disadvantage of this technique is that it requires the system to have a deep understanding of the characteristics of the item. Since content-based filtering depends only on the user's past preferences for certain items, users can only get recommendations similar to items in their own profile, hardly getting diversified options.

Hybrid recommender systems combine multiple recommendation algorithms to avoid the problems of a single technique. Burke [33] distinguished hybrid recommender systems into three basic design ideas: monolithic, parallelized, and pipelined. The monolithic paradigm integrates multiple recommendation algorithms into the same algorithm system, and the integrated recommendation algorithm provides a unified recommendation service. The parallelized paradigm utilizes multiple independent recommendation algorithms, each of which generates its own recommendation results, and fuses these recommendation results in the mixing stage to generate the final recommendation result. In the pipelined paradigm, the recommendation result generated by one algorithm is given to another recommendation algorithm as input, and then the recommendation result is generated, which is input to the next recommendation algorithm, and so on.

With the rapid development of mobile commerce, more and more recommendation services occur in dynamically changing contexts, such as user location, access time, current traffic, and other surrounding environments. Traditional personalized recommendation technology is no longer enough to deal with the new impact caused by contextual factors [34,35]. Therefore, a current trend is to integrate and to apply contextual information in traditional recommendation systems to form a context-based recommendation system, so as to accurately and to efficiently provide information resources that not only conform to the current situation of the user but also satisfies the user's preference [36].

In summary, recent recommendation techniques predict the consumer's purchase intention based on product ratings of what other similar users have purchased or what they themselves have purchased. Unlike sparse consumer purchasing data, consumers conduct extensive online search queries before making a purchasing decision. Different from these studies, this paper tried to extract users' explicit content preferences from online query sessions to alleviate the problems mentioned above.

2.2. Online Query Sessions

Online query sessions contain a wealth of valuable information about users' hobbies, preferences and intensions. The content and the quantity of online search queries can be used to predict product or service demand in the era of big data [37]. Choi and Varian [38] showed how to predict near-term values of economic indicators, e.g., automobile sales, unemployment claims, travel destination planning, and consumer confidence, based on Google search data. Yang, Pan, and Song [39] utilized traditional econometric models to predict hotel demand and hotel occupancy in tourist destinations based on web query volumes. Roscoe et al. [15] debated how online search and the holistic stance of a web search toward a consumer product contributed to decision making, and they assessed decision making by combining analyses of online searches with robust choice modeling. Taking bottled water as an example, this approach revealed how different product attributes (e.g., type of product, type of packaging, and cost) affected users purchase intentions in different degrees. Tibau et al. [40] applied the Exploratory Search Knowledge-intensive Process Model to visualize search patterns and to identify best practices associated with users' decision-making processes. They identified four important characteristics of users' decision-making processes while searching online. Liu and Toubia [18] suggested that marketers should focus their efforts on keywords and queries that reflect content preferences that are well aligned with the content they are trying to promote. Codignola et al. [19] found that these browsing data can be saved with cookies and can be used to show customers potentially suitable items.

Although numerous studies have been done to empirically verify that online queries can explicitly express consumers' content preference or can be used to predict product demand, quantitative studies that can estimate content preferences from online queries in an interpretable manner are lacking. In this paper, we propose a novel Aggregated Latent Dirichlet Allocation (ALDA) topic model that can simultaneously learn the potential topics hidden in user's online search queries and the corresponding webpages. Since online query phrases data is sparse, the ALDA model aggregates click documents corresponding to user queries to assist in more accurately learning users' content preferences.

2.3. Recommendation Based on Graph Learning

Graph is becoming a core area of machine learning. Graph learning is widely used to understand the structure of social networks by predicting potential connections, detecting fraud, understanding consumer behavior, or making real-time recommendations. Graph neural network (GNN) techniques have been widely used in recommender systems because most of the information in recommender systems has a graph structure in nature and GNNs have excellent performance in learning graph structures [41,42]. He et al. [43] proposed a light graph convolution network (LightGCN) model that uses user-interacted item records to enhance user representation and interacted user records to enhance item representation. Multi-layer GNNs can simulate the information transfer process and efficiently establish higher-order connections. Li et al. [44] designed a novel feature interaction graph neural network (Fi-GNN) to model sophisticated feature interactions in a flexible and an explicit fashion, which provides good model explanations for click-through rate prediction. Chang et al. [45] proposed a new graph-based geographical latent representation (GGLR) that models geographic influences between the POIs and the transition patterns of user sequence behavior based on spatial and temporal features, which can capture highly non-linear geographical influences from complex user-POI networks.

The GNN methods mentioned above employ pairwise connections between data. However, data structures in real-world applications can go beyond pairwise connections and they can be even more complicated. Feng et al. [46] proposed a hypergraph neural network (HGNN) framework that can deal with complex data correlations by encoding high-order data correlation (beyond pairwise connections) using its degree-free hyperedges. Chen et al. [47] proposed a neural signed hypergraph to extract non-linear relationships among users, items, and features. He et al. [48] proposed a hypergraph click-through rate prediction framework (HyperCTR) that learns item representations based on multi-modal information interactions among users and items. However, existing research focuses on learning user interaction characteristics with products during and after purchase (e.g., purchase and online review), and it ignores the interactive features in the process of users' learning about product information through search queries before they make a purchase (e.g., product information search). However, user association with a product is a coherent process that should not be isolated into different nodes. Only sorting the user's process of searching-understanding-purchasing-using products and finding opportunity points from each stage can help the recommender system to better discover the potential needs of users. In this paper, we develop a hypergraph framework to handle the interaction behavior of consumers in the whole process of shopping (i.e., searching-understanding-purchasing-using).

3. Materials and Methods

In order to make effective recommendations to users, recommendation systems need to solve two problems. One is to predict consumers' product ratings, that is, recommending products with higher predicted scores to target consumers. The second is the interpretation of the recommendation results, that is, explaining the working mechanism of the recommendation system and the specific reasons for recommending a product to consumers in an appropriate way. Since the recommendation process is still a relatively mysterious process for most consumers, a reasonable explanation of the recommendation results is necessary to improve consumers' trust in the recommendation system, which greatly affects consumers' perception and acceptance of recommendation results. The existing recommendation algorithms generally directly rely on the users' overall rating score for products, and the obtained recommendation results are greatly affected by the sparsity of the rating matrix and the cold start problem. This study believes that this situation is mainly caused by the coarse information granularity of the user's product ratings. That is to say, it is impossible for any product to fully meet all the needs of users, and it is impossible for users to have the same degree of preference for all attributes of a product. The recommendation results generated by directly relying on the user's overall ratings cannot reflect the users' preferences for various attributes of the product, and it is difficult to explain the real reasons for the user's preference for the product.

As consumers are more likely to submit online search phrases to search engines to gather information before making an intended purchase decision. They enter keywords to explicitly express their preferences for product attributes. For example, customers formulate queries such as "best laptop for programming" that directly reflect their content preferences for product configurations. Interpreting consumers' search phrases renders a better understanding of their purchase intentions and preferences for product attributes, which is critical for developing an effective personalized recommendation system.

In this paper, we introduce a sustainable recommender system architecture based on fusing a topic model and a hypergraph neural network, which can deal with the interaction behavior of consumers in the whole process of shopping (i.e., searching-understanding-purchasing-using). Figure 1 shows the topic augmented hypergraph neural network (Topic-HGNN) framework for searching-scenario oriented recommendation. First, we adopt a hypergraph to model the multivariate relationship among users, query topics, items, and item features, which aims at mining the connection information existing in the interaction graph domain. Then, we utilize hyperedge corruption [47] to generate a user–query hypergraph and an item–feature hypergraph, and we utilize the hyperedge convolution layer [46] to obtain user embedding and item embedding. Meanwhile, in order to reduce the influence of text noise words by fusing topic information, we specially design an Aggregated Latent Dirichlet Allocation (ALDA) model to jointly extract users' content preference topics from queries and webpages and to apply Latent Dirichlet Allocation model to extract product feature topics from online reviews. Then, we integrate the topic distribution and convolutional embedding to represent each user and item, which can make up for the lack of topic information in traditional convolutional neural networks. Finally, we use multilayer perceptron to calculate the soft match score between query entities and item entities.

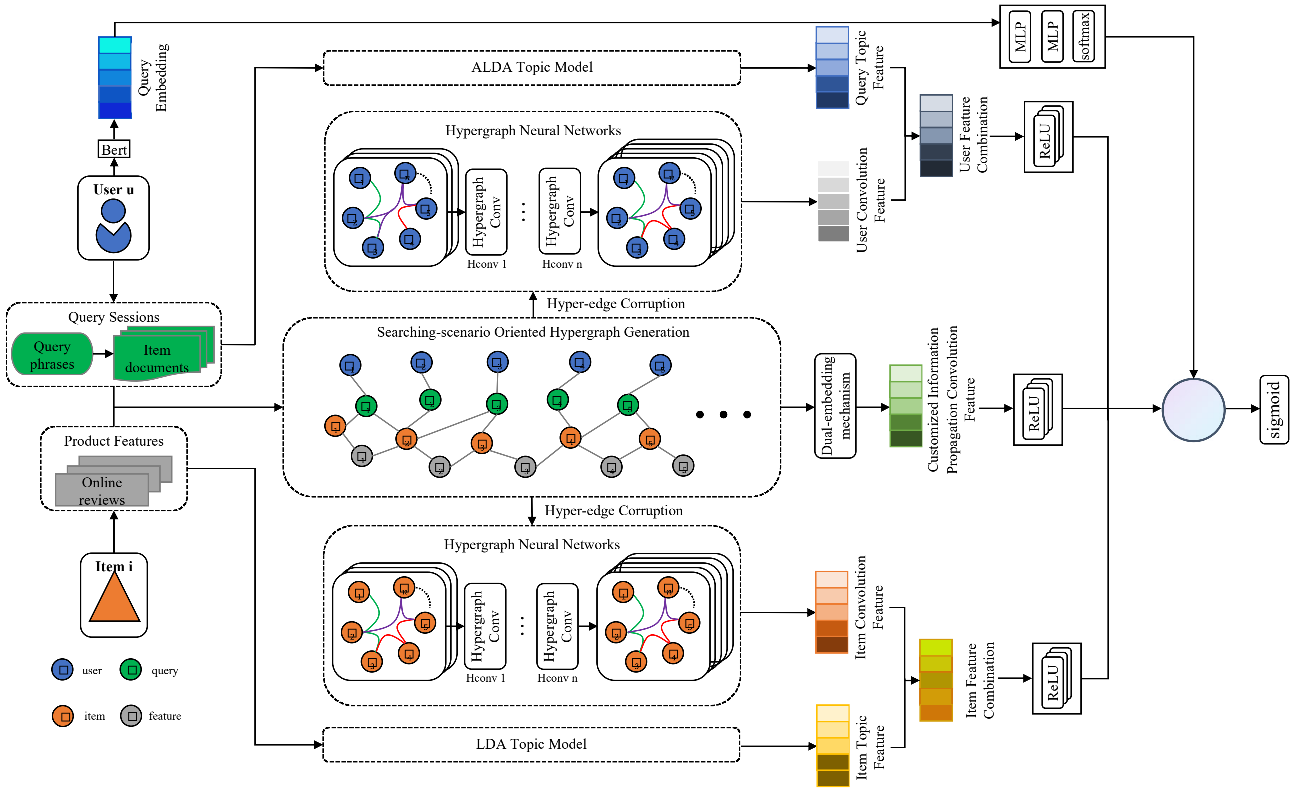

Figure 1. The proposed topic augmented hypergraph neural network (Topic-HGNN) framework for searching-scenario oriented recommendation.

3.1. Searching-Scenario Oriented Hypergraph Generation

Existing research focuses on learning user interaction characteristics with products during and after purchase (e.g., purchase and online review), and ignores the interactive features in the process of users' learning about product information through search queries before they make a purchase (e.g., product information search). However, user association with a product is a coherent process that should not be isolated into different nodes. Only sorting the user's process of searching-understanding-purchasing-using products and finding opportunity points from each stage can help the recommender system to better discover the potential needs of users. Thus, this work considers quaternary relationships between interacting entities (user, query topic, item, and item feature) and employs a hypergraph to model the interaction behavior of consumers in the whole process of shopping.

Let $V = \left\{ V_u, V_q, V_i, V_f \right\}$ denote the vertex set, where V_u represents user vertex, V_q is the query vertex sent by the user, V_i represents item vertices, and V_f is the product feature node extracted from the product online reviews. E represents the set of hyperedges e_j built from V. Each hyperedge "v_u-v_q-v_i-v_f" is a complete purchasing path for the user, which means that user u finds a product i that matches his preference for feature f through query q, and makes a purchase. Thus, $G = (V, E)$ represents a hypergraph, and a hypergraph G can be represented by a $|V| \times |E|$ incidence matrix H, with entries defined as:

$$h(v,e) = \begin{cases} 1, \ v \in e \\ 0, \ v \notin e \end{cases}$$

For a vertex $v \in V$, its degree is defined as $d(v) = \sum_{e \in E} w(e)h(v,e)$, where $w(e)$ represents the weight of the hyperedge e. For an hyperedge $e \in E$, its degree is defined as $\delta(e) = \sum_{v \in V} h(v,e)$. The degree matrices of vertex and hyperedge are represented by the diagonal matrices D_v and D_e, respectively.

3.2. Topic Feature Learning of User and Item

In this section, we introduce the Aggregated Latent Dirichlet Allocation (ALDA) model in detail. ALDA is a bag-of-word model that depicts the semantic relation between user preferences and their online query sessions. Instead of modeling the topic intensities in the query sessions and the topic intensities in the webpages hierarchically [49], the ALDA conjointly models the topic intensities in the query sessions and the topic intensities in the webpages into the same document layer. The data sparsity of online query data is avoided by aggregating corresponding webpages to assist in learning users' content preferences. Consumers' online shopping behavior is usually a learning process. First, users may enter inaccurate keywords to express their needs. Then, users enhance their understanding of products through browsing the search results and adjusting the input keywords. Consumers will repeat this learning process until finding the right product. That is to say, the topics of query keywords and the topic of search results are semantically related to each other. Liu and Toubia [49] assumed the topic intensities in webpages is affected by query keywords while ignoring that webpages can in turn affect the topic intensities in query keywords. Thus, we model the interactive relationship between queries and webpages in ALDA. The graphical representation of ALDA proposed in this paper is illustrated in Figure 2. The main notations in ALDA are listed in Table 1.

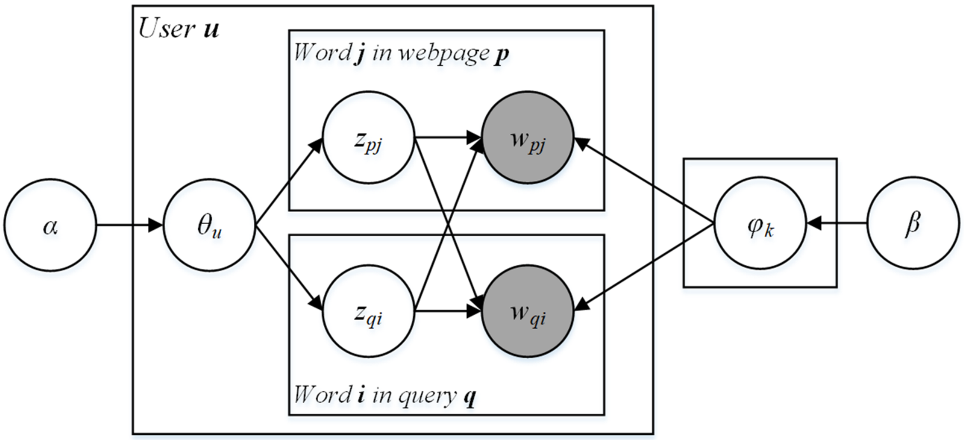

Figure 2. The graphical representation of ALDA model.

Table 1. Summary of the main notations.

Notations	Explanation
$u \in \{1, 2, \cdots, U\}$	Set of users
$q \in \{1, 2, \cdots, Q\}$	Set of user queries
$p \in \{1, 2, \cdots, P\}$	Set of webpages
θ_u	The vector of topic probabilities in users' preferences
α	Dirichlet prior distribution for θ_u
φ_k	The vector of word probabilities for topic k
β	Dirichlet prior distribution for φ_k
z_{qi}	Topic assignment of the ith word in query q
z_{pj}	Topic assignment of the jth word in webpage p
w_{qi}	The ith observed word in query q
w_{pj}	The jth observed word in webpage p

3.2.1. Model Description

First, we introduce the notations of the ALDA model. Supposing that there is a collection of U users in a particular e-commerce platform: $u \in \{1, 2, \cdots, U\}$. The user u entered different queries for a particular search domain: $q \in \{1, 2, \cdots, Q\}$. There are P webpages underlying a particular query q: $p \in \{1, 2, \cdots, P\}$. There are K topics that the user u is interested in: $k \in \{1, 2, \cdots, K\}$. There are V topic words in the vocabulary. w_{qi} represents the ith word in the query q. w_{pj} represents the jth word in the webpage p.

- θ_u denotes the topics probability distribution in user u's preferences.
- φ_k denotes the words probability distribution of the kth topic.
- α is the symmetric Dirichlet prior hyper-parameter for θ_u.
- β is the symmetric Dirichlet prior hyper-parameter for φ_k^q and φ_k^q.
- z_{qi} denotes the topic of the ith word in query q.
- z_{pj} denotes the topic of the jth word in webpage p.
- w_{qi} denotes the ith word in the query q.
- w_{pj} denotes the jth word in the webpage p.

Formally, the generative process of query sessions and webpages based on the ALDA model is described as follows:

Topics: We continue to work on the assumption proposed by Liu and Toubia [49]. Liu and Toubia [49] assumed that search query documents and webpage documents follow the same topic distributions. The topic intensities in the documents are reflected by the words displayed in the documents and each document has different topic intensities. Similar to an LDA, each topic $k \in \{1, 2, \cdots, K\}$ is represented as a topic-word distribution vector φ_k. The vector φ_k follows a *Dirichlet* distribution over V topic words in the vocabulary:

$$\varphi_k \sim Dirichlet(\beta)$$

Queries: To model the ith word w_{qi} observed in the query q, ALDA sequentially samples the topic distribution of the query q and the topic assignment of the ith word in the query q. The generation process of users' query online queries is as follows:

1 For each query q ($q \in \{1, 2, \cdots, Q\}$):

    1.1 Generate topic probabilities θ_u from a homogeneous Dirichlet distribution with parameter α: $\theta_u \sim Dirichlet(\alpha)$

2 For each topic k ($k \in \{1, 2, \cdots, K\}$):

    2.1 Generate φ_k independently from a homogeneous Dirichlet distribution with parameter β: $\varphi_k \sim Dirichlet(\beta)$

3 For each word w_{qi} in the query q:

    3.1 Choose a topic z_{qi} from the K topics with probabilities given by θ_u: $z_{qi} \sim Multinomial(\theta_u)$

3.2 Choose a word w_{qi} from the dictionary with probabilities given by φ_k: $w_{qi} \sim Multinomial(\varphi_k)$

Webpages: To model the *j*th word w_{pj} observed in the webpage *p*, ALDA sequentially samples the topic distribution of the webpage *p* and the topic assignment of the *j*th word in the webpage *p*. The generation process of webpages related to online queries is as follows:

1 For each query p ($p \in \{1, 2, \cdots, P\}$):

1.1 Generate topic probabilities θ_u from a homogeneous Dirichlet distribution with parameter α: $\theta_u \sim Dirichlet(\alpha)$

2 For each topic k ($k \in \{1, 2, \cdots, K\}$):

2.1 Generate φ_k independently from a homogeneous Dirichlet distribution with parameter β: $\varphi_k \sim Dirichlet(\beta)$

3 For each word w_{pj} in the query p:

3.1 Choose a topic z_{pj} from the K topics with probabilities given by θ_u: $z_{pj} \sim Multinomial(\theta_u)$

3.2 Choose a word w_{pj} from the dictionary with probabilities given by φ_k: $w_{pj} \sim Multinomial(\varphi_k)$

3.2.2. Parameter Estimation

It is an intractable task to exactly estimate the parameters θ_u, φ_k. Similar to LDA, we use Gibbs sampling to approximately infer the parameters. First, we need to sample $P(z_{qi}|w_{qi}, w_{pj})$ and $P(z_{pj}|w_{qi}, w_{pj})$ to obtain the topic assignment z_{qi} in query documents and the topic assignment z_{pj} in webpage documents. Thus, the following conditional probability distribution is derived:

$$P(z_{qi} = k, z_{pj} = k | z_{-qi}, z_{-pj}, w_{qi}, w_{pj})$$

$$\begin{aligned}
& P(z_{qi} = k, z_{pj} = k | z_{-qi}, z_{-pj}, w_{qi}, w_{pj}) \\
& \propto P(z_{qi} = k, z_{pj} = k, w_{qi} = t_1, w_{pj} = t_2 | z_{-qi}, z_{-pj}, w_{-qi}, w_{-pj}) \\
= & \int P(z_{qi} = k, z_{pj} = k, w_{qi} = t_1, w_{pj} = t_2, \theta_u, \theta_u, \varphi_k | z_{-qi}, z_{-pj}, w_{-qi}, w_{-pj}) d\theta_u d\theta_u d\varphi_k \\
& = \int P(z_{qi} = k, \theta_u | z_{-qi}, z_{-pj}, w_{-qi}, w_{-pj}) \cdot P(z_{pj} = k, \theta_u | z_{-qi}, z_{-pj}, w_{-qi}, w_{-pj}) \\
& \cdot P\Big(w_{qi} = t_1, \varphi_{k,w_{qi}} \Big| z_{-qi}, z_{-pj}, w_{-qi}, w_{-pj}\Big) \\
& \cdot P\Big(w_{pj} = t_2, \varphi_{k,w_{pj}} \Big| z_{-qi}, z_{-pj}, w_{-qi}, w_{-pj}\Big) d\theta_u d\theta_u d\varphi_{k,w_{qi}} d\varphi_{k,w_{pj}} \\
& = \int P(z_{qi} = k | \theta_u) P(\theta_u | z_{-qi}, z_{-pj}, w_{-qi}, w_{-pj}) \\
& \cdot P(z_{pj} = k | \theta_u) P(\theta_u | z_{-qi}, z_{-pj}, w_{-qi}, w_{-pj}) \\
& \cdot P\Big(w_{qi} = t_1 \Big| \varphi_{k,w_{qi}}\Big) P\Big(\varphi_{k,w_{qi}} \Big| z_{-qi}, z_{-pj}, w_{-qi}, w_{-pj}\Big) \\
& \cdot P\Big(w_{pj} = t_2 \Big| \varphi_{k,w_{pj}}\Big) P\Big(\varphi_{k,w_{pj}} \Big| z_{-qi}, z_{-pj}, w_{-qi}, w_{-pj}\Big) d\theta_u d\theta_u d\varphi_{k,w_{qi}} d\varphi_{k,w_{pj}} \\
& = \int P(z_{qi} = k | \theta_u) Dir(\theta_u | n_{q,-qi} + \alpha) \cdot P(z_{pj} = k | \theta_u) Dir(\theta_u | n_{p,-pj} + \alpha) \\
& \cdot P\Big(w_{qi} = t_1 \Big| \varphi_{k,w_{qi}}\Big) Dir\Big(\varphi_{k,w_{qi}} \Big| n^{q}_{k,-qi} + \beta\Big) \\
& \cdot P\Big(w_{pj} = t_2 \Big| \varphi_{k,w_{pj}}\Big) Dir\Big(\varphi_{k,w_{pj}} \Big| n^{p}_{k,-pj} + \beta\Big) d\theta_u d\theta_u d\varphi_{k,w_{qi}} d\varphi_{k,w_{pj}} \\
& = \int \theta_u^{(k)} Dir\Big(\theta_u \Big| n^{(k)}_{q,-qi} + \alpha\Big) \cdot \theta_u^{(k)} Dir\Big(\theta_u \Big| n^{(k)}_{p,-pj} + \alpha\Big) \\
& \qquad \cdot \varphi_k^{(t_1)} Dir\Big(\varphi_{k,w_{qi}} \Big| n^{q(t_1)}_{k,-qi} + \beta\Big) \\
& \qquad \cdot \varphi_k^{(t_2)} Dir\Big(\varphi_{k,w_{pj}} \Big| n^{p((t_2)}_{k,-pj} + \beta\Big) d\theta_u d\theta_u d\varphi_{k,w_{qi}} d\varphi_{k,w_{pj}} \\
& = \mathrm{E}\Big(\theta_u^{(k)}\Big)^2 \cdot E\Big(\varphi_k^{(t_1)}\Big) \cdot E\Big(\varphi_k^{(t_2)}\Big) \\
& \quad = \frac{n^{(k)}_{q,-qi} + \alpha}{\sum_{k=1}^{K} n^{(k)}_{q,-qi} + K\alpha} \cdot \frac{n^{(k)}_{p,-pj} + \alpha}{\sum_{k=1}^{K} n^{(k)}_{p,-pj} + K\alpha} \cdot \frac{n^{q(t_1)}_{k,-qi} + \beta}{\sum_{v=1}^{V} n^{q(\mathrm{v})}_{k,-qi} + V\beta} \cdot \frac{n^{p((t_2)}_{k,-pj} + \beta}{\sum_{v=1}^{V} n^{p(\mathrm{v})}_{k,-pj} + V\beta}
\end{aligned} \tag{1}$$

Inside, $w_{qi} = t_1$ denotes the ith word in the query q is t_1. $w_{pj} = t_2$ denotes the jth word in the webpage p is t_2. z_{-qi} denotes the topic assignments to all words except the ith word in the query q. z_{-pj} denotes the topic assignments to all words except jth word in the webpage p. w_{-qi} denotes all words except the ith word in the query q. w_{-pj} denotes all words except the jth word in the webpage p. $n_{q,-qi}^{(k)}$ denotes the number of words generated by topic k in the query q excluding the ith word in the query q, $n_{p,-pj}^{(k)}$ denotes the number of words generated by topic k in the webpage p excluding the jth word in the webpage p, $n_{q,-qi} = \left(n_q^{(1)}, n_q^{(2)}, \cdots, n_q^{(k)} - 1, \cdots, n_q^{(K)}\right)$ denotes the number of words generated by topic k in the query q excluding the ith word, $n_{p,-pj} = \left(n_p^{(1)}, n_p^{(2)}, \cdots, n_p^{(k)} - 1, \cdots, n_p^{(K)}\right)$ denotes the number of words generated by topic k in the webpage p excluding the jth word. $n_{k,-qi}^{q(t)}$ denotes the number of times the word t is assigned to the topic k excluding the ith word in the query q, $n_{k,-qi}^{q} = \left(n_k^{q(1)}, n_k^{q(2)}, \cdots, n_k^{q(t)} - 1, \cdots, n_k^{q(V)}\right)$. $n_{k,-pj}^{p(t)}$ denotes the number of times the word t is assigned to the topic k excluding the jth word in the webpage p, $n_{k,-pj}^{p} = \left(n_k^{p(1)}, n_k^{p(2)}, \cdots, n_k^{p(t)} - 1, \cdots, n_k^{p(V)}\right)$.

Algorithm 1 summarizes the overall procedure of Gibbs sampling to estimate the parameters θ_u, φ_k. First, the assignments of topic to each word are initialized according to a uniform distribution. Then, the assignment of topics to each word will be updated by examining Equation (1). Finally, $n_q^{(k)}$, $n_p^{(k)}$, $n_k^{q(v)}$, $n_k^{p(v)}$ can be counted after a sufficient number of iterations. $n_q^{(k)}$ denotes the number of times the topic k occurs in the query q. $n_p^{(k)}$ denotes the number of times the topic k occurs in the webpage p. $n_k^{q(v)}$ denotes the number of times the word v is assigned as a query word to topic k. $n_k^{p(v)}$ denotes the number of times the word v is assigned as a webpage word to topic k.

Here, we only give the derivation of the parameter θ_u, the derivation of other parameters is the same.

$$\begin{aligned} P(\ \theta_u | n_u, \alpha) &= \frac{P(n_u|\theta_u)P(\theta_u|\alpha)}{\int P(n_u|\theta_u)P(\theta_u|\alpha)d\theta_u} \\ &= \frac{Mult(n_u|\theta_u)Dir(\theta_u|\alpha)}{\int Mult(n_u|\theta_u)Dir(\theta_u|\alpha)d\theta_u} \\ &= Dir(\theta_u|\alpha + n_u) \end{aligned}$$

Inside, $n_u = n_q + n_p$

The estimated value of each parameter is:

$$E(\theta_u) = \left(\frac{n_q^{(1)} + n_p^{(1)} + \alpha}{\sum_{k=1}^{K} n_q^{(k)} + n_p^{(k)} + K\alpha}, \cdots, \frac{n_q^{(k)} + n_p^{(k)} + \alpha}{\sum_{k=1}^{K} n_q^{(k)} + n_p^{(k)} + K\alpha}, \cdots, \frac{n_q^{(K)} + n_p^{(K)} + \alpha}{\sum_{k=1}^{K} n_q^{(k)} + n_p^{(k)} + K\alpha}\right) \tag{2}$$

$$E(\varphi_k) = \left(\frac{n_k^{q(1)} + n_k^{p(1)} + \beta}{\sum_{v=1}^{V} n_k^{q(v)} + n_k^{p(v)} + V\beta}, \cdots, \frac{n_k^{q(v)} + n_k^{p(v)} + \beta}{\sum_{v=1}^{V} n_k^{q(v)} + n_k^{p(v)} + V\beta}, \cdots, \frac{n_k^{q(V)} + n_k^{p(V)} + \beta}{\sum_{v=1}^{V} n_k^{q(v)} + n_k^{p(v)} + V\beta}\right) \tag{3}$$

Finally, the topic feature vector of each user can be expressed as θ_u.

Similarly, we can use LDA [21] to mine each product's topic feature vector from its online reviews: θ_i.

Algorithm 1: The Gibbs sampling for ALDA

```
Input : topic number $K$, vocabulary number $V$, document sets, $\alpha$, $\beta$.
Output : $\theta_u$, $\varphi_k$.
1. Initialization
Sample $z_{qi}$, $z_{pj}$ according to the uniform distribution
$n_q^{(k)} = n_q^{(k)} + 1,\ n_q = n_q + 1,\ n_p^{(k)} = n_p^{(k)} + 1,\ n_p = n_p + 1,\ n_k^{q(t)} = n_k^{q(t)} + 1,\ n_k^q = n_k^q + 1,\ n_k^{p(t)} = n_k^{p(t)} + 1,\ n_k^p = n_k^p + 1.$
2. Gibbs sampling
For each query $q$ and webpage $p$ do:
    For each word $w_{qi}$ in query $q$ do:
        (1) $z_{qi} = k \rightarrow n_q^{(k)} = n_q^{(k)} - 1,\ n_q = n_q - 1,\ n_k^{q(t)} = n_k^{q(t)} + 1,\ n_k^q = n_k^q + 1.$
        (2) Sample $z_{qi} = \hat{k} \sim \mathrm{P}\left(z_{qi} = k \middle| z_{-qi}, w_{qi}\right)$ according to Equation (1)
            $n_q^{(\hat{k})} = n_q^{(\hat{k})} - 1,\ n_q = n_q - 1,\ n_{\hat{k}}^{q(t)} = n_{\hat{k}}^{q(t)} + 1,\ n_{\hat{k}}^q = n_{\hat{k}}^q + 1.$
    For each word $w_{pj}$ in webpage $p$ do:
        (1) $z_{pj} = k \rightarrow n_p^{(k)} = n_p^{(k)} - 1,\ n_p = n_p - 1,\ n_k^{p(t)} = n_k^{p(t)} + 1,\ n_k^p = n_k^p + 1.$
        (2) Sample $z_{pj} = \hat{k} \sim \mathrm{P}\left(z_{pj} = k \middle| z_{-pj}, w_{pj}\right)$ according to Equation (1)
            $n_p^{(\hat{k})} = n_p^{(\hat{k})} - 1,\ n_p = n_p - 1,\ n_{\hat{k}}^{p(t)} = n_{\hat{k}}^{p(t)} + 1,\ n_{\hat{k}}^p = n_{\hat{k}}^p + 1.$
3. Parameter estimation
Estimating $\theta_u$, $\varphi_k$ according to Equations (2) and (3)
```

3.3. Convolutional Feature Learning of User and Item

The searching-scenario oriented hypergraph obtains high-order correlations between data, while it contains heterogeneous vertices (i.e., user vertex, query vertex, item vertex, feature vertex). Thus, it is necessary to obtain not only high-order information between paths but also vertex-based semantic information within paths. Therefore, based on the searching-scenario oriented hypergraph, this paper utilizes a dual-embedding mechanism [47] and hyperedge convolution [46] to obtain high-order information between paths and vertex-based semantic information within paths, respectively.

3.3.1. Path Semantic Association Learning

A path contains any number of nodes, these nodes are of the same or different types, so the generated paths have different semantic information. In this paper, dual-embedding mechanism [47] is used to obtain semantic associations among consumers' online queries, their click, purchase, and online review behavior.

The semantic associations among consumers' online queries, their click, purchase, and online review behavior is illustrated as follows. Take the query "harry potter" for example. By using the searching-scenario oriented hypergraph, "harry potter" entered by different users can reach different items such as "harry potter PVC figure", "harry potter book", "harry potter magic wand" or "harry potter LEGO". Obviously, we can obtain more recommendation candidates for the query "harry potter" by using the searching-scenario oriented hypergraph. More importantly, the structural superiority of the searching-scenario oriented hypergraph gives the recommender system a chance to identify different semantic facets of the input search phrases. Similarly, the searching-scenario oriented hypergraph can leverage user behavior to mine related queries with different query phrases. For example, the query "python" entered by user A and the query "Data Analysis" entered by user B can reach the same book "Python for Data Analysis". We can infer from this example that consumer B who bought the book had a preference for using Python even though it was not explicitly expressed in his query. Query-item collaborative filtering greatly solves the item entity recall problem under sparse data.

Therefore, to augment semantic information propagation and training efficiency, we use second-order neighbor relations instead of first-order neighbor relations. To ensure the timeliness of recommended items, we use a strategy of 20% uniform sampling and 80% popularity-based sampling to sample node neighbors.

3.3.2. Convolutional Semantic Features Learning

Not only are there complex associations between paths but the vertices in paths also contain rich semantic information. This paper adopts hyperedge corruption [47] to cut the hyperedge into ordinary edges, which connect the user-query, query-item, and item-feature, respectively. Then, ordinary edges are used to generate association matrices, and the initial weights of the vertices are calculated to generate the hypergraph Laplacian matrix based on meta-path information. Then, this matrix is added to the hypergraph neural network [46] to learn the hyperedge convolution:

$$X^{(l+1)} = \sigma\left(D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} X^{(l)} \Theta^{(l)}\right) \tag{4}$$

where X, D_v, D_e, and Θ is the signal of the hypergraph at l layer, σ denotes the nonlinear activation function.

Therefore, the final convolutional feature can be obtained by connecting L layer features:

$$x_u = \left[X^0, X^1, \cdots, X^L\right] \tag{5}$$

Similarly, for the online reviews of each item i, the corresponding convolutional semantic feature x_i can be obtained through the hypergraph neural network.

3.4. Prediction

For each user u, the obtained convolutional semantic feature x_u and query topic feature θ_u are combined to represent the final user embedding X_u of user u:

$$\mathrm{X}_u = x_u \oplus \theta_u \tag{6}$$

Similarly, the final feature X_i of each item i is:

$$\mathrm{X}_i = x_i \oplus \theta_i \tag{7}$$

Since the number of words in each query is different, the dimension of the word vector matrix is inconsistent, which cannot be processed by the convolutional neural network. Therefore, this paper fixes the number of search phrases in each query as 32, that is, when the number of words is less than 32, it is filled with 0, and when the number of words is greater than 32, the first 32 words are taken. This paper uses BERT to pre-train all the obtained text content to obtain vectors of words X_q.

We want to integrate query embedding, user embeddings, item embedding, and high-order correlations to capture more complex connections. We utilize a deep architecture [48] to predict link relationships between users, queries, items, and features:

$$\hat{y} = \phi_L\left(\phi_{L-1}\left(\ldots \phi_1\left(\left[\mathrm{X}_q; \mathrm{X}_u; \mathrm{X}_i\right]\right)\right)\right) \tag{8}$$

where $[;;]$ concatenates the input vectors and $\{\phi_1, \phi_2, \ldots, \phi_L\}$ are non-linear layers with sigmoid as the active function.

We also take the widely used binary cross-entropy as the loss function:

$$\mathcal{L} = \sum y \log(\hat{y}) + (1-y)\log(1-\hat{y}) + \lambda\omega^2 \tag{9}$$

where ω is the learnable parameters set, λ is the regularization parameter.

4. Results

In order to test the improvement of the proposed Topic-HGNN framework, we conducted experiments based on different datasets obtained from real-world applications. The experiments were designed to verify two aspects of the proposed recommender framework: (1) the quality of topics in online query sessions identified by the ALDA model, and (2) the improvement of recommendation accuracy and novelty of the Topic-HGNN framework that connects the latent topics embedded in consumers' online queries to their click, purchase, and online review behavior.

All empirical evaluations in this paper were implemented on a Dell Precision T5820 workstation with Xeon W-2102 CPU, 8.00 GB RAM, and we chose to implement the program in the Python language.

4.1. Data Description

The public AOL query log dataset (http://www.gregsadetsky.com/_aol-data accessed on 18 September 2019) in the real word is used for experimental verification. This collection consists of 20 M web queries collected from 650 k users over three months in 2006. The data is sorted by anonymous user ID and sequentially arranged. The data set includes {AnonID, Query, QueryTime, ItemRank, ClickURL}. AnonID represents an anonymous user ID number. Query indicates the query issued by the user. QueryTime indicates the time at which the query was submitted for search. If the user clicked on a search result, the rank of the item on which they clicked is listed, and it is marked as ItemRank. If the user clicked on a search result, the domain portion of the URL in the clicked result is listed, which is marked as ClickURL.

We preprocessed the AOL query log dataset before conducting experiments. First, we successively removed query terms containing URL strings, query terms containing special characters, and query terms that did not contain click URLs. Then, we utilized "15 min interval" [50] to derive reasonable session breaks in online queries in order to better investigate the effectiveness of the ALDA model. Finally, we divided each user's search records into training sets and test sets with a ratio of 80%/20%. Part of the AOL query log dataset format is shown in Table 2.

The Retailrocket data (https://www.kaggle.com/retailrocket/ecommerce-dataset accessed on 18 September 2019) was collected from a real-world e-commerce site. The data includes 2,756,101 behavior records from 1,407,580 users, including 2,664,312 views, 69,332 cart additions, and 22,457 purchases.

Table 2. The example of the AOL query log dataset.

AnonID	Query	QueryTime	ItemRank	ClickURL
479	car decals	2006-03-03 23:20:12	4	http://www.decaljunky.com
479	car decals	2006-03-03 23:20:12	1	http://www.modernimage.net
479	car decals	2006-03-03 23:20:12	5	http://www.webdecal.com
479	car window decals	2006-03-03 23:24:05	9	http://www.customautotrim.com
479	car window sponsor decals	2006-03-03 23:27:17	3	http://www.streetglo.net
1020	slot machine tips	2006-04-18 12:43:46	1	http://www.slotadvisor.com
1020	slot machine tips	2006-04-18 12:43:46	4	http://www.thegamblersedge.com
1020	slot machine tips	2006-04-18 12:43:46	8	http://www.gambling.jaxworld.com
1020	slot machine tips	2006-04-18 13:06:52	11	http://www.licensed4fun.com

The entire dataset contains three files: behavioral data file, category relationship file, and item properties file. Each row of data describes the user's behavior on an item at a certain time.

4.2. Evaluation of the ALDA Model

In order to examine the quality of topics in online query sessions identified by the ALDA model proposed in our paper, five typical methods for inferring user preference distributions are selected as baseline methods.

- LDA is a generative probabilistic model in which each document is modeled as a finite mixture over an underlying set of topics and each topic is modeled as an infinite mixture over an underlying set of word distributions [21].
- Twitter-BTM aggregates user-based biterms to learn user specific topic distribution and incorporates a background topic to distinguish user's preference between background words and topical words [51].
- UCIT learns users' short-term and long-term preferences based on their followees' topic distributions, the content of current short texts, and the previously estimated distributions [52].
- HDLDA is a hierarchically dual latent Dirichlet allocation that assumes there is a semantic relation between search query documents and search result documents, and it quantitatively characterizes how consumers translate their content preferences into search queries [49].
- UATM infers topic intensities in user's preference by learning topic intensities in user's preference and topic intensities in followees' preference, which can efficiently alleviate the sparsity problem [53].

We use the AOL query log dataset in this section. By comparing the parameter settings of the above models, we set the hyperparameters $\alpha = 50/K$, $\beta = 0.01$, $\gamma = 0.5$.

4.2.1. Topic Coherence

Topic coherence is mainly used to measure whether the words within a topic are coherent. So, how can these words be considered coherent? If the words support each other, then the group of words is coherent. In other words, if you put words from multiple topics together and cluster them with a perfect cluster, then words from the same topic should be in the same category. *PMI* uses external text datasets to measure the coherence of a topic, which is a fair metric of evaluating the quality of topics extracted by each model. The PMI can be calculated by:

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j) + \epsilon}{p(w_i) \cdot p(w_j)}$$

where w_i and w_j are topic words, and ϵ is a random disturbance term. The larger the value of PMI, the better the coherence between topic words.

To further evaluate the PMI of randomly selected topics, Wikipedia articles downloaded from the official Wikipedia website were used as an auxiliary corpus. We selected the top 5, 10, and 20 words in each topic and calculated the average PMI score. Figure 3 shows the topic coherence results of selected topics learned by each topic discovery model. In the comparison of six models, it clearly shows that the PMI score of our ALDA model is significantly better than the other models. The results demonstrate that topics extracted by our ALDA are more coherent than other models. This is due to the fact that our ALDA conjointly models the topic intensities in the query sessions and the topic intensities in the webpages into the same document layer. The data sparsity of online query data is avoided by aggregating corresponding webpages to assist in learning users interested topics. Because Twitter-BTM and LDA can only model query documents and webpage documents separately, these two models perform worst. Twitter-BTM outperforms LDA because Twitter-BTM inherits BTM's excellent ability to deal with short texts. UCIT and UATM significantly outperforms Twitter-BTM and LDA. This is because UCIT and UATM not only extract topics from content generated by the user themself but also extracts topics from content generated by user clusters that are similar to them. HDLDA can generate more coherent topics than UATM, UCIT, Twitter-BTM, and LDA. This is because HDLDA models query the document and the webpage document in two hierarchical LDA processes. HDLDA can better capture the semantic relation between query and webpage.

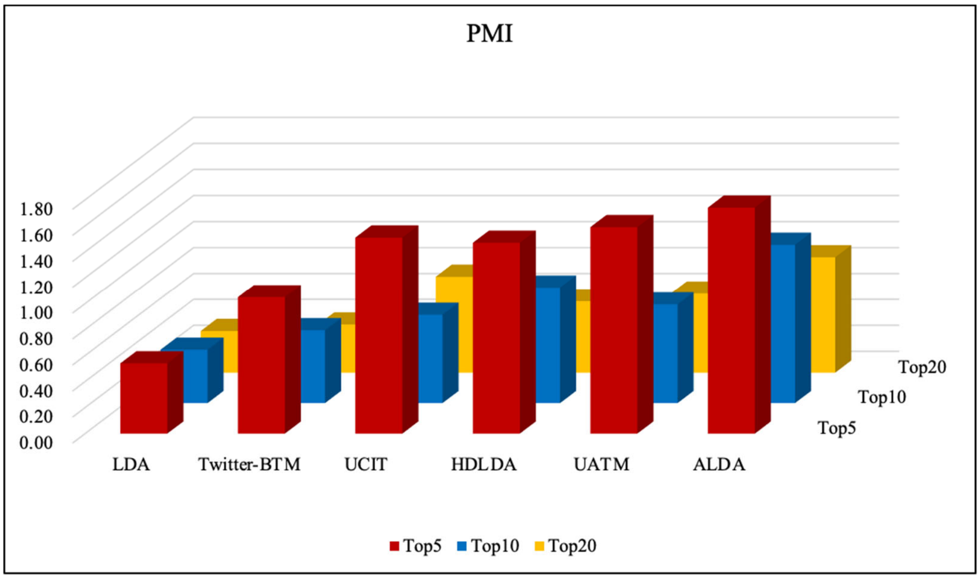

Figure 3. PMI score of each model on AOL query log dataset.

Unlike HDLDA, which models the topic intensities in the query sessions and the topic intensities in the webpages, our ALDA conjointly models the topic intensities in the query sessions and the topic intensities in the webpages into the same document layer. Thus, our ALDA obtained better results than HDLDA.

4.2.2. User's Preference Prediction

We utilize perplexity to compare the accuracy of predicting users' content preference drift estimated by these models. As perplexity in information theory is a measure that is often used to judge probability models or probability distribution prediction samples, we utilize perplexity to evaluate the effect of user's preference inferred by each model. The ability of perplexity is to predict words in new documents, which are not observed. The smaller the value of perplexity, the better the performance of the model in mining user's intention. *Perplexity* can be calculated as follows:

$$Perplexity_{portion}(\mathcal{M}) = \left(\sum_{d=1}^{D} \sum_{i=P+1}^{N_d} p(w_i|\mathcal{M}, w_{1:p}) \right)^{-\frac{1}{\Sigma_{d=1}^{D}(N_d - p)}}$$

where $\mathcal{M}$ is the set of model parameters learned from the training set, d represents the document, and N_d is the number of words in the document.

To make the experimental results more reliable, we sample the observed in the AOL dataset at different scales (from 10% to 90%). It can be seen from Figure 4 that the perplexity of each model gradually decreases with the expansion of the percentage of the observed data. This shows that each model gets better at predicting consumer preferences with the growth of the observed data. Compared with the other five models, the perplexity degree of our ALDA model is the smallest, from 1100 to 2500, which indicates that ALDA preforms best among the six models for identifying consumer interests. This is because ALDA models the interactive relationship between queries and webpages. In reality, a consumer's shopping process is actually a process of understanding and evaluating products. First, users may enter inaccurate keywords to express their needs. Then, users enhance their understanding of products through browsing the search results and adjusting the input keywords. Consumers will repeat this learning process until finding the right product.

That is to say, the topics of query keywords and the topic of search results are semantically related to each other. Thus, modeling this interaction between queries and webpages helps us to more accurately capture changes in consumer's interests and preferences. This is the fundamental reason our model is better than other models in identifying consumers' purchase intentions.

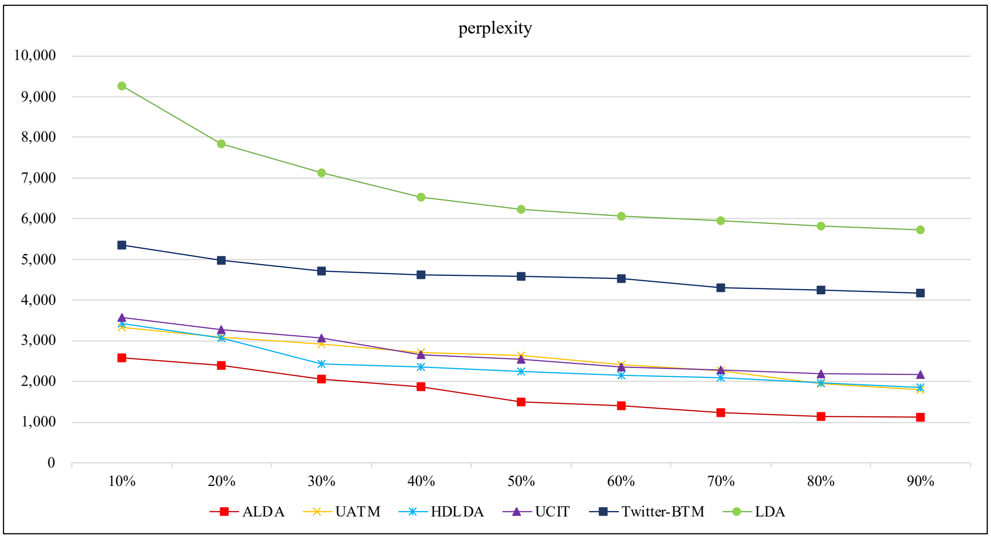

Figure 4. Comparison of user's preference inferring performance.

As LDA and Twitter-BTM do not model how the topics in search queries relate to the topics in the corresponding search results, they obtain the worst performance on understanding users' preference. Both UCIT and UATM learn the topic distributions in the user's content and followees' content, which enables extensive mining and understanding of user's preference and intention, and the experimental results also confirm that UCIT and UATM significantly perform better than LDA and Twitter-BTM. HDLDA models query the document and the webpage document in two hierarchical LDA processes, and they assume that the query document is semantically related to the webpage document, which contributes to a slight lead over UCIT and UATM in understanding the user's interest. Although HDLDA produces good results, it performs worse than ALDA. This is due to HDLDA failing to capture the interactive relationship between queries and webpages. In summary, our ALDA model always outperforms the other comparison models on predicting consumers' purchase intentions.

4.3. Evaluation of Recommendation Results

The proposed Topic-HGNN framework incorporates the topic model into a hypergraph neural network for enhancing user and item embedding representation. Five typical topic model-based recommendation techniques and two state of art neural network-based recommendation methods are selected as baselines.

In order to examine whether the user and the item feature identified by the Topic-HGNN can achieve better personalized recommendations, we utilized precision and diversification to evaluate the recommendation results in detail. The experiment was conducted on the Retailrocket dataset.

- CTR provides an interpretable latent structure for users and items by combining the merits of traditional collaborative filtering and probabilistic topic modeling [54].
- SVD-LDA improves SVD-based recommendations for items with textual content with topic modeling of this content [55].
- CoAWILDA relies on an adaptive online Latent Dirichlet Allocation to model newly available items arriving as a document stream and incremental matrix factorization for collaborative filtering [56].

- AR-LDA uses topic modeling and sequential association rule mining to capture the preference of the user's product changes over time [57].
- EUU-CF extracts topics in Wikipedia by using the LDA model and then uses the topics on user browsing history to extract user preferences [58].
- Graph-CNN is a graph convolutional neural network-based approach to recommend products to users by analyzing their previous interactions [42].
- HyperCTR learns item representations based on multi-modal information interactions among users and items [48].

4.3.1. Precision of Recommendation Results

We adopt two commonly used metrics, *Precision* and *Recall*, to evaluate the accuracy of recommendation results obtained by each recommender method. Precision and Recall are defined as:

$$Precision = \frac{\sum_{u\in U}|R(u)\cap T(u)|}{|R(u)|}$$
$$Recall = \frac{\sum_{u\in U}|R(u)\cap T(u)|}{|T(u)|}$$

where $R(u)$ denotes the recommendation list based on the training dataset, and $T(u)$ denotes the recommendation list based on the test dataset.

To evaluate the accuracy of recommendation results obtained by each recommender technique, we set the number of recommendations from top10 to top100.

Figure 5 shows the comparison of the accuracy of recommendation results generated by each recommender technique.

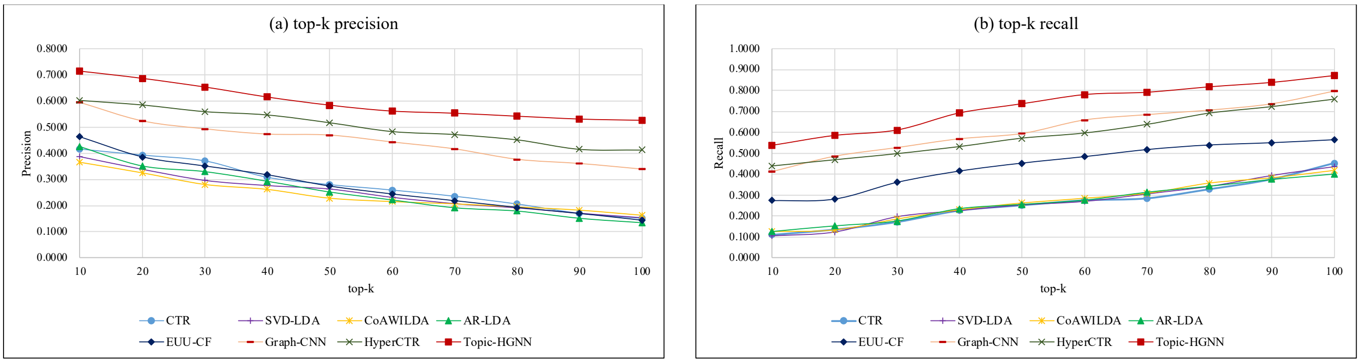

Figure 5. Comparison of recommendation result accuracy: (**a**) precision of top-k items; (**b**) recall of top-k items.

We can observe that the accuracy of recommendation results generated by topic-based methods CTR, SVD-LDA, CoAWILDA, AR-LDA, and EUU-CF are very close to each other and are significantly worse than Graph-CNN, HyperCTR, and Topic-HGNN. This is because topic-based methods focus on improving recommendations for items with textual content. They infer the user's interest based on the user's purchase behavior, which is difficult to refine user preferences for different product attributes and capture high-order correlations between users and items. Different from topic-based recommendation models, Graph-CNN, HyperCTR, and Topic-HGNN infer the user's preference from rich user-product interaction information. Although Graph-CNN and HyperCT also produces good accurate recommendations, it performs worse than Topic-HGNN. This is due to the Graph-CNN and HyperCT only focusing on learning user interaction characteristics with products during and after purchase (e.g., purchase and online review) and ignoring the interactive features in the process of users' learning about product information through search queries before they make a purchase (e.g., product information search). However, user association with a product is a coherent process that should not be isolated into different nodes. Our Topic-HGNN integrates a topic model and a hypergraph neural network, which can deal with the interaction behavior of consumers in the whole process of shopping (i.e., searching-understanding-purchasing-using). Besides, the Topic-HGNN obtains the convolutional semantic features of users and items, and uses the topic model to obtain the corresponding topic features. The result shows that incorporating the topic information from users and items into a convolutional neural network can effectively represent user preferences and item features, which can significantly improve the accuracy of prediction scores. The result also demonstrates the structural superiority of the searching-scenario oriented hypergraph, which gives the recommender system a chance to identify different semantic facets of the input search phrases.

4.3.2. Novelty of Recommendation Results

Only verifying the accuracy of model recommendation results is not enough to explain the personalized effect of a recommendation model. As the collaborative filtering only depends on the user's past purchase behavior, users can only get recommendations similar to items in their own profile and hardly get diversified options. So, experiments are further designed to verify the ability of the recommendation model to discover novel items to the target user. We adopt the novelty metric [59] to measure the ability of recommendation model to find novel items. The lower the Novelty is, the more novel products are recommended. *Novelty* is defined as:

$$novelty = \frac{1}{mk}\sum\nolimits_{u=1}^{m}\sum\nolimits_{i\in L_u} d_i$$

where L_u is the top-k list of a user u, m is the number of users, and d_i is the degree of item i, i.e., the number of users that rated the item i.

We set the number of recommendation items to 10, and experimented on the Retailrocket datasets 32 times each. A smaller novelty value indicated that the recommendation items were more novel.

Figure 6 shows the comparison of the novelty of recommendation results generated by each recommender technique. We can observe that the novelty of recommendation results generated by CTR, SVD-LDA, CoAWILDA, EUU-CF, and AR-LDA are very close to each other and are significantly worse than Graph-CNN, Hyper-CTR, and Topic-HGNN. This is because CTR, SVD-LDA, CoAWILDA, EUU-CF, and AR-LDA infer the user's interest based on the user's historical purchase behavior, which is difficult to discover new products for consumers. This result demonstrates that the topic-based method is significantly worse than the graph-based method. The Topic-based method regards the interaction between users and products as a matrix, and it focuses on mining linear correlation and low-rank information. However, graph-based methods focus on mining interaction information and high-order relation in the graph. Compared with the matrix, the graph can describe more information, such as the link to describe the connection between adjacent vertices, the overall connection between all vertices in the graph, and the link density to describe the community structure in the graph. The graph has a powerful representation ability and the effect of the graph-based method is significantly better than that of the traditional recommendation algorithm.

Our Topic-HGNN is significantly better than Graph-CNN, Hyper-CTR, which demonstrates that Topic-HGNN can identify different semantic facets of input search phrases. Topic-HGNN can obtain semantic associations among consumers' online queries, their click, purchase, and online review behavior that are better than Graph-CNN, Hyper-CTR. Topic-HGNN simultaneously considers heterogeneous interactions and homogeneous interactions in the user purchasing paths, which can better utilize the deep connection information contained in the interactive graph domain, and it is not limited to the observed links.

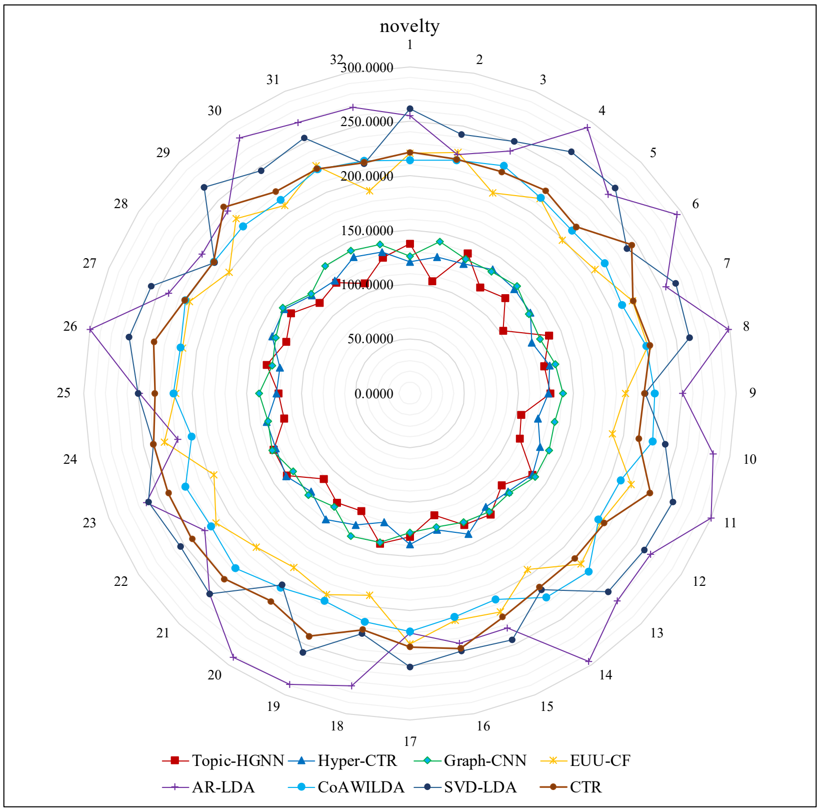

Figure 6. Comparison of recommendation result novelty.

In summary, our Topic-HGNN could improve the novelty of recommendation items without sacrificing accuracy.

4.3.3. Efficiency of Topic-HGNN

The running time and the memory consumption of each method under different query search volumes on Retailrock dataset is shown in Tables 3 and 4. We set the number of recommendation results as 10. From Table 3, it can be seen that the recommendation framework based on a topic model is significantly better than the recommendation framework based on graph learning in terms of running time. Although the recommendation framework based on a topic model is approximately 15% more efficient than the recommendation framework based on graph learning, the quality of the results identified by the recommendation framework based on graph learning on the accuracy, recall, and novelty indicators improved by 53%, 51%, and 46%. This also demonstrates that the method based on graph learning can significantly improve the quality of recommendation results at the expense of a small amount of operating efficiency. Among the three graph-learning-based methods, the running time of our model is slightly higher since our method models the quaternary higher-order relationship among consumers, queries, items, and features. Thus, Topic-HGNN is significantly superior to that of Hyper-CTR and Graph-CNN, when sacrificing a relatively low efficiency.

Table 3. The running time of each method under different query search volumes on Retailrock dataset (the number of recommendation results is 10).

Method	Running Time (10^3 Queries)	Running Time (10^4 Queries)	Running Time (10^5 Queries)
CTR	12.12 ms	2095.54 ms	49,514.16 ms
SVD-LDA	11.87 ms	2294.63 ms	48,510.53 ms
CoAWILDA	11.65 ms	3220.22 ms	46,767.78 ms
AR-LDA	8.02 ms	3076.21 ms	38,881.03 ms
EUU-CF	8.54 ms	3085.96 ms	48,736.47 ms
Graph-CNN	15.57 ms	5014.59 ms	58,294.41 ms
Hyper-CTR	19.56 ms	4963.22 ms	59,324.57 ms
Topic-HGNN	19.67 ms	5038.40 ms	58,290.89 ms

Table 4. The memory consumption of each method under different query search volumes on Retailrock dataset (the number of recommendation results is 10).

Method	Memory Consumption (103 Queries)	Memory Consumption (104 Queries)	Memory Consumption (105 Queries)
CTR	83 MB	347 MB	970 MB
SVD-LDA	89 MB	385 MB	1102 MB
CoAWILDA	96 MB	403 MB	1165 MB
AR-LDA	77 MB	311 MB	928 MB
EUU-CF	79 MB	284 MB	944 MB
Graph-CNN	882 MB	1509 MB	3259 MB
Hyper-CTR	926 MB	1647 MB	3895 MB
Topic-HGNN	974 MB	1802 MB	3971 MB

As can be seen from Table 4, the Topic-HGNN framework does not consume additional memory compared to other graph-based learning methods. This is because the Topic-HGNN is decomposed by hyperedge corruption, importing batches of vertices and hyperedges each time to relieve memory pressure. Therefore, in summary, the Topic-HGNN proposed in this work can produce better recommendation results, while being almost as effective as other graph-based methods.

5. Conclusions

Personalized product recommendation systems are a useful tool adopted by e-retailers to help consumers find items in line with their preferences. Existing research focuses on learning user interaction characteristics with products during and after purchase (e.g., purchase and online review), while ignoring the interactive features in the process of users' learning about product information through search queries before they make a purchase (e.g., product information search). However, users' association with a product is a coherent process that should not be isolated into different nodes. Only sorting the user's process of searching-understanding-purchasing-using products and finding opportunity points from each stage can help the recommender system to better discover the potential needs of users. To this end, we develop a topic augmented hypergraph neural network framework to predict users' purchase intentions by connecting the latent topics embedded in consumers' online queries to their click, purchase, and online review behavior. First, we adopt a hypergraph to model the multivariate relationship among users, query topics, items, and item features, which aims at mining the connection information existing in the interaction graph domain. Then, we utilize the hyperedge corruption to generate a user-query hypergraph and an item-feature hypergraph and utilize the hyperedge convolution layer to obtain user embedding and item embedding. Meanwhile, in order to reduce the influence of text noise words by fusing topic information, we specially design an Aggregated Latent Dirichlet Allocation (ALDA) model to jointly extract users' content preference topics from queries and webpages and apply Latent Dirichlet Allocation model to extract product feature topics from online reviews. Then, we integrate the topic distribution and convolutional embedding to represent each user and item, which can make up for the lack of topic information in traditional convolutional neural networks. Finally, we use multilayer perceptron to calculate the soft match score between query entities and item entities. Extensive empirical evaluations on real-world datasets demonstrate that the proposed framework could improve the novelty of recommendation items without sacrificing accuracy. From the managerial perspective, recommending diversified and novel items to consumers may increase the user's satisfaction, which is conducive to the sustainable development of e-commerce enterprises.

With the rapid development of mobile commerce, more and more recommendation services occur in dynamically changing contexts, such as user location, access time, current traffic, and other surrounding environments. Traditional personalized recommendation technology is no longer enough to deal with the new impact caused by contextual factors. Therefore, our future work will focus on integrating and applying contextual information into the hypergraph framework, which aims at combining context development diagram and user behavior prediction to form a unified and concise context-based recommendation model. In this work, we assumed that search query documents and webpage documents follow the same topic distributions. In reality, search query documents and webpage documents sometimes didn't follow the same topic distributions. Thus, examining the impact in the results when search query documents and webpage documents did not follow the same topic distribution is also a future research topic.

Author Contributions: Conceptualization, X.H. and X.L.; methodology, X.H.; software, X.H.; validation, X.L.; formal analysis, X.L.; investigation, X.H. and X.L.; resources, X.H.; data curation, X.H.; writing—original draft preparation, X.H.; writing—review and editing, X.L.; visualization, X.H.; supervision, X.L.; project administration, X.L.; funding acquisition, X.H. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Fundamental Research Funds for the Central Universities under Grant No. 310422121.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this work can be found at http://www.gregsadetsky.com/_aol-data and https://www.kaggle.com/retailrocket/ecommerce-dataset accessed on 18 September 2019.

Acknowledgments: We would like to thank the editors and the anonymous reviewers for their insightful suggestions.

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

References

1. Pantano, E.; Priporas, C.-V. The effect of mobile retailing on consumers' purchasing experiences: A dynamic perspective. *Comput. Hum. Behav.* **2016**, *61*, 548–555. [CrossRef]
2. Hussien, F.T.A.; Rahma, A.M.S.; Abdulwahab, H.B. An E-Commerce Recommendation System Based on Dynamic Analysis of Customer Behavior. *Sustainability* **2021**, *13*, 10786. [CrossRef]
3. Patten, E.; Ozuem, W.; Howell, K.; Lancaster, G. Minding the competition: The drivers for multichannel service quality in fashion retailing. *J. Retail. Consum. Serv.* **2020**, *53*, 101974. [CrossRef]
4. Lee, D.; Hosanagar, K. How Do Recommender Systems Affect Sales Diversity? A Cross-Category Investigation via Randomized Field Experiment. *Inf. Syst. Res.* **2019**, *30*, 239–259. [CrossRef]

5. Jesse, M.; Jannach, D. Digital nudging with recommender systems: Survey and future directions. *Comput. Hum. Behav. Rep.* **2021**, *3*, 100052. [CrossRef]
6. Archak, N.; Ghose, A.; Ipeirotis, P.G. Deriving the Pricing Power of Product Features by Mining Consumer Reviews. *Manag. Sci.* **2011**, *57*, 1485–1509. [CrossRef]
7. Xie, K.; Wu, Y.; Xiao, J.; Hu, Q. Value co-creation between firms and customers: The role of big data-based cooperative assets. *Inf. Manag.* **2016**, *53*, 1034–1048. [CrossRef]
8. Ebrahimi, P.; Hamza, K.A.; Gorgenyi-Hegyes, E.; Zarea, H.; Fekete-Farkas, M. Consumer Knowledge Sharing Behavior and Consumer Purchase Behavior: Evidence from E-Commerce and Online Retail in Hungary. *Sustainability* **2021**, *13*, 10375. [CrossRef]
9. Koren, Y.; Rendle, S.; Bell, R. Advances in Collaborative Filtering. In *Recommender Systems Handbook*; Springer: New York, NY, USA, 2021; pp. 91–142. [CrossRef]
10. Khojamli, H.; Razmara, J. Survey of similarity functions on neighborhood-based collaborative filtering. *Expert Syst. Appl.* **2021**, *185*, 115482. [CrossRef]
11. Lee, K.; Hwangbo, Y.; Jeong, B.; Yoo, J.; Park, K. Extrapolative Collaborative Filtering Recommendation System with Word2Vec for Purchased Product for SMEs. *Sustainability* **2021**, *13*, 7156. [CrossRef]
12. Lika, B.; Kolomvatsos, K.; Hadjiefthymiades, S. Facing the cold start problem in recommender systems. *Expert Syst. Appl.* **2014**, *41*, 2065–2073. [CrossRef]
13. Son, J.; Kim, S.B. Content-based filtering for recommendation systems using multiattribute networks. *Expert Syst. Appl.* **2017**, *89*, 404–412. [CrossRef]
14. Humphreys, A.; Isaac, M.S.; Wang, R.J.-H. Construal Matching in Online Search: Applying Text Analysis to Illuminate the Consumer Decision Journey. *J. Mark. Res.* **2021**, *58*, 1101–1119. [CrossRef]
15. Roscoe, R.D.; Grebitus, C.; O'Brian, J.; Johnson, A.C.; Kula, I. Online information search and decision making: Effects of web search stance. *Comput. Hum. Behav.* **2016**, *56*, 103–118. [CrossRef]
16. Park, J.; Kim, R.B. A new approach to segmenting multichannel shoppers in Korea and the US. *J. Retail. Consum. Serv.* **2018**, *45*, 163–178. [CrossRef]
17. Kim, D.; Woo, J.; Shin, J.; Lee, J.; Kim, Y. Can search engine data improve accuracy of demand forecasting for new products? Evidence from automotive market. *Ind. Manag. Data Syst.* **2019**, *119*, 1089–1103. [CrossRef]
18. Liu, J.; Toubia, O. Search query formation by strategic consumers. *Quant. Mark. Econ.* **2020**, *18*, 155–194. [CrossRef]
19. Codignola, F.; Capatina, A.; Lichy, J.; Yamazaki, K. Customer information search in the context of e-commerce: A cross-cultural analysis. *Eur. J. Int. Manag.* **2021**, *16*, 28–59. [CrossRef]
20. Liu, Z.; Chen, H.; Sun, F.; Xie, X.; Gao, J.; Ding, B.; Shen, Y. Intent preference decoupling for user representation on online recommender system. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 2575–2582.
21. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
22. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. Recommender systems survey. *Knowl.-Based Syst.* **2013**, *46*, 109–132. [CrossRef]
23. Bellogín, A.; Castells, P.; Cantador, I. Neighbor selection and weighting in user-based collaborative filtering: A performance prediction approach. *ACM Trans. Web* **2014**, *8*, 1–30. [CrossRef]
24. Zhang, Z.; Kudo, Y.; Murai, T. Neighbor selection for user-based collaborative filtering using covering-based rough sets. *Ann. Oper. Res.* **2016**, *256*, 359–374. [CrossRef]
25. Jiang, J.; Lu, J.; Zhang, G.; Long, G. Scaling-up item-based collaborative filtering recommendation algorithm based on hadoop. In Proceedings of the 2011 IEEE World Congress on Services, Washington, DC, USA, 4–9 July 2011; pp. 490–497.
26. Xue, F.; He, X.; Wang, X.; Xu, J.; Liu, K.; Hong, R. Deep Item-based Collaborative Filtering for Top-N Recommendation. *ACM Trans. Inf. Syst.* **2019**, *37*, 1–25. [CrossRef]
27. Ortega, F.; Hernando, A.; Bobadilla, J.; Kang, J.H. Recommending items to group of users using Matrix Factorization based Collaborative Filtering. *Inf. Sci.* **2016**, *345*, 313–324. [CrossRef]
28. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 May 2017; pp. 173–182.
29. Thakur, S.; Sing, J. Online product prediction and recommendation using probability graphical model and collaborative filtering: A new approach. In Proceedings of the 2011 IEEE Recent Advances in Intelligent Computational Systems, Trivandrum, India, 22–24 September 2011; pp. 151–156.
30. Hu, Y.; Guo, C.; Ngai, E.W.; Liu, M.; Chen, S. A scalable intelligent non-content-based spam-filtering framework. *Expert Syst. Appl.* **2010**, *37*, 8557–8565. [CrossRef]
31. Philip, S.; Shola, P.; Ovye, A. Application of Content-Based Approach in Research Paper Recommendation System for a Digital Library. *Int. J. Adv. Comput. Sci. Appl.* **2014**, *5*, 37–40. [CrossRef]
32. Shahi, T.B.; Yadav, A. Mobile SMS spam filtering for Nepali text using naïve bayesian and support vector machine. *Int. J. Intell. Sci.* **2014**, *4*, 24–28. [CrossRef]
33. Burke, R. Hybrid Recommender Systems: Survey and Experiments. *User Model. User-Adapt. Interact.* **2002**, *12*, 331–370. [CrossRef]
34. Annunziata, G.; Colace, F.; De Santo, M.; Lemma, S.; Lombardi, M. ApPoggiomarino: A Context Aware App for e-Citizenship. In Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS (2)), Rome, Italy, 25–28 April 2016; pp. 273–281.

35. Colace, F.; Lemma, S.; Lombardi, M.; Pascale, F. A Context Aware Approach for Promoting Tourism Events: The Case of Artist's Lights in Salerno. In Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS (2)), Porto, Portugal, 26–29 April 2017; pp. 752–759.
36. Ricci, F.; Shapira, B.; Rokach, L. Recommender systems: Introduction and challenges. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2015; pp. 1–34.
37. Li, X.; Pan, B.; Law, R.; Huang, X. Forecasting tourism demand with composite search index. *Tour. Manag.* **2017**, *59*, 57–66. [CrossRef]
38. Choi, H.; Varian, H. Predicting the Present with Google Trends. *Econ. Rec.* **2012**, *88*, 2–9. [CrossRef]
39. Yang, Y.; Pan, B.; Song, H. Predicting Hotel Demand Using Destination Marketing Organization's Web Traffic Data. *J. Travel Res.* **2014**, *53*, 433–447. [CrossRef]
40. Tibau, M.; WM Siqueira, S.; Pereira Nunes, B.; Bortoluzzi, M.; Marenzi, I.; Kemkes, P. Investigating users' decision-making process while searching online and their shortcuts towards understanding. In *International Conference on Web-Based Learning*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 54–64.
41. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef] [PubMed]
42. Shafqat, W.; Byun, Y.-C. Enabling "Untact" Culture via Online Product Recommendations: An Optimized Graph-CNN based Approach. *Appl. Sci.* **2020**, *10*, 5445. [CrossRef]
43. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 639–648.
44. Li, Z.; Cui, Z.; Wu, S.; Zhang, X.; Wang, L. Fi-gnn: Modeling feature interactions via graph neural networks for ctr prediction. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 539–548.
45. Chang, B.; Jang, G.; Kim, S.; Kang, J. Learning graph-based geographical latent representation for point-of-interest recommendation. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 135–144.
46. Feng, Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3558–3565.
47. Chen, X.; Xiong, K.; Zhang, Y.; Xia, L.; Yin, D.; Huang, J.X. Neural Feature-aware Recommendation with Signed Hypergraph Convolutional Network. *ACM Trans. Inf. Syst.* **2020**, *39*, 1–22. [CrossRef]
48. He, L.; Chen, H.; Wang, D.; Jameel, S.; Yu, P.; Xu, G. Click-Through Rate Prediction with Multi-Modal Hypergraphs. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Gold Coast, Australia, 1–5 November 2021; pp. 690–699.
49. Liu, J.; Toubia, O. A Semantic Approach for Estimating Consumer Content Preferences from Online Search Queries. *Mark. Sci.* **2018**, *37*, 930–952. [CrossRef]
50. He, D.; Göker, A. Detecting session boundaries from web user logs. In Proceedings of the BCS-IRSG 22nd annual Colloquium on Information Retrieval Research, Lisbon, Portugal, 14–17 April 2020; pp. 57–66.
51. Chen, W.; Wang, J.; Zhang, Y.; Yan, H.; Li, X. User based aggregation for biterm topic model. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Beijing, China, 26–31 July 2015; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 489–494.
52. Liang, S.; Yilmaz, E.; Kanoulas, E. Collaboratively Tracking Interests for User Clustering in Streams of Short Texts. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 257–272. [CrossRef]
53. Shi, L.; Song, G.; Cheng, G.; Liu, X. A user-based aggregation topic model for understanding user's preference and intention in social network. *Neurocomputing* **2020**, *413*, 1–13. [CrossRef]
54. Wang, C.; Blei, D.M. Collaborative topic modeling for recommending scientific articles. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 448–456.
55. Nikolenko, S. SVD-LDA: Topic modeling for full-text recommender systems. In *Mexican International Conference on Artificial Intelligence*; Springer: Cham, Switzerland, 2015; pp. 67–79.
56. Al-Ghossein, M.; Murena, P.A.; Abdessalem, T.; Barré, A.; Cornuéjols, A. Adaptive collaborative topic modeling for online recommendation. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2–7 October 2018; pp. 338–346.
57. Kang, S.Y.; Kim, J.K.; Choi, I.Y.; Kang, C.D. A Topic Modeling-based Recommender System Considering Changes in User Preferences. *J. Intell. Inf. Syst.* **2020**, *26*, 43–56.
58. Rajendran, D.P.D.; Sundarraj, R.P. Using topic models with browsing history in hybrid collaborative filtering recommender system: Experiments with user ratings. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100027. [CrossRef]
59. Lü, L.; Medo, M.; Yeung, C.H.; Zhang, Y.C.; Zhang, Z.K.; Zhou, T. Recommender systems. *Phys. Rep.* **2012**, *519*, 1–49. [CrossRef]