An Efficient Service Recommendation Algorithm for Cyber-Physical-Social Systems

Xiaoyan Chen, Wei Liang, Jianbo Xu, Chong Wang, Kuan-Ching Li, *Senior Member, IEEE*, and Meikang Qiu, *Senior Member, IEEE*

***Abstract*—Cyber-Physical-Social System (CPSS) technology closely integrates and coordinates computing resources, physical resources, and social information to provide humans with efficient and convenient services. Service recommendation technology satisfies the individual needs of users by collecting, processing, and analyzing user characteristic data. In recent years, service recommendation algorithms based on social networks and collaborative filtering have been widely used in CPSS. However, this type of service recommendation algorithm cannot effectively use the aliasing and auxiliary information in the service, nor can it generate different service recommendation schemes according to different scenarios. We propose a collaborative filtering service recommendation algorithm that combines heterogeneous information networks and topic models for CPSS. The proposed algorithm designs a more effective functional similarity measurement method through the word vectors learned by Word2vec. Then, we use the topic model to cluster the scenario-based service (SBS), thereby greatly reducing the time required for service recommendation. Experimental analysis results show that we finally determine the best combination of Mashup similarity between the two scenarios with better recommendation efficiency and accuracy. Thus, the proposed recommendation algorithm can effectively improve the quality of service of CPSS, and provide a basis for the personalized service of CPSS.**

***Index Terms*—Cyber-physical-social systems (CPSS), recommendation algorithms, scenarios based service (SBS), collaborative filtering.**

Manuscript received 20 December 2020; revised 15 May 2021; accepted 21 June 2021. Date of publication 7 July 2021; date of current version 28 October 2022. This work was partially supported by the National Natural Science Foundation of China under Grant 62072170, the Fundamental Research Funds for the Central Universities under Grant 531118010527 and the Fujian Province Educational Research Projects of Young and Middle-aged Teachers under Grant JAT200456. Recommended for acceptance by Dr. Hai Jiang *(Corresponding author: Xiaoyan Chen.)*

Xiaoyan Chen is with the School of Software Engineering, Xiamen University of Technology, Xiamen 361024, China (e-mail: cxy@xmut.edu.cn).

Wei Liang and Chong Wang are with the College of Computer Science, and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: weiliang99@hnu.edu.cn; f2010w0139@hnu.edu.cn).

Jianbo Xu is with the School of Computer Science, and Engineering, Hunan University of Science, and Technology, Xiangtan 411201, China (e-mail: jbxu@hnust.edu.cn).

Kuan-Ching Li is with the Department of Computer Science, and Information Engr. (CSIE), Providence University, Shalu 43301, Taiwan (e-mail: kuancli @pu.edu.tw).

Meikang Qiu is with the Department of Computer Science, Texas A&M University-Commerce, Commerce, TX 75428 USA (e-mail: qiumeikang @yahoo.com).

Digital Object Identifier 10.1109/TNSE.2021.3092204

I. Introduction

CYBER- Physical-Social System (CPSS) is an emerging paradigm that integrates physical perception systems, human behavior trajectory information, and network communication and control technologies [1]. CPSS is proposed based on Cyber-Physical Systems (CPS) [2]. Compared with CPS, CPSS adds complex human and social factors. The system realizes the connection of physical system and information system through sensor network and realizes the relationship between social system and information system through social sensor network [3]. As shown in Fig. 1, CPSS collects and analyzes information resources in cyberspace and combines the hardware advantages of physical systems to provide people with convenient and efficient personalized services [4][5]. The service recommendation algorithm can provide diversified services for users according to their characteristic information. CPSS integrates various cutting-edge technologies and is widely used in business services, medical health [6], and traffic safety [7]. With the rapid development of information technology, more wireless electronic devices have been used. CPSS collects a large amount of data information from the social environment through physical sensors and networks, and it can effectively improve the service quality of the enterprise by processing and analyzing these data. However, these data are not only huge but also complex. Filtering out valuable information from these data and effectively processing them are keys to improving the service quality. The data contain a large amount of user characteristic information, and protecting these data is equally important [9][10].

Service-oriented computing (SOC) is a unique computing method that adds new design principles, design techniques, and architecture models; it uses the Web service framework to build low-cost, highly reliable, and efficient applications. The realization of SOC requires service-oriented architecture (SOA) [11] as the technical basis. SOA is a service architecture, in which data exchanges between services are carried out through a custom interface; it has the characteristics of coarse-grained and loose coupling. The widespread use of SOA can enhance the flexibility and reliability of enterprise systems. Web service is a remote invocation technology, and it is the most suitable technology for implementing SOA. In addition, it can apply SOA technology to practice. Providing users with accurate and personalized services through this information has become increasingly difficult due to the massive amount of data information in the CPSS system. Thus, designing an efficient and accurate service recommendation algorithm is necessary.

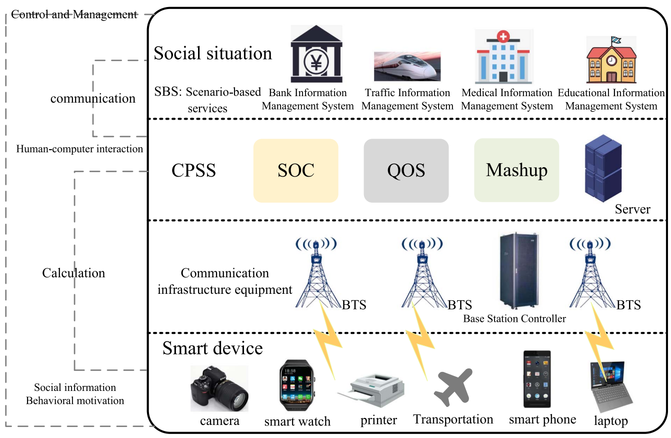


Fig. 1. Cyber-Physical-Social Systems architecture.

Mashup [12] is a Web-based application that integrates information from multiple sources to provide different value-added services. Mashup has strong data compatibility and can support multiple types of data formats, including Web services, APIs, and HTML. Mashup technology is built based on services and information shared by Web applications and then combined to construct new Web applications. However, faced with a large number of Web services, researchers spend considerable time in the selection process. Therefore, selecting a suitable set of Web services from a network database to form Mashup with different functions has become an important issue in the field of service recommendation.

Collaborative filtering algorithm is a widely used recommendation algorithm. It matches users with similar preferences by calculating the similarity between users and builds a list of interests of target users in items. Collaborative filtering algorithm includes two processes of online collaboration and offline filtering. In the online collaboration process, online data analysis is used to find items that users may like, and then in the offline filtering process, some items not worth recommending, such as those with low recommendation scores, are filtered out. However, the recommendation algorithm based on collaborative filtering does not consider the Web service recommendation scenario and cannot perform accurate service recommendation according to specific Mashup requirements.

In this paper, we propose a collaborative filtering service recommendation algorithm that integrates heterogeneous information networks and topic models. The proposed recommendation algorithm can effectively utilize the data information in CPSS and its main contributions are as follows:

1) This paper preprocesses the Mashup in the dataset offline and uses the Latent Dirichlet Allocation (LDA) topic model to cluster the Mashup in the dataset.
2) The proposed algorithm integrates various information about Mashup and Web service and uses word embedding technology [13] to learn the vector representation of words from the text content of Mashup and service.
3) This paper uses Bayesian Personalized Ranking (BPR) algorithm to optimize the combined weight of similarity degree of various Mashups on different meta-paths.

The organization of the remaining chapters are arranged as follows: Section II reviews recent related work. In Section 3, we illustrate the algorithm framework. In Section 4, we illustrate the relevant theoretical knowledge. In Section 5, we propose a collaborative filtering service recommendation algorithm integrating heterogeneous information networks and a topic model. Section 6 shows the experiment and evaluation, and Section 7 concludes the paper.

II. Related Work

The ability of CPSS to collect information has been greatly increased due to the widespread use of portable devices [14], [15]. The service recommendation algorithm analyzes this information to provide users with personalized services. The web service recommendation method based on collaborative filtering recommends services based on the call history of Mashup, user similarity, or service similarity. Qiu *et al.* concentrated on the issue of resource allocations in IoT and utilized the satisfactory level of Quality of Experience (QoE) to achieve intelligent content-centric services [16][17]. Zheng *et al.* [18] proposed a method to predict the lost QoS information by decomposing the integration matrix of neighbor users. Liu *et al.* [19] proposed a QoS-aware Web service recommendation method based on location-aware collaborative filtering. Zhang *et al.* [20] proposed a recommendation algorithm based on group recommendation, emotional awareness recommendation, and multidimensional preference modeling; it can solve the problem of data sparseness in the CPSS system. However, this algorithm cannot make better recommendations for different application scenarios.

To deal with problems, such as information overload in the CPS system, Liu *et al.* [21] proposed an adaptive recommendation algorithm that can detect data changes and overcome the problem of data aging to improve the accuracy of the recommendation algorithm. However, this algorithm requires considerable time to determine the best order of recommendations with high time cost. Jain *et al.* [22] incorporated the following three factors into the service recommendation process: the function of the API, the call history of the existing Mashup about the API, and the popularity of the API. Then, they used the probabilistic topic model, matrix factorization-based collaborative filtering, and Bayes theorem to recommend suitable APIs for the creation of Mashups. Liang *et al.* [23] proposed a recommendation algorithm based on collaborative filtering and meta-path similarity using heterogeneous relationships related to services. The algorithm can learn model parameters with implicit feedback data and has good recommendation accuracy and recommendation efficiency. Samanta *et al.* [24] recommended services by using the Probabilistic Matrix Factorization (PMF) model according to service usage history and solved the problem of the cold startup of new Mashups in the nearest neighbor method. To solve the problem of data sparseness and redundancy, Chen *et al.* [25] proposed a recommendation algorithm based on location perception and user interest. The algorithm uses a deep neural network to extract location and user characteristic information; it has good recommendation accuracy and low time cost. However, this algorithm is only applicable to the field of news recommendation and has limitations.

Bai *et al.* [26] proposed a web service recommendation method based on deep learning; the method has demonstrated the advantages of applying deep learning technology in the field of service recommendation. To recommend suitable applications to users, Xie *et al.* [27] used a heterogeneous information network to propose a service recommendation algorithm based on a weighted meta-graph. The algorithm uses meta-graph to capture complex semantic information and then uses weighted meta-graph to calculate user semantic similarity with the application. The development of the Internet of Things [28] has greatly increased the amount of available data. This information includes a large amount of useless redundant information, which has increased the difficulty of system service recommendation. Lei *et al.* [29] used the potential Latent Dirichlet Allocation (LDA) to analyze the service description and study the relationship between content and location information. In this algorithm, the combination of LDA and Word2VEC effectively improves the accuracy and speed of the service. This algorithm has higher requirements for data, and its performance is poor in a big data environment. Therefore, to improve the recommendation performance in the big data environment, Yang *et al.* [30] proposed a recommendation algorithm based on collaborative deep learning; it effectively improves the detection performance of the item content model in the recommendation system. However, the training efficiency of the model of this algorithm is low.

To improve the ability to analyze CPSS data, a new service recommendation method based on heterogeneous information networks and word embedding technology is proposed. First, we design a more effective functional similarity measurement method based on word vectors learned through Word2vec. Second, we use the topic model to cluster Scenario-Based Service (SBS), thereby greatly reducing the time required for service recommendation by searching for Mashups that have similar functional requirements to Mashups in only few most similar clusters.

III. Relevant Theoretical Knowledge

The framework of the entire algorithm is divided into offline and online modules. The offline module clusters Mashups based on the topic model and uses each Mashup and Web service. According to the work by the offline module, the online module uses the collaborative filtering recommendation algorithm to recommend suitable Web services for the functional requirements of Mashup. The proposed Web service recommendation algorithm specifically includes the following steps:

1) Apply the LDA topic model to Mashups, and group Mashups according to their potential topics;
2) Use Heterogeneous Information Network (HIN) to model Mashups, Web services, Mashups, and service attributes, and Mashup service composition relationships, and calculate The similarity of Mashup on a set of meta-paths;
3) Use BPR algorithm to optimize a set of semantic similarity weight values used in service recommendation;
4) Preprocess the content of Mashup Requirement (MRE);
5) Use the topic model to infer the topic distribution of the MRE, and restrict the candidate Mashup strictly to the Mashup cluster of related topics;
6) Calculate the similarity value between the MRE and each candidate Mashup;
7) Use collaborative filtering technology to predict the MRE pair The rating of all the web services in the service set S, and the service recommendation list is generated according to the size of the rating.

To better understand the collaborative filtering service recommendation algorithm that combines HIN and the proposed LDA, we introduce the relevant theoretical knowledge behind the algorithm, and then the proposed service recommendation algorithm.

A. Heterogeneous Information Network

Heterogeneous Information Network (HIN) is used to calculate path-based semantic relevance information between users and items, such as the similarity based on meta-paths, to assist recommendation tasks. HIN can simulate complex objects and their rich relationships in the recommendation system, and represent the rich auxiliary data in the system by modeling the heterogeneous characteristics of data. Some concepts related to HIN are defined as follows.

Concept 1: According to the description of heterogeneous information network in [31], a heterogeneous information network can be defined as $G = \{\nu, \varepsilon\}$, ν represents a set of objects, ε represents the relationship set. The heterogeneous information network is also associated with the object type mapping function $\phi : \nu \rightarrow A$ and the relation type mapping function $\psi : \varepsilon \rightarrow R$. A and R respectively represent a collection of predefined objects and link types, and $|A| + |R| > 2$. Due to the complexity of HIN, scholars have proposed the concept of Network Schema to describe the meta-structure of the network.

Concept 2: The network schema is expressed as $S = \{A, R\}$. It is a meta-template of the information network $G = \{\nu, \varepsilon\}$. The network has an object type mapping $G = \{\nu, \varepsilon\}$ and a link type mapping $\psi : \varepsilon \rightarrow R$, in other words, the network is on the object type A, and the relationship between the edges is taken from the directed graph of the relationship set R.

There are different types of connections between objects in a heterogeneous information network to represent different relationships between them. The connection between User1-User2 represents the neighbor relationship between two users, and the connection between User-Item represents the rating relationship.

Concept 3: In a heterogeneous information network, objects are connected by different semantic paths, and we call these paths meta-paths. A meta-path ρ defined based on the network mode $S = \{A, R\}$ can be expressed as$A_1 \overset{R_1}{\rightarrow} A_2 \overset{R_2}{\rightarrow} \cdots \overset{R_l}{\rightarrow} A_{l+1}$ (can be abbreviated as$A_1 A_2 \cdots A_{l+1}$), which represents the compound relationship$R_1 \circ R_2 \circ \cdots \circ R_l$ between objects A_1 and A_{l+1}.

Generally speaking, objects can be connected through multiple meta-paths, such as "user-user"(U-U) and "user-project-user" (U-M-U). Different meta-paths usually express different semantics. For example, the U-U path indicates the user relationship between two users (such as the similarity relationship), while the U-M-U path indicates the item browsing/evaluation history between the two users.

In the recommender system, various information involved in the system can be modeled through $G = \{\nu, \varepsilon\}$. In a heterogeneous information network for recommendation systems, the focus is mainly on two entities: users, items, and the relationship between them (ie rating relationship). Assuming that $U \subset \nu$ and $I \subset \nu$ denote users and items, respectively Set, triples $<u, i, r_{u,i}>$ represents user u's rating for the item $r_{u,i}$, $R = \{<u, i, r_{u,i}>\}$ represents the set of evaluation records. The goal of the recommendation system is to predict the rating $r_{u,i'}$ of the user $U \subset \nu$ for the browsed/evaluated items based on the heterogeneous information network $G = \{\nu, \varepsilon\}$ composed of existing $U \subset \nu$, $R \subset \varepsilon$ and existing information.

B. LDA Topic Model

In this era of information explosion [32], [33], people need to obtain useful media content. The LDA topic model [34] analyzes a large amount of unlabeled information from the text and condenses the text topic. The topic in the text processing field is the probability distribution information of the words on the word set. The topic model essentially describes the formal statistical relationship between a set of observed or latent random variables. It focuses on the process of generating text content based on the probability of the text topic.

The LDA model describes the process of generating a text in the database. Initially, we assume that the database has K topic distributions, and each topic distribution is a K item distribution containing V elements, and each element is a word; V indicates all terms in the corpus used when training the model total. Assuming that β_k represents the vector of the multinational distribution of the k-th topic, the vector dimension is the total number of terms in the corpus $|\beta_i| = V$.

The main task of the LDA topic model is to extract the topic of a document from the text collection in the database. To formally describe the LDA topic model, first restate the generation process of the following text, for each text:

(a) LDA assumes that the prior distribution of document topics is the Dirichlet distribution, and the topic distribution of document i is sampled from the Dirichlet distribution $\theta_i = Dirichlet(\overrightarrow{\alpha})$, where $\overrightarrow{\alpha} \in R^K$ is the hyper-parameter of the distribution, and K represents the number of topics;

(b) LDA assumes that the prior distribution of the words in the topic is the Dirichlet distribution. Sampling from the Dirichlet distribution of the topic to generate the topic k, and the word distribution is $\beta_k = Dirichlet(\overrightarrow{\eta})$, where $\overrightarrow{\eta} \in R^V$ is the distribution hyper-parameter, and V represents the total number of all terms in the vocabulary;

(c) For the nth word in any text i in the data, we can get its topic number z_{dn} from the topic distribution θ_i as $z_{dn} = multi(\theta_i)$;

(d) The probability distribution of sampling the word w_{in} from the word distribution is $w_{in} = multi(z_{dn})$;

The k-dimensional Dirichlet distribution is on the k-dimensional polynomial parameters. Based on the probability density function of k-1 dimensional Euclidean space, which is defined as:

$$\mathrm{Dirichlet}(\vec{p}\,|\,\vec{\alpha}) = \frac{1}{A(\alpha)} \prod_{k=1}^{K} p_k^{\alpha_{k-1}} \tag{1}$$

$$\mathrm{A}(\boldsymbol{\alpha}) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\alpha_0)}, \quad \alpha_0 = \sum_{k=1}^{K} \alpha_k, \quad K \geq 3 \tag{2}$$

The central inference problem of LDA is to determine the posterior distribution of latent variables based on the relevant information of a given text:

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)}. \tag{3}$$

C. The Bayesian Personalized Sorting Algorithm

1) BPR Modeling Ideas: Suppose U is the set of all users and I is the set of all items. Suppose that the item recommended to the user contains both items i and j, and the user selects i, then we get a triple $<u, i, j>$. To simplify the description, we use $>_u$ to describe preference of the user U for the project. So the triple $<u, i, j>$ can be expressed as $i>_u j$, indicating that in the project i and j, the user prefers the project i.

2) Optimization of Model Parameters: In the Bayesian personalized sorting algorithm, the sorting matrix of user set U for item set I is $R_{|\mathrm{U}|\times|\mathrm{I}|}$, and $|U|$ and $|I|$ respectively represent the total number of users and items. Bayesian personalized sorting algorithm expects to decompose the matrix R into the user feature matrix $W_{|U|\times\mathrm{k}}$ and project feature matrix $H_{|I|\times k}$, so that:

$$R_{|U||I|} \approx W_{|U||I|} H_{|U||I|}^T \,| = \hat{R} \tag{4}$$

where $\hat{R}$ is the predictive ranking matrix of user set U for item set I. The goal of the Bayesian personalized ranking algorithm is to find suitable matrices W and H. The algorithm solves the model parameters W and H according to the maximization posterior estimation $P(W, H|>_u)$, according to the assumptions of Bayesian theory we have:

$$P(W, H|>_u) = \frac{P(>_u|W, H)P(W, H)}{P(>_u)}, \tag{5}$$

where $>_u$ represents the user's overall ranking of the product collection. Since the user's preference for items has nothing to do with other users, $P(>u)$ is a fixed value, so we get:

$$P(W, H|>_u) \propto P(>_u|W, H)P(W, H), \tag{6}$$

where $P(>u|W,H)$ is affected by the data set used for training, while W,H is only related to users, so $P(>_u|W,H)$ can be rewritten as follows:

$$\prod_{u\in U} P(>_u|W,H) = \lambda(1 - P(i>_u j|W,H))^{1-\delta(i>_u j)} \tag{7}$$

$$\lambda = \prod_{(u,i,j)\in(U\times I\times I)} P(i>_u j|W,H)^{\delta(i>_u j)} \tag{8}$$

where $\delta()$ function is shown in Formula (9):

$$\delta(b) = \left\{ \begin{matrix} 1 \; if \; b \; is \; true \\ 0 \; else \end{matrix} \right\}. \tag{9}$$

So, we can simplify $\prod_{u\in U} P(>u|W,H)$ as the following Formula (10):

$$\prod_{u\in U} P(>_u|W,H) = \prod_{(u,i,j)\in D} P(i>_u j|W,H). \tag{10}$$

To describe the probability of $P(i>_u j|W,H)$, the Bayesian personalized ranking algorithm defines the probability that the user prefers item i compared to item j, as shown in Formula (11):

$$P(i>_u j|W,H) = \sigma(\overline{x}_{uij}(W,H)). \tag{11}$$

It can be decomposed into the following Formula (12):

$$\overline{x}_{uij}(W,H) = \overline{x}_{ui} - \overline{x}_{uj}. \tag{12}$$

For $P(W,H)$, the author assumes that the parameters conform to the normal distribution, and the mean is 0, and the covariance matrix is $\lambda_{(W,H)} I$:

$$P(W,H) \sim N(0, \lambda_{W,H}, I). \tag{13}$$

Finally, we can derive the optimization formula of the model, as shown in Formula (14):

$$\sum_{(u,i,j)\in D} \ln\sigma(\overline{x}_{uij}(W,H)) - \lambda_{W,H}||W,H||^2. \tag{14}$$

IV. Proposed Collaborative Filtering Service Recommendation Algorithm

In the process of creating Mashup, the developer may only describe the functions that Mashup aims to achieve. If the searched service registry has a defined service function category, then the developer can add some related categories as part of the Mashup requirement, the developer sends the function requirement to the service recommendation engine, and the engine returns the ranking result of the service. According to the recommended list, developers select some services that they are interested in to complete the creation process of the Mashup. If the number of selected services in the recommendation list is insufficient, then the developer can send another service recommendation request to the engine. At this point, the service selected by the developer can be included in the new Mashup requirement as an addition. The recommendation process stops when the number of selected services is sufficient or the developer opts out of this process. Therefore, this study summarizes two service recommendation scenarios with different Mashup requirements:

1. Scenario 1: According to the Mashup's functional description requirements (a text description describing the functional requirements of Mashup) and category requirements (the initial category is empty) to recommend suitable services;

2. Scenario 2: According to the function description, some possible categories and some existing services to recommend Mashup demand services.

Algorithm 1 shows a collaborative filtering service recommendation algorithm that integrates heterogeneous information networks and topic models. The algorithm is divided into offline processing module and online service recommendation module. In the offline module, we preprocess the text content of Mashup and Web service, use the LDA topic model to generate topics for Mashup, generate Mashup clusters according to the assigned topics, and then calculate the similarity $\mathrm{PSim}_{\mathrm{P}_l}(\mathrm{m_i},\mathrm{m_j})$ of any two Mashups $\mathrm{m_i}$ and m_j in HIN on a set of meta-path $P1-P6$. After calculating $\mathrm{PSim}_{\mathrm{P}_l}(\mathrm{m_i},\mathrm{m_j})$, we optimize the similarity weight θ_{p_l} on the path based on the Bayesian personalized ranking algorithm and obtain the total similarity value $Sim_P(m_g, m_h)$ between any Mashup m_g and m_h. Then, we preprocess the online Mashup request MRE in the online service recommendation module. Then continue to perform topic inference on the MRE, the k^{cf} Mashup clusters closest to the MRE semantics are determined according to the topic similarity value, and a list of candidate Mashups is generated. Subsequently, the candidate Mashup's ratings are collected for the Web services in the service set, and the collaborative filtering algorithm is used based on the MRE-Mashup similarity to predict the rating of MRE for all Web services. Finally, the top k^{cf} is selected to generate a web service recommendation list according to the ranking. The following paper introduces in detail the related processes of the two modules of the algorithm.

A. Algorithm Offline Processing Module

This section will introduce the offline processing module of the algorithm. Our model includes four main stages, namely preprocessing, Mashup clustering using topic models, Mashup similarity measurement, and weight optimization using BPR.

1) Mashup and Web Service Preprocessing: First, we use the following three steps to preprocess the relevant text content of each Mashup and Web service in the service registry:

1) Word segmentation: Divide the sentence of the text content into individual words;
2) Lexical restoration: convert the deformed word to its basic form. For example, "cluster," "clusters," "clustered" and "clustering" will be converted to "cluster";

Algorithm 1. Proposed collaborative filtering service recommendation algorithm

Input: Text content and labels of Mashups /Web services (M/S), Web Service providers, and Mashup-Service (M-S) composition relationships;

Output: Mashup clustering information,the predictive ratings of Mashup m for Service s;

Preprocess the text content of the Mashup and Web services;
Generate themes for the Mashup and Mashup clusters;
Construct a heterogeneous information network;
Calculate the similarity $\mathrm{PSim}_{\mathrm{P}_l}(\mathrm{m_i},\mathrm{m_j})$ according to Formula (18);
Optimize the similarity weight θ_{p_l} on the path, and then calculate $Sim_P(m_g, m_h)$ according to Formula (22);
Preprocess the online Mashup requirement MRE;
Subject inference on MRE;
Determine k^{clu} Mashup clusters that are closest to the MRE semantics;
Generate a Mashups list;
Predict MRE ratings for all Web services;
Select the top k^{cf} to generate a web service recommendation list.

3) Stop word deletion: delete such as "a," "the," "or," etc. The frequency of use is high but Stop words that do not contribute to the semantics of the text.

2) Use the Topic Model to Cluster Mashup: When a Mashup is developed with a specific function, it enters the Mashup function requirement MFR into the online recommendation system, which contains text content describing the Mashup function. The recommendation system uses collaborative filtering recommendation technology to recommend Web services that satisfies the requirements of the developer. According to the idea that users with similar hobbies have similar ratings for the same item, after matching Mashups similar to MFR, web services that these Mashups interact with are recommended. To reduce the time required to match similar Mashups during online recommendation, the Mashups in the database need to be clustered and grouped in advance, and online recommendation only needs to search for matching objects in the corresponding Mashup cluster.

Prior to the development of a Mashup and publishing on the Internet, the enterprise organization or developer marks the category to which the Mashup belongs. At the same time, the management platform marks the category of the Mashup but causes the phenomenon of Mashup "category explosion." The Programmable Web (PW) website is considered an example. As of December 2018, the categories of Mashup marked on the PW have reached a staggering 460 and still grows continuously. Therefore, using the defined Mashup category is inappropriate for clustering to narrow the search space.

This paper uses the LDA theme model to cluster Mashups to address the "category explosion" problem of Mashups. Clustered Mashups help narrow the search space of the system when recommending online Web services. The LDA topic model is a probabilistic generation model designed to extract text from the corpus. Each text/document in the corpus is modeled as a limited combination of a set of potential topics, and the topic in the corpus text is determined by the distribution probability of words in the text.

When applying the LDA algorithm to the clustering process of the Mashup, the parameters of the topic model should be initially determined, as well as the distribution of topic-word and Mashup-topic. At present, researchers have proposed several methods for estimating model parameter values, such as variational EM algorithm and Gibbs sampling. This paper uses the Gibbs sampling method. Initially, the topic is randomly assigned to all the words in the corpus. The word set is represented by $W = \{w_i\}_{i=1,2,,,N}$, (the Mashup document set, where the word w_i is located is called d_i), and then we continue to resample the topic of each word w_i in W according to Formula (15), as follows:

$$p(z_i = t|z_{-i}, w) \propto \frac{n_{-i,t}^{(w_i)}+\beta}{n_{-i,t}^{(\cdot)}+|V|\beta} \frac{n_{-i,t}^{(d_i)}+}{n_{-i,\cdot}^{(d_i)}+T\alpha}, \tag{15}$$

where $\forall t \in \{1, 2, \ldots, T\}$, z_i indicates the topic assigned to the word w_i, z_{-i} indicates the topic of all words except the word w_i, $n_{-i,t}^{(w_i)}$ refers to the word w_i assigned to the topic t but does not include the current instance statistics; V represents the total number of all words in the word set W, $n_{-i,t}^{(d_i)}$ represents the total number of words assigned to the topic t in the text d_i except w_i; T represents the total number of topics. α and β are the hyperparameters of the model.

After several times of resampling on topics, eliminating the influence of the initial parameters and reaching a stationary state, the probability of each topic in the Mashup document d is expressed as $\hat{\theta}_z^{(d)}$, and the probability of each word in the vocabulary under topic z is $\hat{\phi}_w^{(z)}$. We can estimate them according to Formulas (16) and (17), where ${n_z}^{(d)}$ represents the number of words assigned to the topic in the text, and ${n_z}^{(w)}$ is the number of instances that word assigns to the topic:

$$\hat{\theta}_z^{(d)} = \frac{{n_z}^{(d)} + \alpha}{n^{(d)} + T\alpha}. \tag{16}$$

$$\hat{\phi}_w^{(z)} = \frac{{n_z}^{(w)} + \beta}{n^{(\cdot)} + |V|\beta}. \tag{17}$$

When using LDA to cluster Mashup topics, the number of topics T should be set in advance. Many researchers have conducted studies on the setting of T value. In the work of Griffiths and Steyvers *et al.* [35], as T increases from a small value, the LDA topic model can be more accurately modeled. The topic distribution in the corpus reaches the optimal point, and then the performance of the model is gradually reduced due to the gradual complexity of the model. Bayesian model selection method is used to estimate the value of T that maximizes the posterior probability $p(w|z,T)$ of the LDA model. Blei *et al.* [34] and Liu *et al.* [36] used the "leave out method" to generate test data from training data and performed continuous tests to find the best T value.

Based on the learned topics and Mashup-topic distribution, Mashup is divided into T categories according to the most similar topics, and each topic is assigned to a category. Therefore, the closest topic for each Mashup is determined. However, the problem is that the topic distributions of Mashups are relatively different from each other. Each Mashup is difficult to divide into the closest topics. This paper assigns Mashups to the closest topics k_{clu}; thus, the size of the k_{clu} value directly determines the quality of the model's clustering of the Mashups. The Mashup cannot be easily assigned to the closest topic due to the small k_{clu} value, whereas a large k_{clu} value assigns the Mashup to all of its related topics.

3) Calculate the Similarity on Different Paths: At present, many companies have released a large number of WebAPIs and Mashups on the Internet. Besides, companies have released their related information, including Mashup-Service call relationship, text description (description of function information), category, provider information Wait. This information can be modeled using HIN technology.

According to work done by Xie *et al.* [37], given information about Mashups and Web services, their heterogeneous information network can be defined as $HIN = \{O, R\}$, where $O = <M, S, Co, Ca, P>$ represents the entire set of objects in the heterogeneous information network, and each element in O represents Mashup, service, text content, category, and provider, respectively. $R = <MS, MCo, MCa, SCo, SCa, SP>$ represents the entire set of relationships in a heterogeneous information network. Each element in R is: 1) MS : The relationship between the composition and the composition of Mashups and Web services; 2) MCo: Mashups and their textual content description/described relationship; 3) MCa : Mashups and their category of label/labeled relationship; 4) SC : Web services and their textual content description/ Described relationship; 5) SCa: the label/labeled relationship between Web services and their categories; 6) SP: the provided/provided the relationship between Web services and their providers. Symbols can be used to indicate the relationship between elements in a heterogeneous information network based on Mashup-Service.

The proposed collaborative filtering Web service recommendation method integrating HIN and LDA combines various information of Mashup and Web service, and attributes of both (such as text content, category, and service provider information). In addition, Mashup-service composition relationship is modeled as HIN, and several similarity values between Mashups are measured on a set of meta-paths in HIN. The network model of the heterogeneous information network constructed by Mashup-service (Fig. 2) shows the multiple

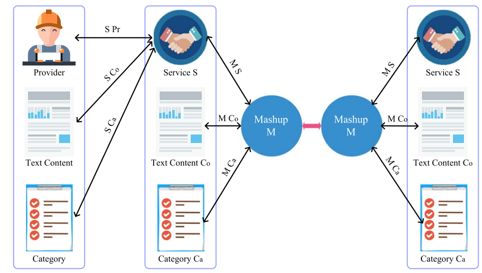


Fig. 2. Details about the HIN modeling information about Mashup and Web Service.

types of paths (called meta-paths) between any two objects. For example, two Mashups can be connected through six meta-paths [31]. Specifically, meta-path P can be formally expressed as $O_1 \xrightarrow{R_1} O_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} O_{l+1}$. Use O to represent the object set, R is used to represent the set of relations, so we have $O_i \in O$, $R_j \in R$. Given a path $p = \{o_1, o_2, \ldots, o_{l+1}\}$, if any i exists such that O_i is the object $e_i = o_i o_{i+1}$ in O and e_i is an element in the relational set, then p is an instance of meta-path P. The semantics of the meta-path P is represented by its relationship chain $R = R_1 \circ R_2 \circ \ldots \circ R_l$. Six different types of semantic similarities between different Mashups are evaluated on the six meta-paths in Table I.

As shown in Table I, after constructing a heterogeneous information network based on Mashup-Service, this paper chooses to calculate the similarity information between Mashups on six meta-paths. Generally speaking, Mashup is composed of several web services. Therefore, each element in the Mashup and Service collection may have the same type or similar text description. Assuming there are Mashup $\mathrm{m_i}$ and $\mathrm{m_j}$, there are six meta-paths between the two, and if one meta-path is selected, there are multiple instances under the path.

$$\mathrm{PSim_P(m_i, m_j)} = \frac{2 \times |p_{m_i \Rightarrow m_j} : p_{m_i \Rightarrow m_j} \in P|}{\kappa}. \tag{18}$$

$$\kappa = |p_{m_i \Rightarrow m_i} : p_{m_i \Rightarrow m_i}| + |p_{m_j \Rightarrow m_j} : p_{m_j \Rightarrow m_j}| \tag{19}$$

The path similarity calculation method between Mashups is shown in Formula 18 and 19, where P is a specified meta path, $\{p_{x \Rightarrow y} \in P\}$ represents the set of instances of Mashup x and y under meta path P:

4) Use Word Embedding Technology to Calculate Similarity: We have obtained a method to calculate the path similarity between Mashups, but because each Mashup or service is unique, the path-based method is unsuitable for measuring the similarity of P_2 and P_5 between Mashups. Scholars have proposed more methods to measure the similarity between content, but they have their limitations. Here, we propose to calculate the semantic similarity between Mashups based on word embedding technology. This neural network-based technology is used to learn the semantic vector representation of words in the text of a large corpus. According to the word vector representation, we use an asymmetric similarity method to calculate the Mashup similarity between Mashup $\mathrm{m_i}$ and $\mathrm{m_j}$ on the P_2 and P_5 paths. The calculation formula is shown in Formula (20), as follows:

TABLE I
SIX DIFFERENT META-PATHS IN MASHUP-SERVICE HIN

ID	Meta path	Semantic information
P1	Mashup - category - Mashups	Two Mashups are marked as belonging to the same category
P2	Mashup-content-Mashup	Both Mashups use the same content to describe their functional requirements
P3	Mashup- Service - Mashup	Both Mashups invoke the same service
P4	Mashup-service - category -Service- Mashup	The two Mashups invoke services with similar categories
P5	Mashup-service-content-service-Mashup	The two Mashups invoke services with similar content
P6	Mashup-service-provider-service-Mashup	Both Mashups invoke services provided by the same provider

$$ESimP_2(m_i, m_j) = \frac{\theta+\delta}{|W(m_i)|+|W(m_j)|}, \tag{20}$$

where $\theta = ESim^{asy}(W(m_i), W(m_j))$ and $\delta = ESim^{asy}(W(m_j), W(m_i))$. $W(m)$ represents the words contained in the preprocessed Mashup text content, and $ESim^{asy}(W_1, W_2)$ calculates the asymmetric similarity between the word set and the Formula (21) Shown:

$$ESim^{asy}(W_1, W_2) = \sum_{w_i \in W_1} \max_{w_j \in W_2} \cdot \eta, \tag{21}$$

where $\eta = \sum_{w_i \in W_1} \frac{E(w_i)E(w_j)}{||E(w_i)|| \cdot ||E(w_j)||}$. As for the calculation of similarity on the P_5 path, since Mashup is composed of multiple Web services, the calculation of the similarity on the path involves all services called by each Mashup of Mashup. Therefore, this paper proposes the text of all services called by Mashup. The content is assembled, replacing the original text content of Mashup to generate new Mashup $\mathrm{m'_i}$ and $\mathrm{m'_j}$, the calculation formula is similar to the above P_2 path.

5) Adjust the Weight of Similarity on Different Meta Paths: The calculation method of the similarity value of two Mashups on different meta-paths is explained. To calculate the total similarity value between Mashups, we perform a weighted summation based on the similarity contribution of different meta-paths. The weight of similarity on different meta-paths provides corresponding assumptions. The Bayesian Personalized Ranking (BPR) algorithm [38] based on Mashup's implicit feedback to the service (such as M-S interactive information) is used.

Specifically, the goal of the BPR algorithm is to select the corresponding Web service $s_i \in S$ from the service set S based on M-S interaction data. Then, it is recommended to Mashup set M in turn according to the rating size, and M is divided into two subsets $MSub_1$ and $MSub_2$ based on whether it has interacted with service s_i. $MSub_1 = \{m_j : (m_j, s_i) \in MS\}$ stands for each Mashup that interacts with the service s_i. In $MSub_2 = M - MSub_1$, the BPR algorithm indicates that each Mashup in $MSub_1$ should be rated higher than all Mashups in $MSub_1$.

To sort Mashups, we first need to estimate the sorting situation of each Mashup m_p in $MSub_1$. Such tasks can be solved by collaborative filtering. Secondly, it is necessary to calculate the total similarity between any Mashup m_q in $MSub_2$. This value is obtained by combining the similarities in the six-element paths according to the corresponding weights. The calculation method is as shown in Formula (22), where P represents all six elements Path, $Sim_{p_i}(m_p, m_q)$ represents the similarity value between Mashup m_p and m_q on the meta path p_i, and θ_i represents the weight coefficient of the meta path p_i.

$$Sim_P(m_p, m_q) = \sum_{p_i \in P} \theta_i \cdot Sim_{p_i}(m_p, m_q), \tag{22}$$

where $SimM_P(m_g, k^{cf})$ containing k^{cf} Mashups similar to p_i is generated based on the total similarity value. The rating calculation method of Mashup m_p for service s_i is shown in Formula (23), where $r(m_q, s_i)$ represents the implicit feedback of m_q to s_i, and this value can be obtained from M-S interaction information. When $(m_q, s_i) \in MS$, $r(m_q, s_i) = 1$, otherwise $r(m_h, s_i) = 0$.

$$\tilde{r}_P = \sum_{m_q \in SimM_P(m_p, k^{cf})} Sim_P(m_p, m_q) \cdot r(m_q, s_i). \tag{23}$$

According to the relevant M-S interactive data, learn similar weights $\theta = \{\theta_1, \theta_2, \ldots \theta_6\}$ on different meta-paths through $p(\theta|MS) \propto p(MS|\theta)p(\theta)$. Under the assumption that all Mashups interacting with the same service are independent, $p(MS|\theta)$ can be represented as shown in Formula (24). Where $m_j \succ_{s_i} m_k$ represents the probability that Mashups that interact with services are rated higher than those that do not interact, which is represented as $\tau(\hat{r}_P(m_j, s_i)) - \hat{r}_P(m_k, s_i))$, and τ represents the Logistic S-type function.

$$p(MS|\theta) = \prod_{s_i \in S} \prod_{m_j \in M_1, m_k \in M_2} p(m_j \succ_{s_i} m_k). \tag{24}$$

$$F = \min_{\theta} -\sum_{s_i \in S} \sum_{\substack{m_j \in M_1 \\ m_k \in M_2}} \ln \tau(\hat{r}_P(m_j, s_i)) - \varphi. \tag{25}$$

The next goal is to maximize the cost function of $p(\theta|MSCM)$. The details of the function are shown in Formula (25), where $\varphi = \min_{\theta} -\sum_{s_i \in S} \sum_{\substack{m_j \in M_1 \\ m_k \in M_2}} \hat{r}_P(m_k, s_i)) + \frac{\lambda}{2}||\theta||^2$, the parameter θ obeys the normal distribution $\theta \sim N(0, \lambda 1)$, $\lambda||\theta||^2/2$ is the L_2 regularization applied to the function parameters to prevent the model from becoming complicated, and the F function is differentiable.

B. Algorithmic Online Service Recommendation Module

After completing the offline module clustering of Mashups, and obtaining the calculation method of similarity between Mashups based on the heterogeneous information network and the BPR sorting algorithm, when the functional requirements of Mashups MRE are sent to the service recommendation engine by the user, the online service recommendation module will Initialize and start working.

1) Subject Inference: Comparing Mashup function requirements MRE with each Mashup consumes considerable computing resources. This paper clusters Mashups to reduce the search space of online service recommendation modules to improve system efficiency. To obtain the topic distribution of MReq and allow the system to locate it in the corresponding Mashup cluster, the "Folding-in" technology [13] is used to randomly assign the topic to each word in the preprocessed MRE text; each word is resampled. After multiple iterations, the topic distribution of MRE is calculated using Formula (16), the k^{clu} topic clusters most related to the MRE topic are selected, and the MRE Mashup search space is limited to the related topic clusters.

2) Introduction of Two Service Recommendation Scenarios: When using collaborative filtering technology for service recommendation, our goal is to find a set of similar Mashups from the narrowed search space. According to this goal, we need to match the Mashup search space, and the candidate Mashup m in $MSP(MRE)$ with the Mashup function requirement MRE.

This paper summarizes two service recommendation scenarios with different Mashup requirements:

1. Scenario 1: According to Mashup's functional description requirements (a text description describing the functional requirements of Mashup) and category requirements (the initial category is empty) to recommend suitable services;
2. Scenario 2: According to the function description, some possible categories and some existing services to recommend Mashup demand services.

According to the two different schemes for web service recommendation, namely, scenario 1 and scenario 2, and their requirements for Mashups are also different. Therefore, we measure the similarity between different types of MRE and m according to the type of Mashup function requirements.

1) If MRE belongs to the situation described in scenario 1 to recommend services, so the proposed algorithm recommends computing path P_1 or the similarity on $\{P_1, P_2\}$. Specifically, we calculate the similarity between the two on the P_2 meta-path based on the learned word vector and the text content of MRE and Mashup. If MRE has category requirements in addition to text content, then we also calculate the similarity between MRE and m on the P_1 meta-path.

2) If $MReq$ belongs to the scenario 2, then we will calculate the similarity between MRE and m on meta-path $\{P_1 - P_6\}$ or $\{P_2 - P_6\}$ respectively.

After measuring the similarity between MRE and m on a series of meta-paths, the total semantic similarity between the two is calculated as shown in Formula (26):

$$Sim_P(MRE, m) = \sum_{p_i \in P} \theta_{P_l} \cdot Sim_{Pl}(MRE, m), \tag{26}$$

$$r(MRE, s_i) = \sum_{m_j \in \left(MRE, k^{cf}\right)} Sim_P\big(MRE, m_j\big)\alpha. \tag{27}$$

where $Sim_{Pl}(MRE, m)$ represents the similarity between MRE and m under the meta-path P_l, θ_{P_l} is the weight value of the meta-path similarity learned according to the BPR algorithm and $\alpha = \sum_{m_j \in (MRE, k^{cf})} r(m_j, s_i)$. Based on the overall similarity value between MRE and m, we obtain k^{cf} Mashups that are closest to each other and express them as Sim_P (MRE, k^{cf}), and then, Mashup requires MRE to score the Web service $s_i \in S$ according to Formula (27) estimates that according to the predicted rating information, the recommendation algorithm generates a corresponding service recommendation list for Mashup requirements.

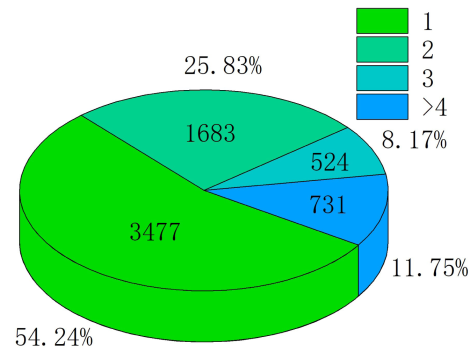


Fig. 3. The number of Mashups containing different component services.

V. Experiment and Evaluation

A. Introduction to Experimental Parameters

The experimental hardware parameters used in this paper are: Intel XeonE5 processor @3.5 GHz, 16GBDDR3 memory, NvidiaQuadroK4000 graphics card, and the operating system is Windows10.

B. Experimental Data Preparation

ProgrammableWeb (referred to as PW) is a world-leading Web API repository. We grab a large number of Web services and Mashup information on the PW server, remove outdated and obsolete Mashups and services, and develop a data set containing 6415 Mashups and 1356 Web services called by Mashups. According to the previous paper, two different service recommendation scenarios, namely, scenario 1 and scenario 2 with different requirements for Mashup to achieve better-recommended performance in a CPSS environment are presented.

1) Experimental Data Statistics: To evaluate the performance of the proposed service recommendation algorithm in two scenarios, we divided the data set into three groups. First, for scenario 1, we randomly select 80% of the data in data set 1 as the test data set and use the remaining 20% as the training data set according to the 8-2 principle of training and test set selection. As shown in Figure 3, for scenario 2, we calculated the number of component services included in each Mashup. About 90% of Mashup calls the number of Web services between 1-3. Therefore, this paper sets up two groups of data according to scenario 2. One group is selected from Mashups with component service of 1, and the other group is selected from Mashups with component service of 2 or more. The

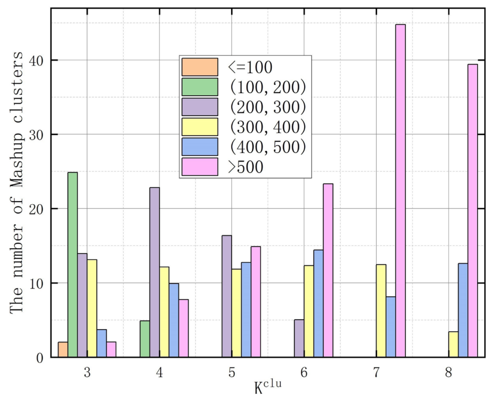


Fig. 4. Mashup cluster size generated by clustering of different k^{clu} values in data set 1 (for scenario 1).

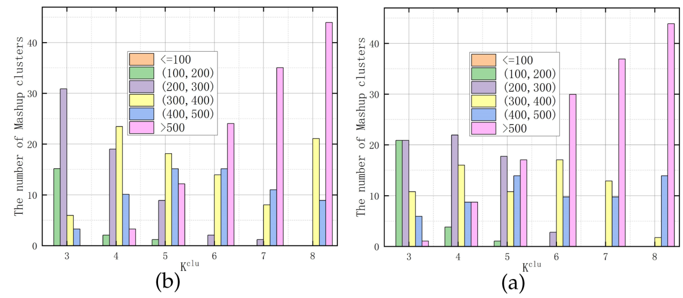


Fig. 5. Mashup cluster size generated by clustering of different k^{clu} values in data set 2 and 3 (for scenario 2).

division of the test data set and training data set of data set 2 and data set 3 follows the 8-2 principle.

2) Offline Processing of Data: This paper uses the LDA topic model to cluster the Mashup, setting the topic model parameter T in order of step size 5 from the interval [10 100]. According to the setting of LDA parameters α and β in literature [38] and [37], this paper sets α and β as 50/T and 0.1 respectively, and iterates 1000 Gibbs samples. Then, the method proposed by Griffiths *et al.* [35] was used to determine the optimal T value, and the optimal T value in the three training sets was 46, 50, and 56 respectively. To evaluate the influence of k^{clu}, this paper set k^{clu} value from 1 to 10 to build different Mashup clusters. Fig. 4 and 5 show the size distribution of the Mashup clusters generated using different k^{clu} values across the three data sets, suggesting that the Mashup cluster grows larger as k^{clu} grows from 0 to 10.

C. Evaluation Criteria

The experimental process uses two popular evaluation criteria, the average precision mean MAP and the normalized discounted cumulative gain NDCG, to evaluate the performance of the recommendation model. The calculation method is shown in Formulas (28) and (29). For the service recommendation algorithm, the overall MAP and NDCG on the test set are calculated based on the average of all Mashup requirements in the test set.

$$MAP = \sum_{i=1}^{N}\left(\frac{N_i}{i}\cdot I(i)\right)\frac{1}{M}, \tag{28}$$

where M represents the set of component services included in the Mashup request $MReq$. N_i represents the number of elements in the M set that appear in the first i places of the recommendation list generated by the recommendation system. Then $I(i)$ indicates whether the i-th service in the recommendation list comes from the M set, with the value of 0 or 1.

$$\text{NDCG} = \frac{1}{|IDCG_N|}\sum_{i=1}^{K}\frac{2^{I(i)}-1}{\log_2(i+1)}, \tag{29}$$

where $IDCG_N$ indicates that the recommendation model is the best score that Mashup can theoretically achieve when requesting $MReq$ to recommend top K services.

D. Comparison Method

We compare the proposed recommendation algorithm with several classic algorithms. These algorithms are as follows:

(1) BPR-SVD: BPR-SVD [38] builds a service recommendation model based on the singular value decomposition algorithm based on Bayesian personalized ranking. The learning scheme of this method is based on stochastic gradient descent and has the function of automatically starting sampling.

(2) PaSRec: PaSRec [23] draws on the idea of a service recommendation model based on collaborative filtering and Bayesian personalized ranking algorithm. A collaborative filtering and meta-path similarity recommendation algorithm is based on the implicit feedback information of Mashup. This Mashup and service based on the three most relevant topics learned by LDA use Formula (20) to calculate the semantic similarity information on the paths P2 and P5. The algorithm uses heterogeneous relationships and implicit feedback data to learn model parameters; it has good recommendation accuracy and recommendation efficiency.

(3) iSRec (TF-IDFCS): iSRec [37] uses TF-IDF technology to represent the collected Mashup and service texts as word vectors, and then calculates the semantic similarity on the P2 and P5 paths according to the cosine similarity method information. This method integrates different information of SBSS and component services to form a heterogeneous information network. Then, the Bayesian personalized ranking algorithm is used to optimize the combined weights of different similarities, and finally obtain a list of recommended services.

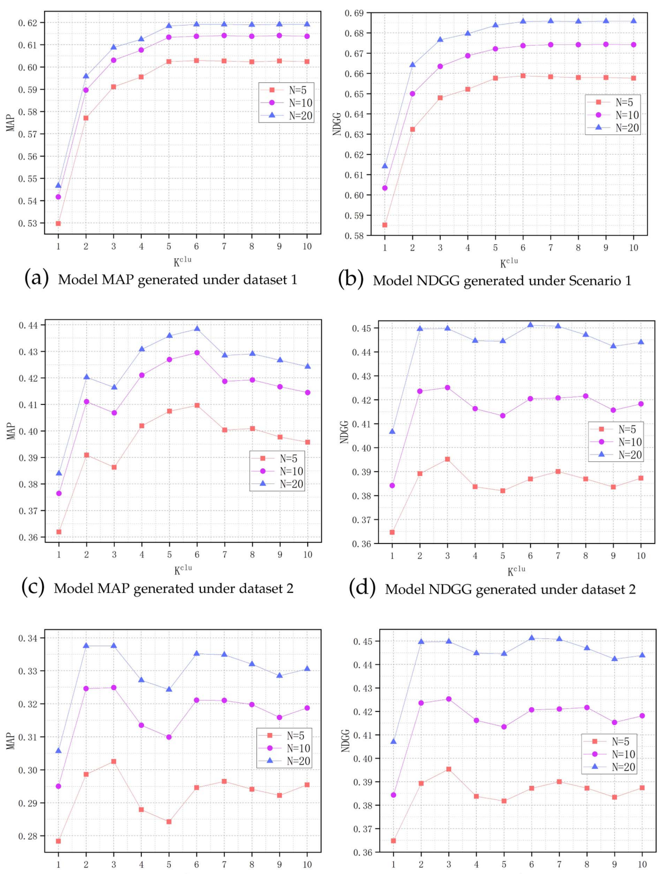
Because of BPR-SVD are not suitable for Mashup requests that do not specify component services, i.e. other services need to be recommended based on existing services. Therefore, the evaluation criteria of these methods are only applicable to the training and test data built-in dataset 2.

E. Analysis of Experimental Results

In the proposed service recommendation algorithm, to improve the recommendation efficiency, the LDA topic model is used to cluster Mashups and reduce the search space for specific Mashup requirements. The k^{clu} parameters are used to

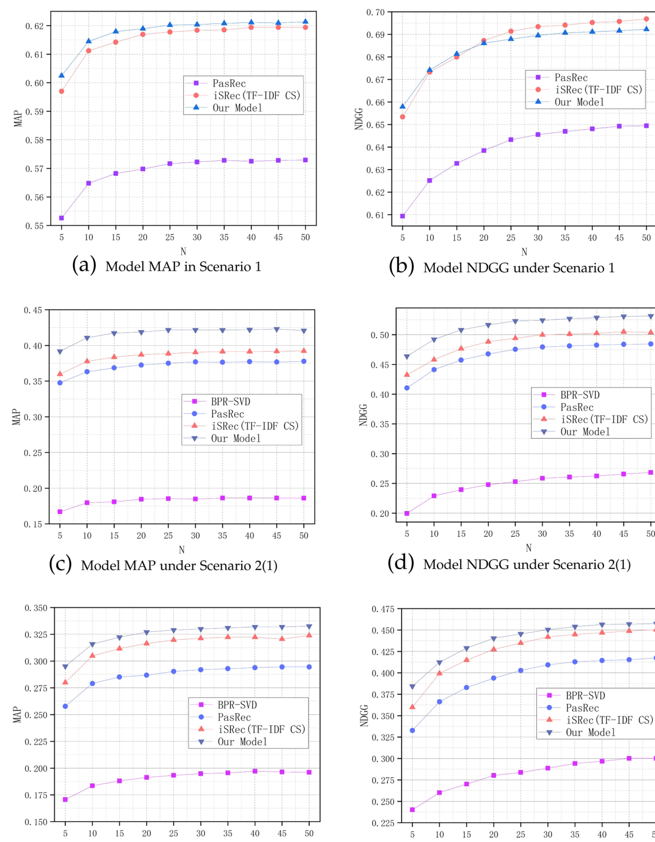
TABLE II
THE EFFECT OF DIFFERENT k^{clu} VALUES ON MODEL PERFORMANCE

	N=5				N=10				N=20			
	kclu=2	kclu=5	kclu=8	kclu=10	kclu=2	kclu=5	kclu=8	kclu=10	kclu=2	kclu=5	kclu=8	kclu=10
Model MAP generated under dataset 1	0.577072981	0.602445652	0.602298137	0.602445652	0.589611801	0.613361801	0.613804348	0.613804348	0.595807453	0.618377329	0.618967391	0.619114907
Model NDGG generated under dataset 1	0.632381306	0.657633531	0.657974777	0.657633531	0.64995549	0.672136499	0.674183976	0.674183976	0.664117211	0.683738872	0.685615727	0.68578635
Model MAP generated under dataset 2	0.390897436	0.407435897	0.400897436	0.395769231	0.411025641	0.426923077	0.419230769	0.414487179	0.42025641	0.435897436	0.429102564	0.424230769
Model NDGG generated under dataset 2	0.389147287	0.382015504	0.386976744	0.387286822	0.423565891	0.413333333	0.421550388	0.418294574	0.449612403	0.444496124	0.447131783	0.444031008
Model MAP generated under dataset 3	0.298617694	0.284241706	0.294099526	0.295434439	0.324597156	0.309913112	0.319770932	0.318744076	0.298617694	0.284241706	0.294099526	0.295434439
Model NDGG generated under dataset 3	0.389319675	0.381774441	0.387252684	0.387442111	0.423583505	0.413402606	0.42167116	0.418140042	0.449630068	0.444565458	0.446942724	0.44387633



(a) Model MAP generated under dataset 1 (b) Model NDGG generated under Scenario 1
(c) Model MAP generated under dataset 2 (d) Model NDGG generated under dataset 2
(e) Model MAP generated under dataset 3 (f) Model NDGG generated under dataset 3

Fig. 6. The effect of different k^{clu} values on performance.



(a) Model MAP in Scenario 1 (b) Model NDGG under Scenario 1
(c) Model MAP under Scenario 2(1) (d) Model NDGG under Scenario 2(1)
(e) Model MAP under Scenario 2(2) (f) Model NDGG under Scenario 2(2)

Fig. 7. Performance Comparison between different methods.

determine the relevant topics of the Mashup. In the experiment, we tested a series of parameter values in the interval [1,10]. Table II and Fig. 6 show the model performance results on the three test sets under different k^{clu} values (i.e., overall MAP@N and NDCG@N). The experimental results show that, as the value of B increases from 1 to 10, the performance of each evaluation standard roughly goes through the following process: it increases rapidly at the beginning and decreases slightly to a stable value after reaching the optimal point. The main reason for the lower performance when the k^{clu} is small is that some similar Mashups that satisfy the functional requirements of Mashups are excluded in the most similar clusters of the k^{clu}, resulting in the model not being able to recommend appropriate related web services. The reason for the decrease in model performance caused by using a larger k^{clu} is that the model considers more candidate Mashups, and some inappropriate Mashups will be recommended. At the same time, to verify the performance of the Web service recommendation algorithm proposed in this paper, we constructed three data sets based on the Mashup and Web service information captured on Programmable Web and conducted a series of experiments on these data sets.

Fig. 7 shows the performance of these service recommendation methods on the three test sets of scenarios 1, 2 (1), and 2 (2). Under Data Set 1, iSRec (TF-IDFCS) outperforms PaSRec on both evaluation indicators. This finding shows the two types of methods for measuring functional similarity between Mashups, as follows: the similarity measurement method based on the proposed word embedding and the vector cosine similarity measurement method based on TFIDF are better than the path-based similarity measurement method used in PaSRec. In addition, in most cases, the proposed method is more effective than the other three methods. Our analysis shows that high-quality word vectors learned using word vector technology help to better measure the functional similarity between Mashups.

Under datasets 2 and 3, the performance improvement effect of the proposed service recommendation algorithm is evident, and the performance of PaSRec is always the lowest. These results show that the path-based functional similarity measurement method has the worst performance in measuring the functional similarity of Mashup. In addition, several methods of constructing HIN based on the information of Mashup and service are much better than the other classic methods that rely solely on the Mashup-service invocation relationship.

VI. Conclusions

CPSS is a rapidly evolving and widely accepted example. However, the data in CPSS have problems, such as data redundancy and data sparseness. To improve the processing ability of CPSS data and improve the accuracy of service recommendations, a collaborative filtering service recommendation algorithm that integrates heterogeneous information networks and topic models are proposed. The algorithm is divided into offline processing module and online service recommendation module. The offline processing module is responsible for preprocessing the Mashup and Web service information in the database. The online service recommendation module is responsible for performing topic inference on the text content of Mashup and then associating it with the nearest Mashup cluster. Then, collaborative filtering technology is used to predict the level of Mashup in the database, and different service recommendations for different services are determined. The experimental results show that compared with mainstream service recommendation algorithms, the proposed service recommendation algorithm has a significant performance improvement. Applying the proposed recommendation algorithm to the CPSS environment can effectively solve the problem of data sparseness in the CPSS system, improve the quality of enterprise service recommendations, and provide users with a better service experience. In future work, we consider applying more data and information to the process of service recommendation to improve the real-time and efficiency of recommendation.

References

[1] B. A. Yilma, H. Panetto, and Y. Naudet, "Systemic formalisation of cyber-physical-social system (CPSS): A systematic literature review," *Comput. Ind.*, vol. 129, 2021, Art. no. 103458.

[2] K. Gai, M. Qiu, H. Zhao, and X. Sun, "Resource management in sustainable cyber-physical systems using heterogeneous cloud computing," *IEEE Trans. Sustain. Comput.*, vol. 3, no. 2, pp. 60–72, Apr. 2018.

[3] M. Zhu *et al.*, "Public vehicles for future urban transportation," *IEEE Trans. ITS*, vol. 17, no. 12, pp. 3344–3353, Dec. 2016.

[4] L. Tao, S. Golikov, K. Gai, and M. Qiu, "A reusable software component for integrated syntax and semantic validation for services computing," in *Proc. IEEE Symp. Service-Oriented System Eng.*, 2015, pp. 127–132.

[5] J. Niu, C. Liu, Y. Gao, and M. Qiu, "Energy efficient task assignment with guaranteed probability satisfying timing constraints for embedded systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2043–2052, Aug. 2014.

[6] W. Liang, S. Xie, D. Zhang, X. Li, and K. Li, "A mutual security authentication method for rfid-puf circuit based on deep learning," *ACM Trans. Internet Technol.*, pp. 1–20, 2020.

[7] Y. Yin, F. Yu, Y. Xu, L. Yu, and J. Mu, "Network location-aware service recommendation with random walk in cyber-physical systems," *Sensors*, vol. 17, no. 9, p. 2059, 2017.

[8] W. Dai, M. Qiu, L. Qiu, L. Chen, and A. Wu, "Who moved my data? privacy protection in smartphones," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 20–25, Jan. 2017.

[9] W. Liang, D. Zhang, X. Lei, M. Tang, K.-C. Li, and A. Zomaya, "Circuit copyright blockchain: Blockchain-based homomorphic encryption for ip circuit protection," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1410–1420, Jul.–Sep. 2022.

[10] X. Li, T. Liu, M. S. Obaidat, F. Wu, P. Vijayakumar, and N. Kumar, "A lightweight privacy-preserving authentication protocol for vanets," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3547–3557, Sep. 2020.

[11] R. Perrey and M. Lycett, "Service-oriented architecture," in *Proc. Symp. Appl. Internet Workshops*, 2003, pp. 116–119.

[12] D. Benslimane, S. Dustdar, and A. Sheth, "Services mashups: The new generation of web applications," *IEEE Internet Comput.*, vol. 12, no. 5, pp. 13–15, Sep.-Oct. 2008.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[14] M. Qiu, K. Zhang, and M. Huang, "An empirical study of web interface design on small display devices," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, 2004, pp. 29–35.

[15] W. Liang, L. Xiao, K. Zhang, M. Tang, D. He, and K. C. Li, "Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14741–14751, Aug. 2022.

[16] K. Gai and M. Qiu, "Optimal resource allocation using reinforcement learning for iot content-centric services," *Appl. Soft Comput.*, vol. 70, pp. 12–21, 2018.

[17] K. Gai and M. Qiu, "Reinforcement learning-based content-centric services in mobile sensing," *IEEE Netw.*, vol. 32, no. 4, pp. 34–39, 2018.

[18] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative web service qos prediction via neighborhood integrated matrix factorization," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 289–299, Jul./Aug. 2012.

[19] X. Liu, S. Agarwal, C. Ding, and Q. Yu, "An LDA-SVM active learning framework for web service classification," in *Proc. IEEE Int. Conf. Web Serv.*, 2016, pp. 49–56.

[20] Y. Zhang, "GroRec: A group-centric intelligent recommender system integrating social, mobile and big data technologies," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 786–795, Sep. 2016.

[21] C. H. Liu, Z. Zhang, and M. Chen, "Personalized multimedia recommendations for cloud-integrated cyber-physical systems," *IEEE Syst. J.*, vol. 11, no. 1, pp. 106–117, Mar. 2017.

[22] A. Jain, X. Liu, and Q. Yu, "Aggregating functionality, use history, and popularity of APIs to recommend mashup creation," in *Proc. Int. Conf. Service-Oriented Comput.* Springer, 2015, pp. 188–202.

[23] T. Liang, L. Chen, J. Wu, H. Dong, and A. Bouguettaya, "Meta-path based service recommendation in heterogeneous information networks," in *Proc. Int. Conf. Service-Oriented Comput.* Springer, 2016, pp. 371–386.

[24] P. Samanta and X. Liu, "Recommending services for new mashups through service factors and top-k neighbors," in *Proc. IEEE Int. Conf. Web Serv.*, 2017, pp. 381–388.

[25] C. Chen, X. Meng, Z. Xu, and T. Lukasiewicz, "Location-aware personalized news recommendation with deep semantic analysis," *IEEE Access*, vol. 5, pp. 1624–1638, 2017.

[26] B. Bai, Y. Fan, W. Tan, and J. Zhang, "DLTSR: A deep learning framework for recommendation of long-tail web services," *IEEE Trans. Services Comput.*, vol. 13, no. 1, pp. 73–85, Jan.-Feb. 2020.

[27] F. Xie, L. Chen, Y. Ye, Y. Liu, Z. Zheng, and X. Lin, "A weighted meta-graph based approach for mobile application recommendation on heterogeneous information networks," in *Proc. Int. Conf. Service-Oriented Comput.* Springer, 2018, pp. 404–420.

[28] W. Liang, J. Long, K.-C. Li, J. Xu, N. Ma, and X. Lei, "A fast defogging image recognition algorithm based on bilateral hybrid filtering," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 17, no. 2, pp. 1–16, Apr. 2021.

[29] C. Lei, H. Dai, Z. Yu, and R. Li, "A service recommendation algorithm with the transfer learning based matrix factorization to improve cloud security," *Inf. Sci.*, vol. 513, pp. 98–111, 2020.

[30] F. Yang, H. Wang, and J. Fu, "Improvement of recommendation algorithm based on collaborative deep learning and its parallelization on spark," *J. Parallel Distrib. Comput.*, vol. 148, pp. 58–68, 2021.

[31] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 17–37, Jan. 2017.

[32] W. Dai, L. Qiu, A. Wu, and M. Qiu, "Cloud infrastructure resource allocation for big data applications," *IEEE Trans. Big Data*, vol. 4, no. 3, pp. 313–324, Sep. 2018.

[33] W. Liang, Y. Fan, K.-C. Li, D. Zhang, and J.-L. Gaudiot, "Secure data storage and recovery in industrial blockchain network environments," *IEEE Trans. Ind. Inform.*, vol. 16, no. 10, pp. 6543–6552, Oct. 2020.

[34] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.
[35] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.
[36] J. Liu, M. Tang, Z. Zheng, X. Liu, and S. Lyu, "Location-aware and personalized collaborative filtering for web service recommendation," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 686–699, Sep./Oct. 2016.
[37] F. Xie, J. Wang, R. Xiong, N. Zhang, Y. Ma, and K. He, "An integrated service recommendation approach for service-based system development," *Expert Syst. Appl.*, vol. 123, pp. 178–194, 2019.
[38] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*.

Xiaoyan Chen received the M.S. degree from the Department of Software Engineering, Xiamen University, Xiamen, China, in 2012. She is currently a Lecturer with the the School of Software Engineering, Xiamen University of Technology, Xiamen, China. Her research interests include blockchain security technology, big data analysis, networks security protection, embedded system and hardware IP protection, and security management in wireless sensor networks (WSN).

Wei Liang received the Ph.D. degree in computer science and technology from Hunan University, Changsha, China, in 2013. During 2014–2016, he was a Postdoctoral Scholar with Lehigh University, Bethlehem, PA, USA. He is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University. He has authored or coauthored more than 110 journal or conference papers, such as the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, and IEEE INTERNET OF THINGS JOURNAL. His research interests include blockchain security technology, networks security protection, embedded system and hardware IP protection, fog computing, and security management in wireless sensor networks (WSN). He was an Application Track Chair of IEEE Trustcom 2015, a Workshop Chair of IEEE Trustcom WSN 2015, and IEEE Trustcom WSN 2016.

Jianbo Xu received the M.S. degree from the Department of Computer Science and Technology, National University of Defense Technology, Changsha, China, in 1994 and the Ph.D. degree from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2003. Since 2003, he has been a Professor with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China. His research interests include network security and distributed computing.

Chong Wang is currently working toward the Graduation degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include blockchain security technology, networks security protection, embedded system and hardware IP protection, fog computing, and security management in wireless sensor networks (WSN).

Kuan-Ching Li (Senior Member, IEEE) received the Licenciatura degree in mathematics, and the M.S. and Ph.D. degrees in electrical engineering from the University of Sao Paulo, Brazil, in 1994, 1996, and 2001, respectively. He is currently a Distinguished Professor with the Department of Computer Science and Information Engineering, Providence University, Taichung, Taiwan, where he is also the Director of the High-Performance Computing and Networking Center. He has authored or coauthored more than 200 scientific papers and papers and is the Coauthor or Co-editor of more than 25 books published by Taylor & Francis, Springer, and McGraw-Hill. His research interests include parallel and distributed computing, big data, and emerging technologies. He is the Editor-in-Chief of the journal *Connection Science* and is an Associate Editor for several leading journals. He is also actively involved in many main conferences and workshops in program/general/steering conference chairman positions and has organized numerous conferences related to computational science and engineering. He is a Fellow of IET.

Meikang Qiu (Senior Member, IEEE) received the B.E. and M.E. degrees from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree in computer science from the University of Texas at Dallas, Richardson, TX, USA. He is the Department Head and a tenured Full Professor with Texas A&M University-Commerce, Commerce, TX, USA. He has published more than 20 books, more than 600 peer-reviewed journal and conference papers, including more than 100 IEEE/ACM Transactions papers. His research interests include cyber security, big data analysis, cloud computing, smarting computing, intelligent data, and embedded systems. Since 2019, he has been an ACM Distinguished Member. He is the Chair of IEEE Smart Computing Technical Committee. He is an Associate Editor of more than ten international journals, including the IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON BIG DATA, and IEEE Transactions on SMC (A: System). He is a Highly Cited Researcher (2020, Web of Science) and an IEEE Distinguished Visitor (2021–2023). He was the recipient of the IEEE Big Data Security Pioneer Award 2021.