Expert Systems With Applications 110 (2018) 191–205

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Deep hybrid collaborative filtering for Web service recommendation

Ruibin Xiong, Jian Wang*, Neng Zhang, Yutao Ma

School of Computer Science, Wuhan University, Wuhan, China

ARTICLE INFO

Article history:
Received 27 September 2017
Revised 29 May 2018
Accepted 30 May 2018
Available online 5 June 2018

Keywords:
Web service recommendation
Mashup
Collaborative filtering
Deep learning

ABSTRACT

With the rapid development of service-oriented computing and cloud computing, an increasing number of Web services have been published on the Internet, which makes it difficult to select relevant Web services manually to satisfy complex user requirements. Many machine learning methods, especially matrix factorization based collaborative filtering models, have been widely employed in Web service recommendation. However, as a linear model of latent factors, matrix factorization is challenging to capture complex interactions between Web applications (or mashups) and their component services within an extremely sparse interaction matrix, which will result in poor service recommendation performance. Towards this problem, in this paper, we propose a novel deep learning based hybrid approach for Web service recommendation by combining collaborative filtering and textual content. The invocation interactions between mashups and services as well as their functionalities are seamlessly integrated into a deep neural network, which can be used to characterize the complex relations between mashups and services. Experiments conducted on a real-world Web service dataset demonstrate that our approach can achieve better recommendation performance than several state-of-the-art methods, which indicates the effectiveness of our proposed approach in service recommendation.

1. Introduction

Service-oriented computing (SOC) has significantly affected software development by utilizing services as fundamental building blocks in constructing low-cost and reliable software applications. With the rapid evolution of SOC and cloud computing, an increasing number of Web services (mainly in the form of RESTful Web APIs) have been published on the Internet. For example, over 16,000 Web services have been published at Programmableweb[1] (PW) by January 1, 2017, almost increased to three times as compared with three years ago. Many Web API marketplaces founded by famous IT companies like Amazon and Microsoft have also published a plenty of Web APIs (we use the two terms, Web service and Web API, interchangeably throughout the paper). Since most user requirements cannot be satisfied by a single Web service, it is necessary to compose existing Web services to offer value-added services (also known as service composition or mashups) for users. However, the overwhelming number of Web services makes it difficult to select relevant Web services manually to satisfy complex user requirements. Therefore, it is vital to proactively and accurately discover suitable Web services according to user requests.

Web service recommendation refers to the process of proactively discovering relevant Web services that can meet user requests. Currently, matrix factorization based collaborative filtering models have been widely employed in Web service recommendation (Jain, Liu, & Yu, 2015; Liu, Tang, Zheng, Liu, & Lyu, 2016; Samanta & Liu, 2017; Tian, Wang, He, Sun, & Tian, 2017; Zheng, Ma, Lyu, & King, 2013), which can recommend Web services for mashup construction or predict service qualities by leveraging existing usage histories. However, matrix factorization is deemed as a linear model of latent factors and is thus difficult to capture complex interactions between users and items when the interaction matrix is highly sparse (He et al., 2017). According to the statistics of PW, the largest Web service registry, the sparsity of the mashup-service invocation matrix is about 99.83%, which is extremely sparse. How to accurately characterize the complex relations between mashups and services within an extremely sparse matrix becomes an intractable issue.

Recently, deep learning methods have been successfully applied in recommender systems (Zhang, Yao, & Sun, 2017), due to their powerful representation learning abilities. They can be used to learn hidden structures from the interactions of users and items. Inspired by the idea of deep learning based collaborative filter-

* Corresponding author.
E-mail addresses: ruibinxiong@whu.edu.cn (R. Xiong), jianwang@whu.edu.cn (J. Wang), nengzhang@whu.edu.cn (N. Zhang), ytma@whu.edu.cn (Y. Ma).

[1] http://www.programmableweb.com/.

https://doi.org/10.1016/j.eswa.2018.05.039
0957-4174/

ing technologies (Cheng et al., 2016; Guo, Tang, Ye, Li, & He, 2017; He et al., 2017; Paradarami, Bastian, & Wightman, 2017; Xue, Dai, Zhang, Huang, & Chen, 2017), in this paper, we propose a novel deep hybrid collaborative filtering approach for service recommendation (referred to as DHSR) to capture the complex invocation relations between mashups and services. Since the textual contents including descriptions and tags of services and mashups are also crucial in service recommendation, DHSR further integrates collaborative filtering with textual content within a deep neural network. The main contribution of our work is summarized as follows:

- We propose a novel deep learning based hybrid approach that combines collaborative filtering and textual content. The invocation interactions between mashups and services as well as their textual functionalities are seamlessly integrated into a deep neural network, which can be used to characterize complex relations between mashups and services within an extremely sparse interaction matrix.
- We conduct a series of experiments using real-world Web services crawled from PW to evaluate the proposed approach. Experimental results demonstrate that our approach can achieve better recommendation performance than several state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 formulates the problem of service recommendation for mashup development. Section 4 introduces the details of the proposed approach, and Section 5 presents the experimental results and analysis. Finally, Section 6 summarizes the paper and puts forward our future work.

2. Related work

As one of the fundamental research issues in the field of SOC, Web service recommendation has been widely investigated. The studies of this area can fall within the scope of three categories: functionality-based Web service recommendation, social network-based Web service recommendation, and collaborative filtering-based Web service recommendation.

(1) Functionality-based Web service recommendation

Functionality-based Web service recommendation refers to recommending services by matching user requests with service descriptions. Earlier studies that use keyword-based service profile matching usually suffer from poor retrieval performance; therefore, many explicit semantics based approaches had been proposed to improve the performance of service matching. These approaches (Paliwal, Shafiq, Vaidya, Xiong, & Adam, 2012; Rodriguez-Mier, Pedrinaci, Lama, & Mucientes, 2016; Roman, Kopecký, Vitvar, Domingue, & Fensel, 2015) leveraged domain ontologies or dictionaries to enrich semantics of descriptions of both services and user requests, and adopted logic-based reasoning for semantic similarity calculation; however, they are limited by manually defining ontologies and semantically annotating descriptions, which make it difficult to be applied to large scale service data.

Besides these explicit semantics-based approaches, many other efforts integrate functionality based service recommendation with machine learning or data mining technologies. Meng, Dou, Zhang, and Chen (2014) used keywords to indicate user preferences and recommended services according to their semantic compatibility with user preferences. Zhang, Wang, and Ma (2017) proposed to extract domain service goals from textual descriptions to meet users' intentional requests. Yao, Wang, Sheng, Ruan, and Zhang (2015) presented an approach to service recommendation based on services' functional features and the co-invocation among services.

(2) Social network-based Web service recommendation

Social network-based Web service recommendation refers to utilizing social network relationships of developers or services in Web service recommendation. For example, Cao, Liu, Tang, Zheng, and Wang (2013) integrated user interests and social relations in recommending services for mashup development. Chen, Paik, and Hung (2015) designed a social network for service recommendation by combining multiple relations among users, services, and topics. Xu, Cao, Hu, Wang, and Li (2013) constructed a global social service network based on complex networks and proposed a service discovery approach based on the service network. Gao, Chen, Wu, and Bouguettaya (2016) presented a service recommendation method by modeling users' historical preferences, functionalities of services and mashups, as well as invocation relations between mashups and services. Liang, Chen, Wu, Dong, and Bouguettaya (2016) adopted heterogeneous information network to describe heterogeneous objects including mashups, services, tags, and providers, as well as their relations and further proposed a meta-path based Web service recommendation method. Their approach comprehensively analyzed and integrated multiple factors that may contribute to the invocation relations between mashups and services, and can thus achieve high recommendation performance.

(3) Collaborative filtering (CF)-based Web service recommendation

CF-based Web service recommendation refers to recommending services according to the past composition history, the similarity of users, or the similarity of services. They are firstly used in quality of service (QoS) prediction, which can be used to select high-quality services in Web service recommendation. For example, Zheng et al. (2013) proposed an approach to predict missing QoS information by using neighborhood integrated matrix factorization. Liu et al. (2016) also presented a location-aware CF method for QoS-aware Web service recommendation. Tian et al. (2017) proposed a time-aware CF algorithm based on implicit feedback for Web service recommendation, where three kinds of time effects including user bias shifting, Web service bias shifting, and user preference shifting, are integrated into a latent factor model.

Recently, many hybrid approaches have been proposed to recommend services by incorporating multiple factors such as service invocation history and functionalities. For example, Yao, Sheng, Segev, and Yu (2013) proposed a hybrid approach by combining CF and content based recommendation, which can dynamically recommend Web services that fit users' interests. Jain et al. (2015) incorporated three factors into the service recommendation process: APIs' functionalities, usage history of APIs by existing mashups, and popularities of APIs. They leveraged probabilistic topic models, matrix factorization based collaborative filtering, and Bayes' theorem to recommend APIs for mashup creation. In their latest work (Samanta & Liu, 2017), they further used the Hierarchical Dirichlet Process (HDP) to discover functionally relevant services based on their specifications, and leveraged Probabilistic Matrix Factorization (PMF) to recommend services based on usage history and tackle the cold start problem for new mashups through their closest neighbors.

Unlike existing studies, we proposed a hybrid Web service recommendation approach by incorporating CF and textual content within a deep neural network. Our approach is a combination of functionality-based and CF-based Web service recommendation approach. A deep learning based recommendation approach for long-tail Web services was proposed recently (Bai, Fan, Tan, & Zhang, 2017), which exhibited the advantages of applying deep learning technologies in this field. They leveraged the deep learning model SDAE (stacked denoising autoencoders) and time information to learn feature representations. Their work is mainly a