

### 基于多目特征交叉的服务推荐算法

高文斌, 王睿, 祖家琛, 董晨辰, 胡谷雨

引用本文

高文斌, 王睿, 祖家琛, 董晨辰, 胡谷雨 [基于多目特征交叉的服务推荐算法](#) [J]. 计算机科学, 2023, 50(6A): 210800242-7.

GAO Wenbin, WANG Rui, ZU Jiachen, DONG Chenchen, HU Guyu. [Service Recommendation Algorithm Based on Multi-features Crossing](#) [J]. Computer Science, 2023, 50(6A): 210800242-7.

---

### 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于Apriori算法填充数据及改进相似度的推荐算法](#)

Recommendation Algorithm Based on Apriori Algorithm and Improved Similarity

计算机科学, 2022, 49(11A): 211000005-5. <https://doi.org/10.11896/jsjcx.211000005>

[复杂网络社团发现综述](#)

Survey of Community Detection in Complex Network

计算机科学, 2022, 49(11A): 210800144-11. <https://doi.org/10.11896/jsjcx.210800144>

[基于评论方面级用户偏好迁移的跨领域推荐算法](#)

Cross-domain Recommendation Based on Review Aspect-level User Preference Transfer

计算机科学, 2022, 49(9): 41-47. <https://doi.org/10.11896/jsjcx.220200131>

[基于用户场景的Android 应用服务推荐方法](#)

Recommendation of Android Application Services via User Scenarios

计算机科学, 2022, 49(6A): 267-271. <https://doi.org/10.11896/jsjcx.210700123>

[视频缓存策略中QoE和能量效率的公平联合优化](#)

Fair Joint Optimization of QoE and Energy Efficiency in Caching Strategy for Videos

计算机科学, 2022, 49(4): 312-320. <https://doi.org/10.11896/jsjcx.210800027>

# 基于多目特征交叉的服务推荐算法

高文斌<sup>1</sup> 王睿<sup>1</sup> 祖家琛<sup>1</sup> 董晨辰<sup>2</sup> 胡谷雨<sup>1</sup>

<sup>1</sup> 陆军工程大学指挥控制工程学院 南京 210007

<sup>2</sup> 蚌埠学院计算机与信息工程学院 安徽 蚌埠 233000

(wenbinnj@qq.com)

**摘要** 随着 Web 服务数量的迅速增长,服务过载的问题逐步显现。为了解决服务过载的问题,帮助用户快速定位高质量服务,服务推荐成为了服务计算领域的研究热点。针对目前服务推荐中冷启动及数据稀疏的难点问题,提出了一种基于多目特征交叉的服务质量(Quality of Service, QoS)预测推荐算法(Service Recommendation Algorithm Based on Multi-features Crossing, SRMFC),通过“词嵌入”方法实现多目特征的引入,提升算法在应对冷启动时的表现;同时,应用神经网络完成多目特征的自动交叉,相比于传统协同过滤(Collaborative Filtering, CF)、因子分解机(Factorization Machine, FM)等方法,该算法能实现特征之间相互关系的深入挖掘,从而提升算法在应对数据极度稀疏场景下的学习能力。在公共数据集上的实验结果表明,基于多目特征交叉的服务推荐算法在不同数据稀疏性场景下,相比于近几年主流的服务推荐算法,服务质量预测误差至少降低 20%。

**关键词:** 服务过载;冷启动;数据稀疏;QoS 预测;服务推荐

**中图法分类号** TP301

## Service Recommendation Algorithm Based on Multi-features Crossing

GAO Wenbin<sup>1</sup>, WANG Rui<sup>1</sup>, ZU Jiachen<sup>1</sup>, DONG Chenchen<sup>2</sup> and HU Guyu<sup>1</sup>

<sup>1</sup> School of Command and Control Engineering, Army Engineering University of PLA, Nanjing 210007, China

<sup>2</sup> School of Computer Science and Information Engineering, Bengbu University, Bengbu, Anhui 233000, China

**Abstract** With the rapid growth of the number of web services, the problem of service overload has gradually emerged. To relieve service overload, and help users position high-quality services rapidly, service recommendation has become a hot research topic in the field of service computing. Aiming at the difficulties of cold start and data sparseness in current service recommendation, this paper proposes a quality of service(QoS) prediction recommendation algorithm SRMFC based on the multi-features crossing, which implements multi-features through the “word embedding” method to improve the performance of the algorithm in dealing with the cold start. At the same time, a neural network is used to complete the automatic cross of multi-features. Compared with traditional collaborative filtering, factorization machine and other methods, the proposed algorithm can achieve in-depth exploration of the relationship between features, and improve the learning ability of the algorithm in dealing with extremely sparse data scenarios. Experiments on public data sets show that, under different data sparsity scenarios, the service quality prediction error of the SRMFC intersection decrease by at least 20% compared with the mainstream service recommendation algorithm in recent years.

**Keywords** Service overload, Cold start, Data sparseness, QoS prediction, Service recommendation

## 1 引言

随着服务化理念的推广和面向服务的计算模式(SOC)的广泛应用,越来越多的企业选择为用户提供服务化的产品。作为面向服务的架构(SOA)最主流的实现方式,Web 服务(Web Service, WS)的数量呈指数级增长。截至 2021 年 6 月 29 日,programableweb<sup>1)</sup>上发布的 Web 服务数量已经达到 22200 个。服务发现效率越来越低下,服务过载问题逐步显现。服务发现发生在服务消费者和服务注册中心之间,指服务消费者向服务注册中心发送携带服务功能性需求的服务调用请求,服务注册中心通过检索服务数据库,为服务消费者

匹配满足功能性需求的服务的过程;而服务过载指通过常规的服务发现手段,已经难以实现对服务有效的处理和利用了。如何解决服务过载问题成为了目前服务计算领域的一个研究热点。

解决服务过载问题需要解决两个问题:首先,对于相同功能性的服务,如何区分服务的优劣;其次,如何提高服务发现的效率,从而帮助用户快速定位高质量服务。产生以上问题的原因主要有以下 3 点:1)服务质量与其多个相关指标之间的关系还没有统一的标准定义;2)真实的 Web 服务质量评估需要先调用 Web 服务,这是一项耗时且耗费资源的工作;3)由于对用户环境的高度依赖<sup>[1]</sup>,两个不同的 Web

基金项目:国家自然科学基金(62076251)

This work was supported by the National Natural Science Foundation of China(62076251).

通信作者:胡谷雨(hugugu@189.com)

<sup>1)</sup> www.programableweb.com

服务的质量可能会有很大差异。

针对如何区分服务优劣的问题,学术界的做法是通过服务的非功能性属性来区分服务。服务的非功能性属性包括响应时间(RespondingTime, RT)、吞吐量(Throughput, TP)、价格(Price, Pr)、可靠性(Reliability, Re)等<sup>[2]</sup>。在提高服务发现效率方面,各方学者都做了深入研究。尤其在推荐系统成为学术界关注的热点以后,结合推荐系统的服务发现成了服务计算领域的一个研究热点。协同过滤算法<sup>[3-9]</sup>是最早应用到服务推荐中的算法,在初期取得了一定成效。但由于该算法只使用用户 ID 信息和服务 ID 信息,随着用户数量、服务数量的增加,传统协同过滤算法在应对数据极度稀疏以及用户冷启动问题上的效果越来越差。针对这一问题,一方面,部分专家学者转而从服务上下文信息入手,将更多的用户及服务的周边信息引入服务推荐系统中,包括用户位置信息以及服务位置信息等,帮助新用户的历史数据稀疏的情况下,完成高质量推荐。另一方面,部分专家学者从算法的角度入手,提出了一系列复杂度更高、特征学习能力更强的算法。包括基于矩阵分解的推荐算法<sup>[10-11]</sup>、基于因子分解机的算法<sup>[12]</sup>等。

本文针对服务过载面临的两方面问题,提出了一种基于多目特征交叉的服务推荐算法 SRMFC 算法。引入用户位置信息、服务位置信息等服务上下文信息以提高推荐系统在冷启动上的性能;同时,引入深度交叉网络 Deep Crossing,以解决多目特征之间的高维交叉问题,使得多个特征之间能实现更高阶的交叉学习,提高在大规模推荐系统中,数据稀疏情况下的推荐精度。实验证明 SRMFC 算法的推荐效果比现有基于协同过滤及因子分解等算法在 QoS 误差上至少降低了 20%。本文的主要贡献如下:

(1)与传统基于聚类位置感知服务推荐系统不同,将用户经度、用户纬度、服务经度、服务纬度 4 个服务调用过程中的上下文信息通过 Embedding 嵌入服务推荐算法中,减少了聚类的过程,且避免了基于聚类的推荐算法中需要人为确定聚类数目的问题,提升了服务推荐系统在应对冷启动问题方面的性能及算法的自适应性。

(2)在引入用户 ID、服务 ID 以及位置信息等多个特征以后,将 Deep Crossing 方法适配入服务推荐中,在 Deep Crossing 方法的基础上,改变了原有 Deep Crossing 网络部分,降低了网络的复杂度,并在服务的 QoS 预测上获得了更高的精度。相比于目前服务推荐中使用的协同过滤算法、因子分解机算法,Deep Crossing 在特征交叉上的效果更好,很好地解决了因子分解机中高阶特征交叉导致的参数爆炸问题。

(3)在真实 Web 服务 QoS 数据集 WSdream<sup>[13]</sup>上验证了 SRMFC 算法的性能,实验结果表明,本文算法在冷启动场景及数据高度稀疏的情况下,QoS 预测有较高的准确率。

## 2 相关工作

基于用户、服务的协同过滤是服务推荐中最早使用的推荐算法。Xu 等<sup>[4]</sup>提出了一种改进协同过滤的个性化 Web 服务推荐算法 UPCF,改进了传统协同过滤算法忽略了用户喜好的问题,提升了算法在用户个性化方面的性能。Wang 等<sup>[5]</sup>提出了一种结合用户和服务相似的服务推荐算法,并在真实数据集上进行了验证。Chen 等<sup>[6]</sup>提出了一种考虑 QoS 波动的协同过滤算法 BRACF,实验表明,他们的方法优于当时最新的协同过滤服务推荐算法。Zhang 等<sup>[7]</sup>提出了一种基于聚类和基于内存的混合 Web 服务推荐算法 CBHP,该方法

结合了基于内存方法易实现的优点和基于模型方法在大规模数据上的自适应性的优点,实验表明该方法可以提高推荐准确性。Sullivan 等<sup>[8]</sup>考虑到用户位置对于服务 QoS 的影响,综合考虑了用户位置信息以及服务的 QoS 信息,提出了一种基于服务历史信息的协同过滤服务推荐算法,并在 Web 服务真实数据集 QWS 上做了验证。Adeli 等<sup>[9]</sup>提出了一种基于服务流行度的协同过滤算法 PDCF,将服务的流行度作为服务推荐的依据,实验结果表明该方法优于传统协同过滤算法。Sun 等<sup>[10]</sup>将改进矩阵分解算法 BaisSVD 应用到服务推荐当中,实验表明,该方法比传统矩阵分解算法的效率更高。Chang 等<sup>[11]</sup>将矩阵分解和图结合起来,将“用户-服务”关系表示成无向图,并实现服务推荐。Tang 等<sup>[12]</sup>将因子分解机引入服务推荐中,实现特征之间二阶交叉,提高了 QoS 预测准确率。

协同过滤(Collaborative Filtering, CF)是推荐系统中的经典算法,其最先与服务推荐相结合。目前基于协同过滤的服务推荐算法主要聚焦在深入挖掘用户-服务矩阵上,但是这一类方法存在信息利用率低的问题,用户-服务矩阵的信息被充分利用,但是上下文信息如服务位置信息、用户位置信息等十分重要的上下文信息被忽略,导致在数据稀疏时,推荐精度大幅下降。

基于矩阵分解的服务推荐方法首次引入了隐向量和特征交叉的概念,并通过用户隐向量和用户隐向量的点积计算最终 QoS 得分,但是由于该类方法只使用“用户”和“服务”两个特征,且特征交叉使用线性的“向量点积”,虽然相比于基于用户和基于服务的协同过滤算法在性能上有所提升,但在面临数据稀疏和大规模推荐场景时,仍然难以对优化目标进行有效的拟合。基于 FM 的服务推荐算法在特征交叉上做了进一步提升,输入特征之间均可以实现两两交叉,由于复杂度的限制,其在二阶以及高阶的交叉理论上可行,但在实际应用中难以实现。为了提高模型的表达能力,本文提出了一种基于多目特征交叉的服务推荐算法 SRMFC。

## 3 SRMFC 算法

本节首先描述了 SRMFC 方法的整体框架和数据计算流程,而后对方法的具体细节进行描述,具体包括模型以及算法。为了更好地说明 SRMFC 的网络结构,对经典 Deep Crossing 方法<sup>[14]</sup>也进行了介绍。SRMFC 总体架构如图 1 所示。

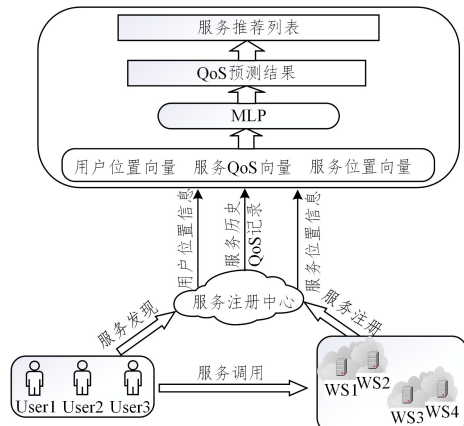


图 1 服务推荐总体架构

Fig. 1 Overall structure of service recommendation

SRMFC 主要分为 3 个功能模块,包括数据收集与整合、网络结构、预测与推荐。图 1 中服务注册中心及其以下部分表示服务消费者、服务提供商、服务注册中心之间的关系,服务推荐算法部署在服务注册中心中,从服务注册中心获取数据,完成训练,为用户提供推荐服务。

### 3.1 数据收集与整合

数据收集与整合部分包括服务位置信息的收集与处理、用户位置信息的收集与处理、服务 QoS 矩阵的构建,上述信息均可从服务注册中心的用户调用服务历史记录中获得。

(1) 服务位置信息包括部署服务的服务器所在经度和纬度,该数据由服务提供商在注册服务时提供给服务注册中心。

(2) 用户位置信息包括用户调用相应服务时所使用设备的经度和纬度,该数据在用户调用过的历史服务记录中。

(3) “用户-服务”QoS 矩阵的构建包括用户排序、服务排序、QoS 指标的选取以及数据的采集。

**定义 1** 满足用户个性化需求的服务 QoS

$$QoS_i = \sum_{j=1}^k \omega_{i,j} * Q_{i,j} \quad (1)$$

其中,  $k$  为选取的 QoS 指标总数;  $\omega_{i,j}$  表示 QoS 指标对应的权重。根据用户的偏好,可将用户区分为速率敏感性型、业务密集型、价格敏感型等群体,对应于不同的权重  $\omega$ ,以提高推荐系统的用户个性化能力。  $Q_{i,j}$  为 QoS 指标,分为连续型和离散型,连续型主要包括 RT, TP, Pr 等,离散型包括 Re 等。针对连续型指标,由于其取值范围相差较大、量纲不同,且部分指标与服务 QoS 成正相关,如 TP,部分反之,如 RT,为统一量纲,使用式(2)对各指标进行标准化处理。

$$Q = \begin{cases} \frac{q - \min}{\max - \min} & (i) \\ \frac{\max - q}{\max - \min} & (ii) \end{cases} \quad (2)$$

其中,(i)表示当 QoS 指标和 QoS 成正相关;(ii)表示当 QoS 指标和 QoS 成负相关。其中  $\max$  和  $\min$  分别为进行标准化的 QoS 指标取值中的最大值和最小值。  $q$  为每个服务对应 QoS 指标的取值。例如:服务集合  $\{ws1, ws2, ws3\}$  的 RT 取值分别为  $\{1, 2, 3\}$ ,那么对于  $ws2$  的 RT 指标规格化过程为:

$Q_2 = \frac{2-1}{3-1} = 0.5$ 。SRMFC 中选取了响应时间 (Responding time, RT) 和吞吐量 (Throughput, TP) 作为 QoS 指标。已有研究表明,绝大多数用户更倾向于响应更快的应用<sup>[10]</sup>,因此,SRMFC 在定义服务 QoS 时,将 RT 的权重设置为 0.6, TP 的权重设置为 0.4,  $QoS_i = 0.6 * RT_i + 0.4 * TP_i$ 。

表 1 列出了最终构建的服务 QoS 矩阵的部分内容,表中“—”表示用户 1 未调用过服务 2。而 SRMFC 算法的最终目的是预测出服务 QoS 矩阵中的“—”位置对应的 QoS 值,并为目标用户推荐高 QoS 的服务。

表 1 部分“用户-服务”QoS 矩阵数据

Table 1 Part of “user-service” QoS matrix data

用户 ID	服务 ID	响应时间	吞吐量	QoS
1	1	0.98	0.334	0.7216
1	2	—	—	—
1	3	0.228	0.543	0.3540
2	1	0.527	0.692	0.5930
...	...	...	...	...

### 3.2 网络结构

为更好地解释 SRMFC 的网络结构,本文首先介绍经典 Deep Crossing 网络结构。Deep Crossing 模型作为推荐系统中具有节点意义的算法,完整地实现了将深度学习应用至特征交叉领域。相比于传统推荐算法独立地使用原始特征,Deep Crossing 着重关注不同特征之间的自动交叉。

如图 2 所示,经典的 Deep Crossing 模型主要包括 4 个部分,分别为嵌入层 (Embedding)、堆叠层 (Stacking)、残差网络层 (Residual Units)、及输出层 (Scoring)。模型的目标函数为:

$$\log loss = -\frac{1}{N} \sum_{i=1}^N (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)) \quad (3)$$

其中,  $i$  表示训练样本,  $N$  表示样本总数,  $y_i$  表示每个样本的标签,  $p_i$  为模型的输出。

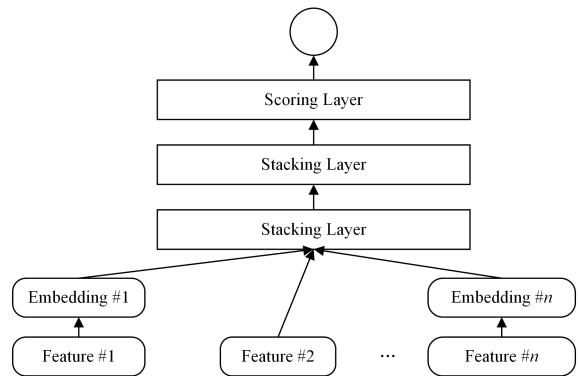


图 2 Deep Crossing 网络结构

Fig. 2 Deep Crossing network structure

Embedding 层的作用在于可将非数值型数据转换成向量。形式化表示为:

$$\mathbf{X}^0 = \text{Max}(0, \mathbf{W}_j \mathbf{X}_j^i + \mathbf{b}_j) \quad (4)$$

其中,  $\mathbf{X}_j^i$  是输入特征,下标  $j$  表示一个特征的索引;  $\mathbf{W}_j$  为参数矩阵,  $\mathbf{b}_j$  为偏置;  $\mathbf{X}^0$  为经过 Embedding 后的输出向量。

堆叠层将多个 Embedding 层输出连接在一起,  $\mathbf{X}^0 = [\mathbf{X}_0^0, \mathbf{X}_1^0, \dots, \mathbf{X}_m^0]$ ,其中  $m$  为 Embedding 层输出的特征数总数。

图 3 为 Deep Crossing 残差网络中的一个残差块,可形式化表示为:

$$\mathbf{X}^0 = \mathcal{F}(\mathbf{X}^l, \{\mathbf{W}_0, \mathbf{W}_1\}, \{\mathbf{b}_0, \mathbf{b}_1\}) + \mathbf{X}^l \quad (5)$$

其中,  $\mathbf{X}^l$  和  $\mathbf{X}^0$  分别表示网络的输入和输出;  $\{\mathbf{W}_0, \mathbf{W}_1\}, \{\mathbf{b}_0, \mathbf{b}_1\}$  分别表示网络的权重和偏置。残差网络的优势在于其能解决网络的表达能力随网络深度增加而退化的问题。

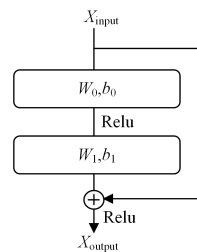


图 3 Deep Crossing 中的残差块结构

Fig. 3 Residual block structure in Deep Crossing

图 4 给出了 SRMFC 的网络结构,自下而上分别为输入层、Embedding 层、连接层、网络层、预测得分层。

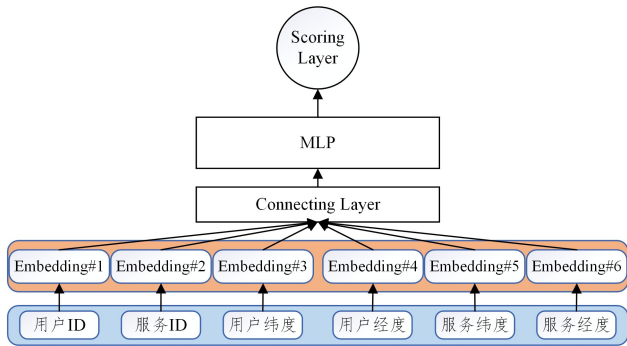


图4 SRMFC网络结构

Fig. 4 SRMFC network structure

SRMFC的输入层包括用户ID、服务ID,以及服务和用户的上下文信息。由于数据集的限制,模型中上下文部分只使用了用户经纬度、服务经纬度。相较于现有的基于聚类的位置感知服务推荐算法<sup>[10-12,15-16]</sup>,SRMFC将用户位置信息、服务位置信息独立编码后喂入网络,作为用户和服务的辅助信息,而不是通过聚类算法先对用户、服务计算各自的相似群体。在时间上,SRMFC避免了聚类过程带来的开销,提高了算法在时间上的性能。在算法精度上,基于聚类的服务推荐算法<sup>[16]</sup>大多需要先人为设定聚类个数这一超参数,算法训练过程中需要对该超参数进行大量且完备的验证以寻找出合适的聚类个数,导致算法的泛化能力差。

Embedding层输出节点为108,每个输入使用单独的Embedding网络。Embedding输出节点数作为模型重要的超参数,网络详细参数设置在4.4节给出。引入Embedding层的目的在于将稀疏的输入向量稠密化,输入向量在经过One-hot编码以后不利于特征之间的交叉。Embedding层的作用可形式化为:

输入 $\mathbf{X}^I = [x_1, x_2, \dots, x_M]_{1 \times M}$ ,有且仅有一个 $x_i \in \{x_1 \dots x_M\}$ 值为1,其他为0。输出 $\mathbf{X}^O = f(\mathbf{X}^I) = [y_1, y_2, \dots, y_N]_{1 \times N}$ , $y_1, y_2, \dots, y_N$ 为实数, $N < M$ 。

若未经过Embedding,则特征之间的交叉存在如下问题:

(1)对于输入特征

$$\mathbf{X}_1^I = [x_{1,1}, x_{1,2}, \dots, x_{1,M}]_{1 \times M}, x_{1,i} = 1, i \in [1, M] \quad (6)$$

$$\mathbf{X}_2^I = [x_{2,1}, x_{2,2}, \dots, x_{2,M}]_{1 \times M}, x_{2,j} = 1, j \in [1, M] \quad (7)$$

(2)交叉特征 $\mathbf{X}_1^I * \mathbf{X}_2^I$

$$\begin{cases} [0, 0 \dots 0, 0]_{1 \times M}, & i \neq j \\ [0 \dots 1 \dots 0]_{1 \times M}, & i = j \end{cases} \quad (8)$$

由式(8)可见,仅当交叉的特征两两相同时,交叉特征才不为0,那么在模型训练时,不相同的特征之间的交叉无论前向、后向传播都无法完成。不同特征之间的信息也会丢失。而经过Embedding后的输出向量 $[y_1, y_2 \dots, y_N]_{1 \times N}$ 中 $y_1, y_2 \dots, y_N$ 均为实数,因此避免了在特征交叉后得到0的可能。

连接层(Connecting Layer)可形式化表示为:

(1)对于来自Embedding层的6个输入

$$\mathbf{X}_j^I = \{x_{(j,1)}, x_{(j,2)}, \dots, x_{(j,108)}\}, j \in [1, 6] \quad (9)$$

(2)输出

$$\mathbf{X}^O = \{x_{(1,1)} \dots x_{(1,108)}, x_{(2,1)} \dots x_{(2,108)}, x_{(3,1)} \dots x_{(3,108)}, x_{(4,1)} \dots x_{(4,108)}, x_{(5,1)} \dots x_{(5,108)}, x_{(6,1)} \dots x_{(6,108)}\} \quad (10)$$

网络层为4层MLP网络,节点数分别为1024,512,256,128。预测得分层输出节点为1,输出的内容为输入用户在

调用输入服务时的QoS估计。网络结构如图5所示,算法步骤如算法1所示。与Deep Crossing不同的是,SRMFC模型并未使用残差网络,原因如下:

(1)残差网络对深层网络的退化问题有效,但SRMFC模型的网络深度只有4层,使用残差网络的提升效果并不明显,反倒增加了网络的参数量。

(2)服务推荐场景中,用户服务矩阵是十分稀疏的,能够提供给模型的训练数据有限,由于相同深度的残差网络的参数量相较于全连接网络要多,因此难以训练出合适的模型。

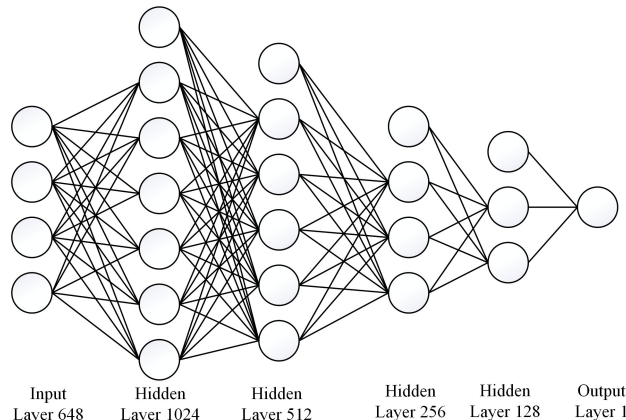


图5 网络层结构

Fig. 5 Network layer structure

### 算法1 SRMFC算法

输入:用户ID、服务ID、用户经纬度、服务经纬度

输出:用户调用服务时的QoS预测

//模型训练

1. 加载训练数据集,将数据集按稀疏度参数 $\tau$ 拆分为 TRAINING\_DATA 和 TEST\_DATA;
  2. 初始化 SRMFC 模型参数,加载学习率 $\eta$ 、学习率衰减系数 $\varphi$ 、BATCH\_SIZE、EPOCH;
  3. For Every EPOCH:
  4. 对 TRAINING\_DATA 中的数据,按 BARCH\_SIZE 大小,喂入网络训练;
  5. 计算训练数据损失并返回损失值 Training\_loss,根据 adam 算法计算梯度,调整网络参数;
  6. 根据网络,预测 TEST\_DATA 中数据的值,并与实际标签值计算 MAE 误差,返回误差值(Test\_loss);
  7. 监测 Test\_loss,当 Test\_loss 与上一轮相比出现增长时, $\eta = \eta * \varphi$ ;
  8. Test\_loss 累计增长 5 次时,提前结束训练,否则直至所有 EPOCH 循环结束后结束训练;
  9. 返回训练后的模型并保存。
- // 模型预测
10. 加载训练完成的模型;
  11. 读取输入数据;
  12. 输出服务的预测 QoS 值。
  13. 对输出 QoS 排序,根据 Top-k 为用户推荐 k 个服务。
  14. End for

其中稀疏度参数 $\tau$ 的设定是为了对比算法在不同数据稀疏性的场景下的推荐准确率,实验中采取改变测试集数据(TEST\_DATA)占整个数据集(DATASET)比例的方法,模拟数据稀疏性;稀疏度 $\tau \in (0, 1)$ , $\tau$ 的值越大,表示数据越稀疏。表示具体定义方法如式(11)所示:

$$\tau = \frac{TEST\_DATA}{DATASET} \quad (11)$$

例如:稀疏度  $\tau=0.9$  表示训练数据占数据集的 10%,测试数据占数据集的 90%。

## 4 实验与分析

### 4.1 数据集与清洗

为了检验 SRMFC 算法在真实服务推荐场景下的有效性,我们采用 Zheng 等提出的真实 Web 服务 QoS 数据集 WSDream<sup>[13]</sup>,该数据集采集了 339 个用户对于 5 825 个 Web 服务的响应时间、吞吐量、用户和服务经纬度、用户 IP 地址等共 1 974 675 条数据。数据集中存在缺失数据,如部分只有响应时间或只有吞吐量,对于这部分数据,在生成训练数据和测试数据之前将其剔除,并根据式(1)、式(2)生成训练标签。

### 4.2 评估指标

针对特定用户的服务推荐列表根据服务的 QoS 生成,QoS 预测的精度直接决定了最终的服务推荐列表。因此,实验通过比较测试集服务 QoS 预测值,与真实值的误差来评估算法的推荐性能,实验中使用平均绝对误差(Mean Absolute Error, MAE)和均方根误差(Root Mean Squared Error, RMSE)作为误差度量指标。MAE 和 RMSE 的定义如式(12)、式(13)所示:

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - f(x_i)| \quad (12)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^m (y_i - f(x_i))^2}{m}} \quad (13)$$

其中, $f(x_i)$ 表示待评估模型的预测值, $y_i$ 表示样本真实值, $m$ 表示样本总数。MAE、RMSE 的值越小,表示预测值与实际值之间的误差越小,模型的推荐性能越好。

### 4.3 算法准确性对比

首先,对 SRMFC 算法和目前主流的服务推荐算法的性能进行比较,选取的对比方法主要包括:

UPCC<sup>[17]</sup>和 IPCC<sup>[18]</sup>:这两个方法是服务推荐中传统的基于用户的协同过滤,以及基于服务的协同过滤。

PDCF<sup>[9]</sup>:该方法以服务流行度作为是否推荐的评价指标,其他服务推荐算法中尚未使用过该指标。

BiasSVD<sup>[10]</sup>:基于 FunkSVD 改进的 BiasSVD,在 FunkSVD 的基础上改进了目标函数,加入了偏置项。

UPCF<sup>[3]</sup>:该方法考虑了用户的偏好,通过引入偏好信息提升推荐精度。

GMF<sup>[11]</sup>:该方法结合无向图和矩阵分解,将“用户-服务”矩阵的关系通过无相图表示,以实现 QoS 预测。

BRACF<sup>[6]</sup>:该方法考虑了不同 QoS 指标之间的量纲差异,设计了相应的相似度计算方法。

LAFM<sup>[12]</sup>:该方法在因子分解机的基础上,加入用户位置信息,以预测服务 QoS。

实验中设置稀疏度分别为 80%,85%,90%,95%,对各个算法的推荐精度进行验证。

表 2 不同稀疏度下各算法的推荐精度对比

Table 2 Comparison of recommendation accuracy of each algorithm with different sparsity

Methods	稀疏度 80%		稀疏度 85%		稀疏度 90%		稀疏度 95%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UPCC	0.4865	1.2201	0.5157	1.2604	0.5552	1.3157	0.6362	1.3805
IPCC	0.4564	1.2075	0.5102	1.2614	0.5946	1.3435	0.6357	1.3995
PDCF	0.5665	1.5169	0.6098	1.6434	0.6117	1.6746	0.7223	1.8940
BiasSVD	0.4874	1.3838	0.5295	1.4025	0.5943	1.4237	0.6334	1.4956
UPCF	0.4310	1.1604	0.4775	1.2021	0.5380	1.2835	0.5965	1.4732
GMF	0.4233	1.0963	0.4327	1.1512	0.4737	1.2432	0.5301	1.3657
BRACF	0.4014	1.1034	0.4103	1.1423	0.4312	1.2312	0.5003	1.2902
LAFM	0.3999	1.0923	0.4259	1.1106	0.4498	1.1926	0.5136	1.2674
SRMFC	0.2958	1.0105	0.3040	1.0834	0.3208	1.1205	0.3556	1.1734

从表 2 中可以看出,各算法的评估指标均随着稀疏度的增加而增大,尤其是 UPCC,IPCC,PDCF 这 3 种方法只利用了用户和服务信息而未考虑上下文,随着数据的稀疏性增加,算法性能明显下降。而 BiasSVD 虽然与 UPCC,IPCC,PDCF 这 3 种方法相同,只利用了用户和服务信息,但是该方法通过引入用户、服务隐向量的方式,实现了特征之间的交叉关系挖掘,在 UPCC,IPCC,PDCF 这 3 种方法的基础上,提高了特征的表达能力,从而获得了更高的推荐精度。UPCF,GMF,BRACF,LAFM 以及 SRMFC 均引入了用户、服务之外的其他特征,如位置信息、用户兴趣等。虽然评估指标会随着稀疏性的增加而增加,但是增幅小于 UPCC,IPCC,PDCF 等方法。相比于基于 MF 的 GMF 以及基于 FM 改进的 LAFM 算法,SRMFC 在推荐精度上的优势十分明显,在稀疏度分别为 80%,85%,90%,95%的情况下,SRMFC 的 QoS 预测误差比目前预测性能最优的 LAFM 算法分别提高了 26%,28%,28%,30%。分析认为:多目特征的引入有助于提升算法在应对冷启动时的推荐精度。在输入特征相同的情况下,SRMFC 相比于其他算法,可以实现

特征之间更高阶的交叉,同时激活函数的存在也为特征交叉引入了非线性成分,并最终使得 SRMFC 算法推荐精度的大幅提升。

### 4.4 Embedding 层节点数量的影响

Embedding 层节点数直接影响模型输入信息的表示,节点数过少会导致各输入信息之间不能有效区分,不同输入值的输出值之间相似度过高,不利于后续模型预测;节点数过多则会造成单个 Embedding 层参数过多,网络过于复杂,训练难以收敛并最终影响模型的预测能力。因此,实验首先对该超参数进行验证。

图 6 给出了 Embedding 层节点数对测试集误差的影响。实验中其他参数设置为:数据稀疏度  $\tau=0.95$ , BATCH\_SIZE=64,初始学习率为 0.01,学习率衰减系数为 0.1, EPOCH=35。从图中可以看出,当节点数由低到高增长时,测试集 MAE 先降低后升高,最优值取在当 Embedding 节点数为 128 时。由此可得,当 Embedding 节点数为 128 时,可以平衡数据表示的完整性和 Embedding 层训练的可靠性。后续实验中均取 Embedding 层节点数为 128。

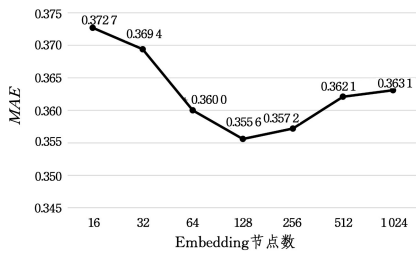


图6 Embedding层节点数量对推荐精度的影响

Fig. 6 Influence of number of embedded layer nodes on recommendation accuracy

#### 4.5 BATCH\_SIZE 的影响

实验中涉及多个稀疏度不同的实验,稀疏度直接决定了训练数据量。不同的训练数据量需要适配不同的 BATCH\_SIZE,因此,实验验证了 BATCH\_SIZE 对测试误差的影响。

图7给出了稀疏度分别为80%,85%,90%,95%时,BATCH\_SIZE对于最终测试集误差的影响。从图中可以看出,BATCH\_SIZE的大小对算法推荐精度影响不大,当BATCH\_SIZE的值小于128时,随着BATCH\_SIZE的增长,MAE的值出现波动,但整体上稳定;当BATCH\_SIZE的值大于128时,MAE的值随着BATCH\_SIZE的值增大而逐步增大,增幅为在实验最优值上增加1%~2%。总体上该实验中BATCH\_SIZE对于最终MAE的影响不太明显。分析后认为:出现这种情况的原因在于数据集中数据分布较为平缓,波动不明显,导致BATCH\_SIZE对于最终结果的影响不显著。在保证训练精度的同时,更大的BATCH\_SIZE有利于节省训练时间,因此训练中BATCH\_SIZE的值均选取128。

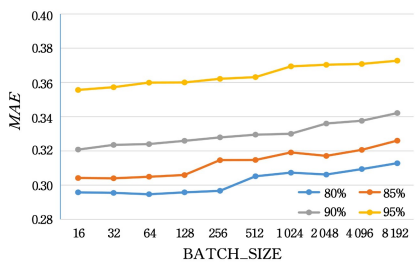


图7 不同稀疏度下 BATCH\_SIZE 的大小对实验精度的影响

Fig. 7 Influence of BATCH\_SIZE on experimental accuracy with different sparsity

#### 4.6 稀疏度对于预测精度的影响

从图8中可以看出,随着稀疏度的增加,QoS预测精度逐渐降低,稀疏度低于95%时,MAE降低较为平缓,随着稀疏度继续增长,MAE曲线趋于陡峭。虽然与目前性能优异的算法相比,SRMFC在各个稀疏度下的性能都比其他算法更优,但是在极度稀疏,如稀疏度99.5%时,其性能还可以再优化。

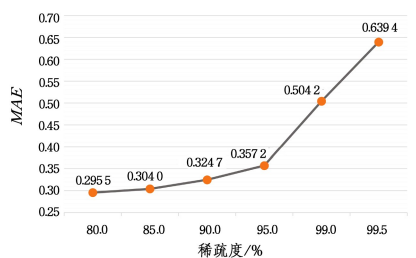


图8 稀疏度对于推荐精度的影响

Fig. 8 Influence of sparsity on recommendation accuracy

**结束语** 服务过载的问题是目前服务发现中的核心问题,通过服务推荐的方法解决服务过载问题是目前主流的解决思路。本文提出的基于多特征交叉的服务推荐算法,通过引入更多的上下文信息及应用神经网络实现高阶特征交叉,提高了算法在冷启动以及历史数据稀疏的情况下的推荐精度,其推荐精度高于目前所有的服务推荐算法。下一步的工作将放在提升数据极度稀疏下的推荐精度上。之所以在第4节实验部分验证99.5%稀疏度下的推荐精度,是因为根据数据集的大小可计算出99.5%稀疏度下,平均到每个用户所使用的服务数量为29个。但对于大多数用户而言,其使用过的服务数量可以用个位数衡量,这种情况下,稀疏度将增长至99.95%以上。如何提高极度稀疏性场景下的推荐精度将会成为决定服务推荐能否更好地提升用户体验的关键。

#### 参考文献

- [1] AL-MASRI E, MAHMOUD Q H. Discovering the best web service: A neural network-based solution [C] // IEEE International Conference on Systems, San Antonio IEEE, 2009.
- [2] LIN X Y, LIU X Q, TANG M D, et al. An empirical study of correlation between Web service QoS and user location [J]. 2013, 35(9): 83-88.
- [3] PEERZADE S S. Web service recommendation using PCC based collaborative filtering [C] // 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS). IEEE, 2018.
- [4] XU K, ZHU X K, JING X Y. Research on Personalized Web Service Recommendation Based on Improved Collaborative Filtering [J]. Computer Technology and Development, 2018, 28(1): 64-68.
- [5] WANG S, ZHAO Y, HUANG L, et al. QoS prediction for service recommendations in mobile edge computing [J]. Journal of Parallel & Distributed Computing, 2017, 127(MAY): 134-144.
- [6] CHEN Z, SHEN L, LI F, et al. Web service QoS prediction: when collaborative filtering meets data fluctuating in big-range [J]. World Wide Web, 2020, 23(3): 1715-1740.
- [7] ZHANG X J, WANG Z J, ZHANG W J. Personalized Web Services Recommendation Based on Hybrid Collaborative Filtering Algorithm [J]. Journal of Frontiers of Computer Science and Technology, 2015, 9(5): 565-574.
- [8] SULLIVAN A, CATUR CANDRA M Z. Web Service Recommendation System using History and Quality of Service [C] // 2019 International Conference on Data and Software Engineering (ICoDSE). 2019: 1-6.
- [9] ADELI S, MORADI P. QoS-based Web Service Recommendation using Popular-dependent Collaborative Filtering [J]. Journal of AI and Data Mining, 2020, 8(1): 83-93.
- [10] SUN D, NIE T. A web service recommendation algorithm based on BaisVD [C] // 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC). 2020: 29-32.
- [11] CHANG Z, DING D, XIA Y. A graph-based QoS prediction approach for web service recommendation [J]. Applied Intelligence, 2021, 51: 6728-6742.

- [12] TANG M D,ZHANG T T,YANG Y T,et al. Quality-aware Web service recommendation method based on factorization machine[J]. Chinese Journal of Computers, 2018, 41 (6): 1300-1313.
- [13] SHAN Y,HONES T R,JIAO J,et al. Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features [C]// The 22nd ACM SIGKDD International Conference. San Francisco, United States, ACM, 2016.
- [14] ZHENG Z B, LYU M R. WS-DREAM: A distributed reliability assessment Mechanism for Web Services[C]// 2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC(DSN). 2008:392-397.
- [15] TANG N ,XIONG Q Y,WANG X B,et al. Web service recommendation based on location clustering and tensor decomposition [J]. CEA, 2016, 52(15):65-72.
- [16] BOTANGEN K A, YU J, SHENGQ Z, et al. Geographic-aware Collaborative Filtering for Web Service Recommendation[J]. Expert Systems with Applications, 2020, 151: 113347.
- [17] SHAO L,ZHANG J,WEI Y,et al. Personalized QoS prediction for web services via collaborative filtering[C]// IEEE International Conference on Web Services. Salt Lake City, UT, 2007: 439-446.
- [18] SARWAR B,KARYPIS G,KONSTAN J,et al. Item-based collaborative filtering recommendation algorithms [C] // Proceedings of International Conference on World Wide Web. ACM Press, 2001:285-295.



**GAO Wenbin**, born in 1995, postgraduate. His main research interests include big data application and services computing.



**HU Guyu**, born in 1963, Ph.D, professor, Ph.D supervisor. His main research interests include computer networks, administration of the satellite networks and intelligent network management.