

基于因子分解机的质量感知 Web 服务推荐方法

唐明董^{1),2)} 张婷婷²⁾ 杨亚涛³⁾ 郑子彬³⁾ 曹步清²⁾

¹⁾(广东外语外贸大学信息科学与技术学院 广州 510006)

²⁾(湖南科技大学计算机科学与工程学院 湖南 湘潭 411201)

³⁾(中山大学数据科学与计算机学院 广州 510006)

摘 要 随着 Web 服务市场的发展,Web 服务数量日益庞大,服务的质量日益受到重视,因此为用户评估服务质量并推荐高质量的服务成为了极为重要的问题.传统的协同过滤方法用于 Web 服务推荐,可能会因为数据稀疏导致性能不高.近年较流行的矩阵分解技术可以用来克服推荐系统的数据稀疏问题,但是计算的时间复杂度较高,可扩展性差.为同时提高 Web 服务质量推荐的精度和效率,文中引入了一种通用的因子分解机模型到 Web 服务推荐中.因子分解机具有线性的计算时间复杂度,不仅能适应数据高度稀疏的推荐系统环境,而且很容易结合用户和推荐对象的上下文信息以进一步提升性能.文中提出的质量感知 Web 服务推荐方法,是在因子分解机的基础上,考虑了 Web 服务质量与用户(或服务)位置之间的相关性.该方法先利用位置信息计算每个用户(或服务)的相似邻居,然后将相似用户(或服务)的影响与因子分解机结合来为目标用户预测目标服务的质量,最后在预测服务质量的基础上为目标用户发现和推荐高质量的服务.在真实的 Web 服务调用数据集上开展实验表明,该方法在预测精度上优于其它协同过滤及因子分解推荐算法.同时由于该方法具有较低的时间复杂度,可以较好地解决大规模 Web 服务推荐系统的可扩展问题.

关键词 服务推荐;分解机;协同过滤;QoS 预测;位置感知;服务计算

中图法分类号 TP301 **DOI 号** 10.11897/SP.J.1046.2018.01300

QoS-Aware Web Service Recommendation Based on Factorization Machines

TANG Ming-Dong^{1),2)} ZHANG Ting-Ting²⁾ YANG Ya-Tao³⁾ ZHENG Zi-Bin³⁾ CAO Bu-Qing²⁾

¹⁾(School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510006)

²⁾(School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, Hunan 411201)

³⁾(School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510006)

Abstract With the development of the Web services market, the amount of Web services becomes increasingly massive and quality of service (QoS) plays a more important role than ever. Against this background, identifying and recommending high-quality services for users has become a vital issue. Since a Web service is likely to exhibit different QoS performance to users due to their different network conditions, personalized Web service recommendation becomes a requisite. Based on historical service invocation records, traditional collaborative filtering approaches such as memory-based collaborative filtering have been employed by some studies to address the Web service recommendation problem. However, these approaches usually suffer from the data sparsity and cold-start issues, which probably cause low performance or even failures of Web service QoS predictions. Recently there is a trend of applying factorization models such as Matrix Factorization (MF) to recommender systems to tackle the data sparsity issue. Though their

收稿日期:2016-10-24;在线出版日期:2017-08-22. 本课题得到国家自然科学基金(61572186,61572188,61402168)、湖南省高校创新平台开放基金(15K043)、湖南省研究生科研创新项目(CX2016B574)资助. 唐明董,男,1978年生,博士,教授,中国计算机学会(CCF)会员,主要研究领域为服务计算、软件工程与数据挖掘等. E-mail: mdtang@126.com. 张婷婷,女,1992年生,硕士,主要研究方向为服务计算. 杨亚涛,男,1992年生,硕士,主要研究方向为服务计算. 郑子彬,男,1982年生,博士,副教授,博士生导师,主要研究领域为服务计算与大数据分析. 曹步清,男,1979年生,博士,副教授,主要研究方向为服务计算与面向服务的软件工程.

application in Web service recommendation can yield more accurate Web service QoS predictions, they usually have a high time complexity and poor scalability. What's more, adding extra features to those factorization models is yet a non-trivial task. In order to attain both high accuracy and high efficiency in Web service recommendation, this paper adopts a general factorization model, i. e. , Factorization Machines (FM), which are actually a new model class that combines the advantages of Support Vector Machines (SVM) with factorization models. The model equation of FMs can be calculated in linear time, and FMs not only are adaptable to high data sparsity settings, but also are flexible to combine the context information of users and items as well as their interactions. Based on the classic FMs, the following paper proposes a QoS-aware Web service recommendation approach which is a location-aware FM model that takes into account the influence of Web service locations and user locations on Web service QoS prediction. Based on the observation that users in the same or close regions usually would perceive similar QoS values when invoking a Web service, the approach employ the location information to identify a set of neighbors for the active user, and combine the experiences of the active user's neighborhood to predict the QoS of a target Web service. In a similar manner, the neighborhood of a target Web service can also be located and their QoS values are employed to predict its QoS values for the active user. Based on the neighborhood computation of both users and services, we develop several neighborhood-based regularization terms and integrate them with the classic FMs to improve the QoS prediction accuracy. To evaluate the performance of the proposed approach, we conduct a set of extensive experiments using a real-world dataset with 1974675 Web service invocations. The experimental results show that the proposed approach outperforms the other collaborative filtering approaches (including popular matrix factorization approaches) in QoS prediction accuracy. We also conduct an analytical analysis to show that the proposed approach has a low computation time complexity, which indicates that it has a good scalability and thus is suitable for large-scale Web service recommender systems.

Keywords service recommendation; factorization machines; collaborative filtering; QoS prediction; location-aware; service computing

1 引 言

Web 服务是一种基于 Web 环境的具有自适应、自描述、模块化并具有良好的互操作能力的应用程序模块,广泛用于开发大规模的分布式应用程序. 随着 Web 服务数量的日益庞大,以及具有相同或者相似功能的 Web 服务的不断涌现,如何发现和选择高质量的服务成为了一个难题. 在此背景下,服务推荐技术开始得到关注^[1]. 为了向用户推荐高质量的服务,一个关键问题是如何获得 Web 服务的质量参数值. Web 服务的质量参数一般用 QoS(Quality of Service)代表. 然而,现实中多数 Web 服务的 QoS 信息对用户来说是未知的. 并且,Web 服务的 QoS 参数(如响应时间、吞吐量等)会受网络环境等因素的影响^[2],导致不同位置的用户在同一个服务上可

能观察到不同的 QoS 值. 尽管用户可以通过亲自调用 Web 服务来评估它的 QoS,但是由于服务的用户并不是评价服务的专家,要在短时间对大量候选服务的 QoS 进行准确评估是不太现实的.

由于 Web 服务的 QoS 值是与具体用户相关的,近年来不少工作^[2-5]利用协同过滤技术来进行个性化的 QoS 预测和服务推荐,取得了一定的成效. 然而,传统的协同过滤技术在应用时受数据稀疏的影响较大,且存在冷启动以及可扩展性差等问题. 最近,因子分解等基于模型的协同过滤技术在推荐系统领域吸引了越来越多的关注,并被引入到 Web 服务推荐中^[6-8]. 这类方法的优点是可以在一定程度上缓解数据稀疏和冷启动用户问题,因此有利于提高推荐准确性. 但是以往的多数工作对 Web 服务的 QoS 特性并没有充分的认识和加以利用,因此在 QoS 预测方面的性能还有待进一步提高.

为了说明本文的研究动机,图 1 给出了一个 Web 服务调用场景,其中 $s_1, s_2, \dots, s_5$ 代表 Web 服务, $u_1, u_2, \dots, u_7$ 代表分布在不同地区的服务用户,它们之间的连接表示用户-服务的历史调用关系.假设表 1 中的数值代表图 1 中服务调用产生的响应时间,符号“-”代表对应的用户-服务之间没有发生调用.以往对真实 Web 服务调用数据的分析^[9-10]表明,同一地区的用户,由于通常具有相似的网络环境和网络性能,有较大可能在相同 Web 服务上观察到相似的性能.例如,同一个互联网自治系统(Autonomous System, AS)的用户,他们访问同一组 Web 服务所产生的响应时间,用皮尔逊相关系数计算,它们的相似度在 0.8 以上^[9-10].因此,图 1 中 u_1 和 u_2 可能在服务 s_2 上体验到相似的响应时间, u_6 和 u_7 可能在服务 s_5 上体验到相似的响应时间.如何利用这些 Web 服务 QoS 特性改进 QoS 预测性能是本文工作的目标.为此,本文提出利用 Web 服务 QoS 的位置相关特性,将之与最新的因子分解机(Factorization Machine, FM)模型结合,提出了一种位置感知的因子分解机模型(Location-Aware Factorization Machine),简称 LAFM.概括地说,本文工作主要有以下贡献:

(1)将最新的因子分解机模型引入到 Web 服务推荐及 QoS 预测中,与传统的协同过滤方法相比,因子分解机结合了支持向量机和传统因子分解模型的优点,对用户与服务的每一次交互使用因子分解参数进行建模,能够进一步提高 Web 服务 QoS 预测的性能;

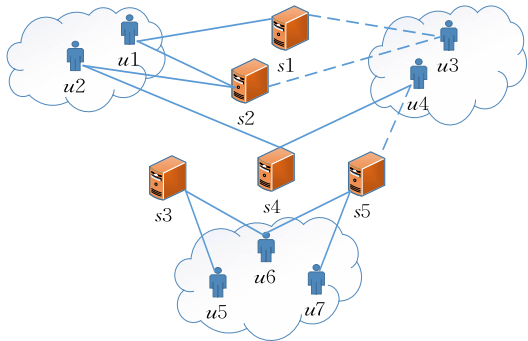


图 1 一个 Web 服务调用场景

表 1 用户调用服务的响应时间 (单位:s)

	s1	s2	s3	s4	s5
u1	0.8	0.3	-	-	-
u2	-	0.4	-	1.2	-
u3	1.5	2.0	-	-	-
u4	-	-	-	0.2	1.0
u5	-	-	0.5	-	-
u6	-	-	0.4	-	0.1
u7	-	-	-	-	0.2

(2)利用 Web 服务和用户的网络位置信息,并将之与因子分解机结合,提出了一种位置感知的因子分解机模型,在为当前用户预测目标 Web 服务的 QoS 时,将相似用户和相似服务的作用也考虑进来,进一步提高了 QoS 预测性能;

(3)使用真实的 Web 服务 QoS 数据集对提出的方法进行了实验评估.结果验证了所提方法具有较高的性能,而且表明该方法可以很好地处理数据稀疏和冷启动用户问题.

2 相关工作

协同过滤^[11]是目前最流行的个性化推荐技术,被广泛用于各种商业推荐系统.协同过滤方法大致分为基于记忆的方法和基于模型的方法两种.基于记忆的方法又分为基于用户的和基于物品的.基于用户的协同过滤方法^[12]通过发现与当前用户偏好相似的用户,将它们喜欢的物品推荐给当前用户.基于物品的协同过滤方法^[13]通过发现与当前用户消费过的物品相似的其它物品,并将之推荐给当前用户.基于模型的方法^[14]倾向于通过对历史数据进行学习来建立一个全局模型,从而预测用户-物品之间的偏好关系.协同过滤方法最近在 Web 服务 QoS 预测和 Web 服务推荐中得到了应用.

Shao 等人^[2-3]提出了基于用户的协同过滤 Web 服务 QoS 预测方法. Zheng 等人^[4]结合了基于用户和基于服务的协同过滤技术,提出了一种混合式协同过滤 Web 服务推荐方法.在它们的基础上,一些工作提出了改进的方法.这些工作主要分为两类.一类工作主要聚焦于充分挖掘用户-服务 QoS 矩阵中的隐藏信息和特征来改进协同过滤.例如 Jiang 等人^[15]通过利用用户和服务的个性化 QoS 特征改进它们的相似度量. Wu 等人^[16]和 Chen 等人^[17]使用数据平滑和填充技术对用户-服务 QoS 矩阵进行预处理,可以在一定程度上克服数据稀疏问题,从而提高 QoS 预测精度. Ma 等人^[18]在分析客观的 Web 服务 QoS 值的基础上,对传统协同过滤方法进行精细化定制,提升了 QoS 预测性能.第二类工作不仅利用 QoS 矩阵中的信息,同时也考虑了用户或服务的上下文信息(如位置)来改进协同过滤.例如, Chen 等人^[19]利用 IP 地址来对用户进行聚类,从而提高相似用户发现的效率和质量. Lo 和 Yin 等人^[20]利用地理位置(经纬度)来帮助发现相似用户,并与协同过滤结合以改进服务 QoS 预测.还有一些

工作^[9-10,21]则利用了用户和服务的网络位置信息来帮助改进 Web 服务 QoS 预测。

基于记忆的协同过滤方法的缺点是在数据很稀疏时精度不高,而且计算相似度的时间开销比较大,可扩展性较差。现实中,Web 服务和服务用户的数量都增长很快,且每个用户往往只调用了少数 Web 服务,故用户-服务 QoS 矩阵可能非常稀疏。因此,基于记忆的协同过滤服务推荐方法在实际应用时可能会低效。为了解决推荐系统中存在的数据稀疏、冷启动和可扩展性问题,基于模型的协同过滤技术(如因子分解)开始受到关注,并被用于 Web 服务 QoS 预测和服务推荐。例如,Zheng 等人^[6]提出了一种基于矩阵分解的服务 QoS 预测方法,先计算用户之间的相似性,在假定相似用户具有相似潜在特征的基础上,改造经典的矩阵分解模型,从而提高 QoS 预测性能。Xu 等人^[22]将用户的信誉考虑进来,在使用矩阵分解进行 QoS 预测时,赋予信誉较高的用户较大的权重,可以提高存在不可靠 QoS 数据时的预测准确性。Wang 等人^[23]针对多维 QoS 值预测提出了基于张量分解的方法。

尽管基于因子分解的 QoS 预测方法要优于传统的协同过滤方法,但是仍然存在通用性差、可扩展性不高等问题。为了克服传统因子分解模型存在的问题,Rendle 在 2010 年首次提出了因子分解机^[24]。因子分解机不仅结合了传统因子分解模型和支持向量机的优势,还利用因子分解对变量之间的交互进行建模,特别适合于数据稀疏的环境。最近的一些研究^[25-28]表明,与传统的因子分解模型(如矩阵分解)相比,因子分解机不仅可以提高推荐的精度,还可以提高推荐的时间效率。并且,很容易将用户和项目的上下文信息结合进来对因子分解机进行扩充。受此启发,本文针对 QoS 感知的 Web 服务推荐,提出了一种改编的因子分解机模型。该模型将

用户和服务的网络位置信息结合到经典的因子分解机中,充分利用当前服务、目标服务以及它们的相似邻居之间的关系,以提升 Web 服务 QoS 预测的性能。

3 方 法

本节先介绍 LAFM 方法的整体框架和总体流程,然后展开描述 LAFM 方法的细节,包括模型和算法步骤。为了便于理解,本节对经典的因子分解机模型也进行了描述。LAFM 方法的总体框架如图 2 所示。LAFM 主要包括以下几个功能模块:

- (1) 用户-服务 QoS 矩阵构建。该矩阵收集了全体用户调用 Web 服务的历史记录。矩阵中每一项记录,代表了一个用户与一个 Web 服务的一次交互,可能包含多个 QoS 参数值。
- (2) 相似用户邻居计算。根据当前用户所在的网络位置,综合考虑其他用户的位置和 QoS 数据来计算他们与当前用户的相似性,以便为当前用户找到相似用户集合。
- (3) 相似服务邻居计算。根据当前服务所在的网络位置,综合考虑其他服务的位置和 QoS 数据来计算它们与当前服务的相似性,以便为当前服务找到相似服务集合。
- (4) 基于因子分解机的 QoS 预测。计算出相似用户和相似服务后,通过将相似用户和相似服务信息添加到经典的因子分解机,得到改进的因子分解机模型。该模型不仅利用当前用户与服务的交互关系,还利用了它们与相似用户和相似服务的关系来进行 QoS 预测。
- (5) Web 服务推荐。根据预测的 QoS 值,可以为用户对候选 Web 服务的质量进行评估和排序,从而为用户推荐预测质量较好的候选 Web 服务。

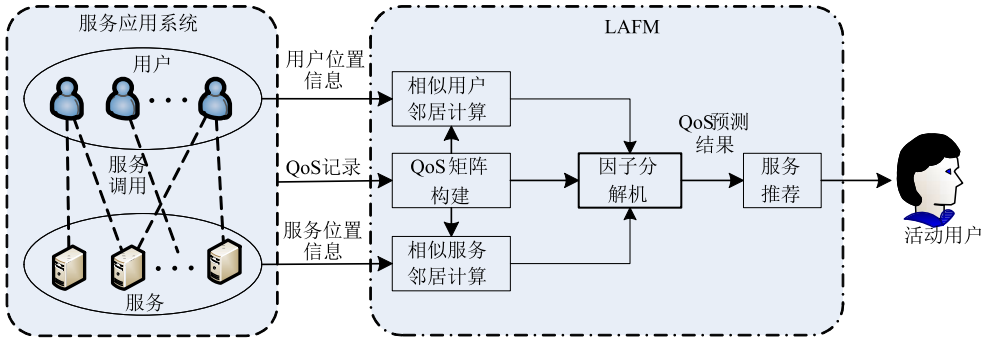


图 2 LAFM 方法的总体框架

3.1 QoS 矩阵的构建

基于协同过滤的 Web 服务推荐依赖于历史的用户-服务调用记录,需要获得服务调用所产生的 QoS 数据.主要有两种方法.一种方法是利用服务用户的反馈,即允许和鼓励用户在使用 Web 服务后反馈它的 QoS 信息,包括响应时间、可用性及信誉等.另一种方法是使用 QoS 监测系统,监测 Web 服务的调用并记录其 QoS 信息.在传统的 Web 服务部署方式下,这种方法比较难以实现,因为 Web 服务常被部署在提供者的服务器上,服务的 QoS 难以监测.然而,随着云计算的广泛应用,越来越多的 Web 服务部署在云平台上,因而云平台提供者有能力使用服务 QoS 监测系统监测部署其上的服务质量^[29].现实中,这样的服务 QoS 监测系统有可能成为云计算平台的一种基础设施.所有服务调用产生的 QoS 记录可以用一个矩阵来表示,文中称为 QoS 矩阵,其中每个 QoS 记录可能包括多个 QoS 参数值.为了便于后面的阐述,我们使用下列符号对用户、服务、矩阵和 QoS 记录进行形式化定义:

(1) 设 U 代表所有 Web 服务用户集合, S 代表所有 Web 服务集合.

(2) $M = \{r_{i,j} | u_i \in U, s_j \in S\}$ 代表用户调用服务产生的 QoS 记录的集合.其中每行代表一个用户的 QoS 向量;每列代表一个服务的 QoS 向量;每个 QoS 记录 $r_{i,j}$ 代表用户 u_i 调用服务 s_j 产生的 QoS 值.如果 u_i 没有调用过 s_j ,则 $r_{i,j}$ 为空.

真实场景中,由于一个用户往往仅使用了全部 Web 服务中的极小部分,因此上述由用户和服务交互形成的 QoS 矩阵应当十分稀疏,这对准确的 Web 服务 QoS 预测构成了挑战.

3.2 相似邻居的计算

协同过滤算法的一个重要步骤是查找相似邻居.传统的协同过滤方法需要搜索整个用户(服务)集合来计算相似用户(服务).当用户和服务数量很庞大时,这种方法可能低效.现实中,Web 服务的 QoS 受到用户和服务位置的影响.位置靠近的用户在同一个服务上体验到的 QoS(如响应时间)常常较为接近;来自相同地区的两个服务对同一个用户来说, QoS 也有较大可能比较相似.因此,本文利用位置信息来改善相似用户或相似服务的查找,即优先在本地为当前用户(服务)查找相似用户(服务),如果不能找到,再扩大范围查找.本文使用国家和 AS 为用户和服务划分区域,相比于其它类型的位置信息(如经纬度),本文提出的划分方法与互联网

的划分较为一致,因此更加自然和准确^[9-10].图 3 显示了一个为用户划分区域的例子,先按国家划分,然后可以继续按 AS 划分.为了查找相似邻居,我们按照 AS、国家、全体集合从小到大的范围进行搜索.以查找相似用户为例,其具体步骤描述如下:

(1) 先搜索与当前用户 a 位于同一个 AS 的 Web 服务用户,计算 a 和其他用户间的相似度,然后采用传统的 Top-K 算法确定相似集合的大小.如果在同一个 AS 的用户数不足 K 个,则进行下一步;

(2) 搜索与当前用户 a 位于同一个国家的 Web 服务用户,计算 a 和其他用户间的相似度,采用 Top-K 算法获得最相似的用户.如果找到的相似用户数量不够 K 个,则继续下一步操作;

(3) 搜索全体用户集合,计算当前用户和其他用户间的相似度,然后,采用 Top-K 算法获得最相似的用户集合,完成相似用户的查找.

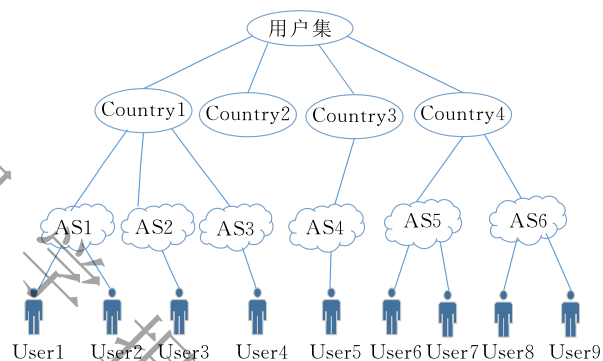


图 3 对用户划分区域

在上述步骤中,一个关键问题是如何计算不同用户间的相似度.传统的协同过滤方法常常采用皮尔逊相关系数(Pearson Correlation Coefficient, PCC)来度量用户间的相似性.设 a 为当前用户,使用 PCC 计算它和用户 u 的相似度的公式如下:

$$\text{sim}(a, u) = \frac{\sum_{s \in S_a \cap S_u} (r_{a,s} - \bar{r}_a) \cdot (r_{u,s} - \bar{r}_u)}{\sqrt{\sum_{s \in S_a \cap S_u} (r_{a,s} - \bar{r}_a)^2} \sqrt{\sum_{s \in S_a \cap S_u} (r_{u,s} - \bar{r}_u)^2}} \quad (1)$$

其中, $S_a \cap S_u$ 代表 a 和 u 调用过的共同服务集合; $r_{a,s}$ 为用户 a 对服务 s 观察到的 QoS 值, $\bar{r}_u$ 为用户 u 调用服务观察到的平均 QoS 值; $\text{sim}(a, u) \in [-1, 1]$ 且值越大说明两个用户越相似.一般认为 $\text{sim}(a, u) \leq 0$ 意味着两个用户不相似.

然而,传统的 PCC 在数据稀疏时或者面对冷启动用户可能失效.例如,如果当前用户 a 使用过的服务太少或者 a 是从未使用过任何服务的新用户,它

和邻近的用户可能没有共同使用过的服务,那么 PCC 相似度计算结果是 0(代表不相似),因此可能会降低查找相似用户的成功率.为此,我们使用一种改进的皮尔逊相关系数(PCC)来计算不同用户间的相似度.我们预先需要将当前用户 a 的缺失 QoS 值用相同区域的用户的平均 QoS 值进行填充.预填充方法可以增加训练数据的密度,在一定程度上可以提高推荐系统的预测质量^[16,30].考虑到相同地区的 Web 服务用户常常有相似的 Web 服务调用体验,我们采用如下的填充方法:

- (1) 在前述的步骤(1)中用同一个 AS 中的所有用户的平均 QoS 值填充当前用户的缺失 QoS 值;
- (2) 在步骤(2)中用同一个国家中的所有用户的平均 QoS 值填充缺失 QoS 值.

同一个 AS 或者国家的用户 QoS 向量的平均值可以看成是该区域用户的中心 QoS 值,相比于以往的填充方法^[16,30]需要使用聚类中心值,上述方法不需要聚类,计算过程简单且有效.为了平衡原始 QoS 值和填充 QoS 值的影响,我们对 PCC 计算公式进行改编,引入了一个权重系数如式(2)所示:

$$w_{as} = \begin{cases} \theta, & \text{原始值} \\ 1-\theta, & \text{填充值} \end{cases} \quad (2)$$

其中 $\theta \in [0,1]$ 是调整原始 QoS 值和填充 QoS 值影响力的参数. $\theta=1$ 时,意味着仅使用原始的 QoS 值计算相似度,忽略填充的 QoS 值.通过引入该权重参数,我们提出了一种改进的 PCC 计算方法来计算活动用户与其它候选用户之间的相似性:

$$\text{sim}(a, u) = \frac{\sum_{s \in S_a \cap S_u} w_{as} (r_{a,s} - \bar{r}_a) \cdot (r_{u,s} - \bar{r}_u)}{\sqrt{\sum_{s \in S_a \cap S_u} w_{as}^2 (r_{a,s} - \bar{r}_a)^2} \sqrt{\sum_{s \in S_a \cap S_u} (r_{u,s} - \bar{r}_u)^2}} \quad (3)$$

其中, w_{as} 是分配给 $r_{a,s}$ 的权重,它的值取决于 $r_{a,s}$ 是原始的 QoS 值还是填充值.

前面讨论的是如何查找相似用户,而查找相似服务的过程与此类似.设 t 为目标服务,同样可以使用位置相邻的服务 QoS 平均值预先填充它的空缺 QoS 值,并且引入权重参数 w_{ut} 来区分原始 QoS 值和填充 QoS 值.计算服务 t 和另外一个服务 s 之间相似度的公式如下:

$$\text{sim}(s, t) = \frac{\sum_{u \in U_s \cap U_t} w_{ut} (r_{u,t} - \bar{r}_t) \cdot (r_{u,s} - \bar{r}_s)}{\sqrt{\sum_{u \in U_s \cap U_t} w_{ut}^2 (r_{u,t} - \bar{r}_t)^2} \sqrt{\sum_{u \in U_s \cap U_t} (r_{u,s} - \bar{r}_s)^2}} \quad (4)$$

其中 $U_s \cap U_t$ 代表同时调用过服务 s 和 t 的用户集

合; $r_{u,s}$ 为用户 u 对服务 s 观察到的 QoS 值; $\bar{r}_s$ 为服务 s 的平均 QoS 值.为了避免文字重复,对查找相似服务的过程不再赘述.

从以上的描述可以看出,提出的相似邻居计算方法,不但考虑了用户(服务)的网络位置,提高了计算效率和查找准确性,而且能够缓解数据稀疏和冷启动问题,有利于提高服务 QoS 预测性能.

3.3 经典的因子分解机模型

因子分解机^[24]是在传统因子分解模型(也被称作隐因子模型)的基础上提出来的.它的优势在于能够通过特征向量去模拟因子分解模型,既结合了特征工程法(Feature Engineering)的普遍性和适用性,又能够利用因子分解模型对不同类型变量之间的交互作用进行建模估计,能够快速地完成学习任务,取得很好的预测精度.此外,因子分解机能很好地适应稀疏数据环境,具有线性时间复杂度,且具备通用性,可以用于各种预测问题^[28].

从表达形式上来看,因子分解机是在多项式回归模型的基础上去掉了自相关项,保留了交叉项.但是它在交叉项的权重上面引入了因子分解的概念,解决了数据稀疏性带来的参数估计缺陷.同时通过引入因子分解参数,在学习权重的时候会综合考虑各个因变量之间的关系.推荐系统常用的二阶因子分解机模型表达式可定义为

$$\hat{y}(x) = w_0 + \sum_{i=1}^p w_i x_i + \sum_{i=1}^p \sum_{j>i}^p \langle v_i, v_j \rangle x_i x_j \quad (5)$$

模型参数为

$$w_0 \in \mathbb{R}, W \in \mathbb{R}^p, V \in \mathbb{R}^{p \times k} \quad (6)$$

其中, p 是变量的个数, $k \ll p$ 表示因子分解的维度, v_i 表示矩阵 V ($V \in \mathbb{R}^{p \times k}$) 中的向量, w_0 代表全局偏置, w_i 代表第 i 个变量的权重. $\langle v_i, v_j \rangle$ 表示两个向量的内积,可表示如下:

$$\langle v_i, v_j \rangle = \sum_{f=1}^k v_{i,f} v_{j,f} \quad (7)$$

定义了 FM 的模型后,需要从训练集中学习出参数 w_0, W, V . 为方便表述,可以将 FM 模型中的参数统一表示为 $\Theta = \{w_0, w_1, \dots, w_p, v_{1,1}, \dots, v_{p,k}\}$. 类似于其它监督学习算法, FM 模型中的参数值可以通过定义和最优化损失函数来学习:

$$\text{OPT}(S) = \arg \min_{\Theta} \sum_{(x,y) \in S} L(\hat{y}(x|\Theta), y) \quad (8)$$

式(8)的目标是对观测数据集 S 中的每一对 (x, y) , 求出实际观测值 y 与预测值 $\hat{y}(x|\Theta)$ 之间误差之和, 并对其最小化, 以最优化参数集合 Θ . 损失函数 L 的定义如下:

$$L(\hat{y}(x), y) = (\hat{y}(x) - y)^2 \tag{9}$$

FM 模型中包含了大量的参数,为避免过拟合,需要加入正则化项,即

$$Reg(w_0, w_i, v_{i,f}) = \lambda_1 w_0^2 + \lambda_2 \|w_i\|_F^2 + \lambda_3 \|v_{i,f}\|_F^2 \tag{10}$$

其中, $\lambda_1, \lambda_2, \lambda_3$ 表示正则化的系数,加入正则化项后,优化函数变为

$$OPT(S) = \arg \min_{\theta} \sum_{(x,y) \in S} L(\hat{y}(x), y) + Reg(w_0, w_i, v_{i,f}) \tag{11}$$

式中, $\hat{y}(x)$ 是 FM 预测的值, y 是真实的观测值. 为了求解最小化损失函数中的参数,通过引入随机梯度下降法(SGD)进行优化学习. 随机梯度下降方法在进行每一步下降的时候是随机选取一个样本进行梯度计算,公式如下:

$$\frac{\partial}{\partial w_0} L(\hat{y}(x), y) = \hat{y}(x) - y \tag{12}$$

$$\frac{\partial}{\partial w_i} L(\hat{y}(x), y) = (\hat{y}(x) - y)x_i \tag{13}$$

$$\frac{\partial}{\partial v_{i,f}} L(\hat{y}(x), y) = (\hat{y}(x) - y) \left(x_i \sum_{j=1}^n v_{j,f} x_j - v_{i,f} x_i^2 \right) \tag{14}$$

计算出梯度后更新权重,即沿着目标函数梯度下降的方向,进行如下迭代:

$$w_0 \leftarrow w_0 + \eta \frac{\partial}{\partial w_0} L(\hat{y}(x), y) \tag{15}$$

$$w_i \leftarrow w_i + \eta \frac{\partial}{\partial w_i} L(\hat{y}(x), y) \tag{16}$$

$$v_{i,f} \leftarrow v_{i,f} + \eta \frac{\partial}{\partial v_{i,f}} L(\hat{y}(x), y) \tag{17}$$

其中, $\eta > 0$ 为计算时的学习速率,或者理解为下降速度,它的取值会影响模型的收敛速度.

3.4 位置感知的因子分解机

由于 FM 模型具有精确度高和复杂性低的优势,

便于添加上下文信息等内容进行扩充^[25],因此我们结合位置信息和 FM 模型来提高 Web 服务 QoS 预测性能. 不考虑位置信息的 FM 的计算公式可写作如下:

$$\hat{y}(x) = w_0 + w_u + w_i + \langle v_u, v_i \rangle \tag{18}$$

其中, x 代表用户的一次服务调用, y 是服务调用产生的 QoS 值. 请注意,式(18)的右端没有出现变量 x (与式(5)比较),是因为特征向量是由 0/1 构成(参考图 4). 用户 u 调用服务 i 的时候,只有 u 和 i 对应的 x 为 1,其他的都为 0. 同样交叉因子中只有 u 和 i 对应的 x 为 1,所以只有 $\langle v_u, v_i \rangle$ 有值. 我们提出的位置感知的因子分解机将相似用户和相似服务的特征考虑了进来,计算公式定义如下:

$$\hat{y} = w_0 + w_u + w_i + \frac{1}{|N_u|} \sum_{m \in N_u} (w_m + \langle v_u, v_m \rangle + \langle v_i, v_m \rangle) + \frac{1}{|N_i|} \sum_{n \in N_i} (w_n + \langle v_u, v_n \rangle + \langle v_i, v_n \rangle) + \frac{1}{|N_u| |N_i|} \sum_{m \in N_u} \sum_{n \in N_i} \langle v_m, v_n \rangle \tag{19}$$

其中, N_u 为用户 u 的相似用户的集合, N_i 为服务 i 的相似服务集合, $|N_u|$ 和 $|N_i|$ 分别为用户集合和服务集的大小. 公式右端的后面 3 项分别建模了相似用户、相似服务的影响以及它们之间的关系,其中 w_m 和 w_n 分别代表相似用户和相似服务影响的权重, $\langle v_u, v_m \rangle, \langle v_u, v_n \rangle$ 分别建模了用户 u 和相似用户及相似服务的相关性, $\langle v_i, v_m \rangle, \langle v_i, v_n \rangle$ 分别建模了服务 i 和相似用户及相似服务的相关性, $\langle v_m, v_n \rangle$ 建模了相似用户和相似服务的相关性. 这些新增加的项目代表了影响 QoS 预测结果的潜在因素,有利于提升预测结果.

特征向量X				目标向量Y	
X_1	1 0 0 0 ...	1 0 0 0 ...	0 1 1 0 ...	0 0 1 1 ...	2.7 y_1
X_2	1 0 0 0 ...	0 1 0 0 ...	0 0 1 1 ...	1 1 0 1 ...	1.8 y_2
X_3	0 1 0 0 ...	1 0 0 0 ...	1 0 0 1 ...	1 0 1 1 ...	5.6 y_3
X_4	0 1 0 0 ...	0 0 1 0 ...	0 0 1 1 ...	1 0 1 0 ...	0.6 y_4
X_5	0 0 1 0 ...	1 0 0 0 ...	0 1 0 1 ...	0 1 0 1 ...	2.3 y_5
X_6	0 0 1 0 ...	0 0 1 0 ...	1 1 0 0 ...	1 0 0 1 ...	4.4 y_6
X_7	0 0 1 0 ...	0 1 0 0 ...	0 1 0 0 ...	1 0 1 0 ...	0.2 y_7
X_8	0 0 0 1 ...	0 0 0 1 ...	1 0 1 0 ...	0 1 1 0 ...	4.9 y_8
X_9	0 0 0 1 ...	0 1 0 0 ...	1 0 1 0 ...	0 0 1 0 ...	1.8 y_9
	A B C D	S1 S2 S3 S4	A B C D	S1 S2 S3 S4	
	用户	服务	相似用户	相似服务	

图 4 特征向量示例图

下面用一个例子来说明 LAFM 模型. 以预测服务的响应时间为例, 如图 4 所示, 左侧的大框(包含 4 个小框)代表输入的特征向量, 每个特征向量(如 $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_9$)是由一个服务调用记录(即一次用户-服务交互)构造而来. 最右侧的框代表服务调用产生的 QoS 值, 在这里是响应时间. 每个特征向量由 4 类特征变量构成, 除了用户和服务特征, 还包括通过相似邻居计算得到的相似用户和相似服务. 图中 A, B, C, D 表示用户, $S1, S2, S3, S4$ 分别表示服务, 图中每一行表示一个特征向量实例. 以第一行特征向量 $\mathbf{X}_1$ 为例, 左侧的用户类变量值表示对应的是用户 A , 服务类变量值表示对应的是服务 $S1$, 右侧的目标变量值表示用户 A 调用服务 $S1$ 的响应时间为 2.7s , 此外, 从相似用户变量看出用户 A 的相似用户为 B 和 C , 从相似服务变量看出服务 $S1$ 的相似服务为 $S3$ 和 $S4$. 这里, LAFM 的预测问题就是如何将左侧的特征向量准确映射到右侧的响应时间. 如前面所描述的, 求解该问题的关键是估计公式(19)中的各个参数. 通过利用训练集中的每个样本实例, 可以借助 Rendle 提供的开源工具 libFM 来实现因子分解机的学习与预测, 具体实现不再赘述, 读者可参考文献[24].

3.5 Web 服务推荐

使用上述位置感知的因子分解机模型可以为一个用户预测所有候选 Web 服务的 QoS 值. 在预测服务 QoS 的基础上, 就可以向用户推荐高质量的服务. 算法 1 是对本文所提推荐方法 LAFM 的一个高层次的完整描述, 由于前面已给出了各个步骤的计算细节, 这里不再赘述.

算法 1. 基于 LAFM 的 Web 服务推荐.

输入: 原始用户-服务 QoS 矩阵, 用户位置信息, 服务位置信息, 当前测试用户 a

输出: QoS 预测值填充后的 QoS 矩阵, 推荐给 a 的最佳 Web 服务列表

1. 根据 QoS 矩阵和用户位置, 计算每个用户的相似用户集合;
2. 根据 QoS 矩阵和服务位置, 计算每个服务的相似服务集合;
3. 构造并实现位置感知的因子分解机模型, 训练模型参数;
4. 对原始 QoS 矩阵中的缺失 QoS 值用上述模型预测, 得到预测值填充的 QoS 矩阵;
5. 针对当前测试用户 a , 搜索预测后的 QoS 矩阵, 返回 QoS 最佳的 Web 服务列表.

4 算法复杂度分析

FM 能够处理大数据环境下的高维数据, 并且具有线性复杂度, Rendle 在文献[24]中对此进行了论述. 经典的二阶 FM 模型可转换如下:

$$\begin{aligned}
 & \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \\
 &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j - \frac{1}{2} \sum_{i=1}^n \langle \mathbf{v}_i, \mathbf{v}_i \rangle x_i x_i \\
 &= \frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n \sum_{f=1}^k v_{i,f} v_{j,f} x_i x_j - \sum_{i=1}^n \sum_{f=1}^k v_{i,f} v_{i,f} x_i x_i \right) \\
 &= \frac{1}{2} \sum_{f=1}^k \left(\left(\sum_{i=1}^n v_{i,f} x_i \right) \left(\sum_{j=1}^n v_{j,f} x_j \right) - \sum_{i=1}^n v_{i,f}^2 x_i^2 \right) \\
 &= \frac{1}{2} \sum_{f=1}^k \left(\left(\sum_{i=1}^n v_{i,f} x_i \right)^2 - \sum_{i=1}^n v_{i,f}^2 x_i^2 \right) \quad (20)
 \end{aligned}$$

经过转换后, 式(20)具有线性计算复杂度, 简单表示为 $O(kn)$, 依赖于因子分解的维数 k , 以及特征因子的个数 n . 由于 k 一般可以取常数, 所以可认为 FM 具有线性复杂度. 根据式(19)分析 LAFM 的时间复杂度, 主要是由最后 3 项引起的. 根据式(20)很容易得出, 式(19)等号右边第 3 项和第 4 项的时间复杂度分别为 $O(|N_u| \times kn)$ 和 $O(|N_i| \times kn)$, 而第 5 项的时间复杂度为 $O(|N_u| \times |N_i| \times kn)$, 所以 LAFM 的时间复杂度可表示为 $O(|N_u| \times |N_i| \times kn)$. 因为 $|N_u|$ 和 $|N_i|$ 不超过 Top-K 的值, 可以取常数, 所以 LAFM 的时间复杂度最后也可以认为是线性的. 当然, 需要说明的是, 我们的方法在使用 FM 进行预测之前, 需要为每个用户和服务查找相似邻居, 这也需要一定的时间代价. 但是由于我们优先在本地搜索相似用户(服务), 而不是搜索全部用户(服务), 能够大幅降低这方面的时间代价. 以上复杂度分析表明, 我们的方法 LAFM 能高效处理大规模的数据.

5 实验

5.1 数据集

为了验证本文方法的有效性, 我们采用了从 wsdream.com 网站上获得的真实 Web 服务 QoS 数据集来进行实验评估. 该数据集最早发布于文献[31-32], 包含 1974675 条 QoS 记录, 这些记录是通过 339 个计算机用户对 5825 个 Web 服务进行调用得到的. 据分析, 这 339 个用户是分布在 137 个自治

系统和 31 个国家中,5825 个服务是分布在 1021 个自治系统和 74 个国家中,每个用户和每个 Web 服务之间都有一条通过调用产生的 QoS 记录,记录中主要包含响应时间(Response Time 或 Round Trip Time,简称 RTT)、吞吐量(Throughput)等服务质量参数.本文实验使用服务的 RTT 数据来评估我们提出的方法.通过抽取每条 QoS 记录中的 RTT 数据,我们得到 339×5825 的 RTT 矩阵.该 RTT 矩阵进一步划分为训练数据和测试数据,实验中使用训练数据求出服务和用户的相似邻居,以及 LAFM 模型的参数,实验结果是通过比较测试数据的 RTT 值与预测 RTT 值得出.

5.2 评估指标

我们通过计算预测的 RTT 值与实际的 RTT 值之间的偏差,来评估本文方法预测 Web 服务 RTT 的准确性.为此,我们采用两个常用的误差度量指标:平均绝对误差 MAE(Mean Absolute Error)和均方根误差 RMSE(Root Mean Squared Error).这两个指标的计算公式如下:

$$MAE = \frac{\sum_{u \in T} |r_{u,i} - \hat{r}_{u,i}|}{|T|} \tag{21}$$

$$RMSE = \sqrt{\frac{\sum_{u \in T} (r_{u,i} - \hat{r}_{u,i})^2}{|T|}} \tag{22}$$

其中, $r_{u,i}$ 是用户 u 在服务 i 上观测到的实际 RTT 值, $\hat{r}_{u,i}$ 是预测的 RTT 值, T 是测试数据集, $|T|$ 是测试集的大小.MAE 和 RMSE 的值越小,代表预测的精度越高.

5.3 预测性能对比

为了比较 LAFM 方法与其它方法在预测性能上的优劣,实验中同时测试了下面几个较经典的协同过滤式预测方法:

(1)UPCC. 基于用户的协同过滤算法,先使用 PCC 计算用户间的相似度,然后使用当前用户的其它相似邻居在目标服务上的观测值来为当前用户预

测目标服务的 QoS 值.

(2)IPCC. 基于物品的协同过滤算法,先使用 PCC 计算 Web 服务之间的相似度,然后使用目标服务的其他相似邻居的 QoS 值来为当前用户预测 QoS 值.

(3)UIPCC. UIPCC^[4]是 UPCC 和 IPCC 方法的结合,通过将这两种方法的预测值综合起来进行预测.

(4)NMF. NMF^[33]是在保持非负性约束下一种矩阵分解方法.由于在很多情况下非负性假设是合理的,因此非负矩阵分解被越来越多地应用到推荐系统中.

(5)PMF. PMF^[34]是在传统矩阵分解方法的基础上添加了概率分布,利用贝叶斯推导用户和物品的隐式特征的后验概率,最后与矩阵分解相结合进行最终的推荐.

(6)FM. FM^[24]是经典的因子分解机模型,没有添加用户和服务的位置信息.

在比较实验时,为了模拟实际的 Web 服务应用场景,随机移除 RTT 矩阵中一部分数据来稀疏化数据,得到矩阵密度为 5%,10%,15%,20% 四种情况.稀疏化后的矩阵作为训练数据,而被移除的 RTT 数据作为测试数据.使用相同的训练集和测试集对上述方法及我们的方法进行对比实验.实验中防止过度拟合的正则化参数设置为 $\lambda_1 = 80, \lambda_2 = 0.08, \lambda_3 = 0.03$,设置最大相似邻居数量 Top-K = 20,因子分解维度 $k = 100$.

表 2 显示了 LAFM 及各种对比方法的预测结果.可以看到,无论在何种矩阵密度下,LAFM 的 MAE 和 RMSE 值均要小于其他的方法,表明该方法的预测精度更高.而 UPCC、IPCC 和 UIPCC 三种基于记忆的协同过滤方法的预测性能最低,特别是在矩阵密度较低时,说明它们不适用于数据稀疏的推荐环境,主要原因是它们对数据的利用程度不高.NMF 和 PMF 利用矩阵分解理论对每次用户-服务

表 2 RTT 预测性能对比(MAE 和 RMSE 的值越小代表预测精度越高)

方法	Matrix Density=5%		Matrix Density=10%		Matrix Density=15%		Matrix Density=20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UPCC	0.9472	1.9466	0.8775	1.7550	0.7191	1.6628	0.7220	1.6182
IPCC	0.8344	1.8269	0.7362	1.6288	0.6365	1.5463	0.6109	1.5016
UIPCC	0.7652	1.7224	0.7003	1.5179	0.6029	1.4304	0.5512	1.3936
NMF	0.5959	1.5224	0.5783	1.4466	0.5698	1.3984	0.5495	1.3658
PMF	0.5746	1.4622	0.5511	1.3323	0.5295	1.2269	0.4816	1.2197
FM	0.5380	1.3690	0.4826	1.2386	0.4565	1.1815	0.4365	1.1435
LAFM	0.5136	1.3045	0.4498	1.2069	0.4259	1.1479	0.3999	1.1086

调用进行了建模,提高了数据利用率. 相对来说,FM 的预测性能要优于其他传统的协同过滤模型和矩阵分解方法. 与 FM 相比,LAFM 由于加入了服务和用户位置的影响,因此进一步提高了 QoS 预测的精度. 从表 2 中还可以观察到,当矩阵密度较小时,不论何种方法,MAE 和 RMSE 均有所变大,意味着,数据稀疏的确对基于协同过滤的 QoS 预测方法有较大影响. 矩阵密度具体如何影响 QoS 预测,还将在后面进行评估.

5.4 维度对预测性能的影响

本文方法中,维度(k)决定了用于确定特征向量

中各变量之间交叉作用权重的因子的个数. 实验中设置正则化参数 $\lambda_1=80, \lambda_2=0.08, \lambda_3=0.03$, 最大相似邻居 $\text{Top-K}=20$, 矩阵的密度取 5% 或 10%. 图 5 显示了维度 k 对预测精度的影响. 随着 k 的增加,MAE 和 RMSE 的值先快速减少,然而当 k 增大到 200 以后,MAE 和 RMSE 值的改变逐渐不明显. 实际应用中维度(k)的取值不宜过小或者过大,因为过小的维度意味着使用的特征因子个数太少,不利于提高预测的性能,上述实验结果证明了这一点;而过大的维度意味着算法的复杂度也会提高,降低了算法的运行性能.

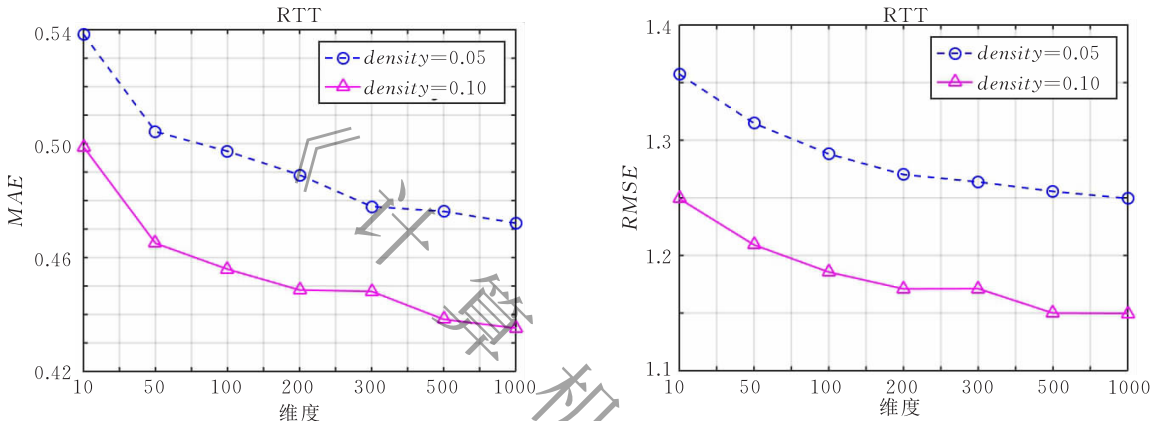


图 5 维度对预测性能的影响

5.5 矩阵密度对预测性能的影响

矩阵密度代表了训练数据的稀疏程度. 本实验旨在评估矩阵密度对预测性能的影响. 实验中,我们设置正则化参数 $\lambda_1=80, \lambda_2=0.08, \lambda_3=0.03$, 最大相似邻居数量 $\text{Top-K}=20$, 然后变化矩阵密度,使之从 2% 逐步增长到 20%, 每步增长 2%. 从图 6 看到,MAE 和 RMSE 的值在矩阵密度增加的起初阶

段迅速减少,意味着预测精度有较快的提高. 当矩阵密度持续增加到一定值时,如超过 10% 时,MAE 和 RMSE 的值下降速度逐渐变得不明显. 这些观察结果表明,适当增加数据密度可以迅速提高预测精度,因此现实中可以通过收集更多的 QoS 记录数据来提高预测精度.

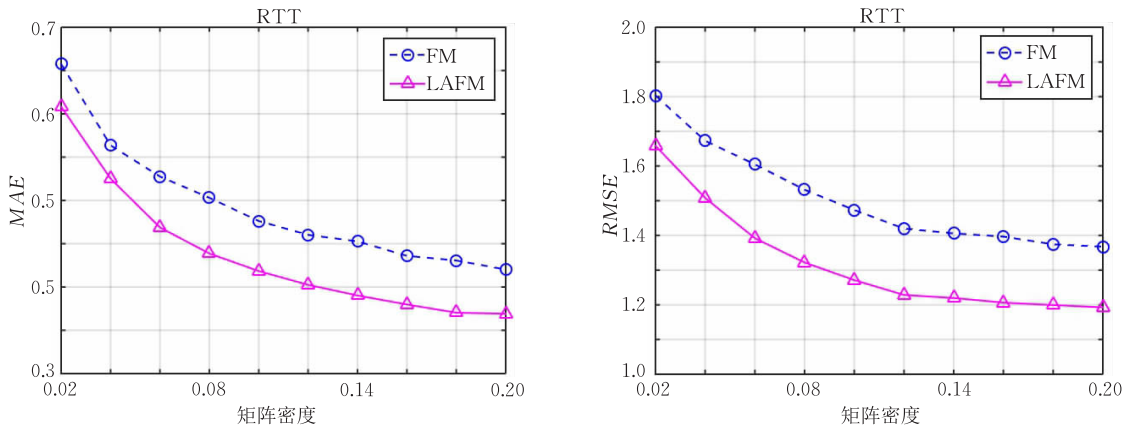


图 6 矩阵密度对预测性能的影响

5.6 参数 Top-K 对预测性能的影响

Top-K 参数决定每个用户或 Web 服务的相似

邻居集合的大小. 为了评估它的影响,实验中将 Top-K 值从 5 逐步增加到 30, 步长为 5. 从图 7 可

以观察到,初始时,随着 Top-K 值的增加,MAE 和 RMSE 的值减少很明显,意味着预测精度有明显提高.但是当 Top-K 超过 20 时,MAE 和 RMSE 的值变化不明显或者甚至有所上升.这个现象说明,参数 Top-K 取适当的值有利于提高预测精度,原因是:

过小的 Top-K 值导致相似邻居集合过小,可能会遗漏掉一些有用的相似邻居;而过大的 Top-K 取值,导致相似邻居集合过大,可能会包含了相似程度低的用户,这些用户的 QoS 信息实际上是噪声.这两种情况都会影响预测性能.

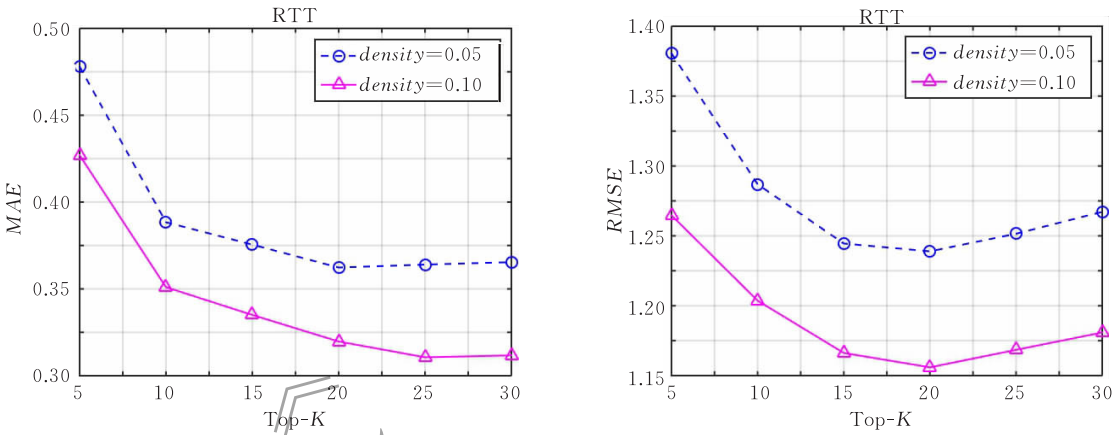


图 7 Top-K 对预测性能的影响

5.7 参数 θ 对预测性能的影响

权重参数 θ 是在为用户或者服务查找相似邻居时引入的,决定了原始 QoS 值和填充 QoS 值在相似度计算中的作用大小.为了评估参数 θ 的影响,我们分别在矩阵密度为 5% 和 10% 两种情况下进行了对比实验.实验中设置正则化参数 $\lambda_1 = 80, \lambda_2 =$

$0.08, \lambda_3 = 0.03$,相似邻居数量上限 $\text{Top-K} = 20, \theta$ 的值从 0 逐步增长到 1.0,步长为 0.1. 从图 8 可以观察到,初始时,MAE 和 RMSE 的值随着 θ 增加而降低.但是当 θ 超过 0.7 时,MAE 和 RMSE 的值有所上升.这个现象也说明了恰当的 θ 值有利于提高预测的精度.

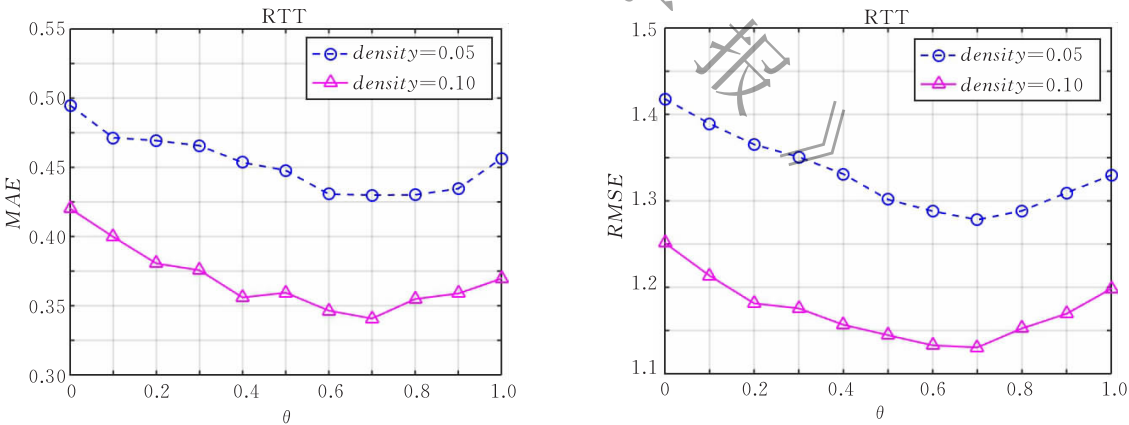


图 8 θ 对预测性能的影响

5.8 正则化参数对预测性能的影响

正则化参数主要控制正则化项对目标函数的影响程度,其中参数 λ_1 控制全局偏差的正则化, λ_2 控制权重向量的正则化, λ_3 控制 QoS 矩阵的正则化.为了评估正则化参数对预测精度的影响,实验中设置因子分解维度 $k = 100, \text{Top-K} = 20$,矩阵密度分别

取 5% 和 10%. 从图 9~图 11 可以观察到,上述参数的取值也会在一定程度上影响预测的精度. MAE 和 RMSE 随着正则化参数值的增加起初有所降低,但是随着这些参数值的持续增加,MAE 和 RMSE 又有上升的趋势.当 $\lambda_1 = 80, \lambda_2 = 0.08, \lambda_3 = 0.03$ 时,我们的方法差不多可以取得最优的预测精度.

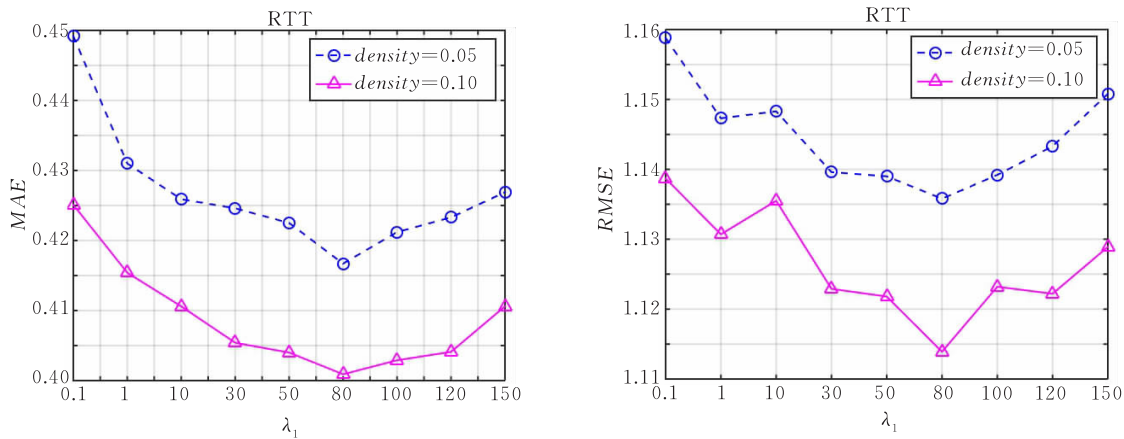


图 9 λ_1 对预测性能的影响

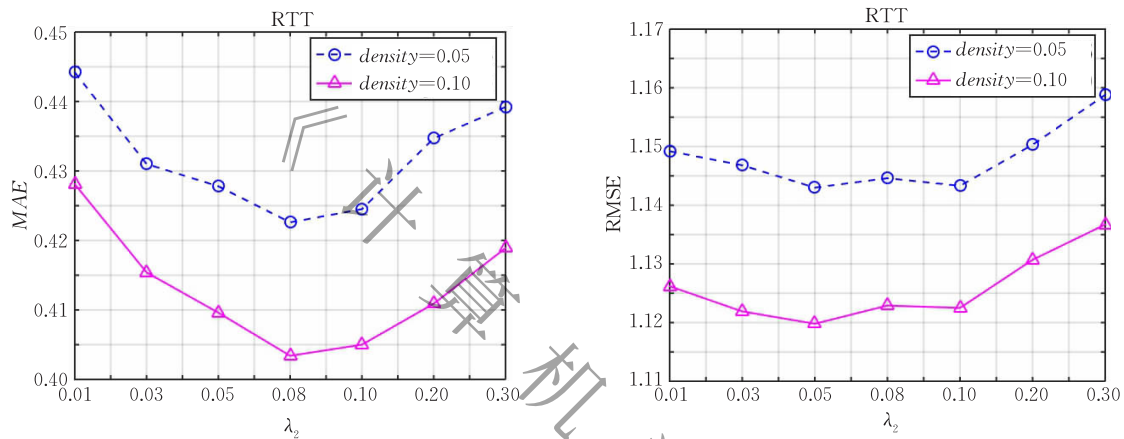


图 10 λ_2 对预测性能的影响

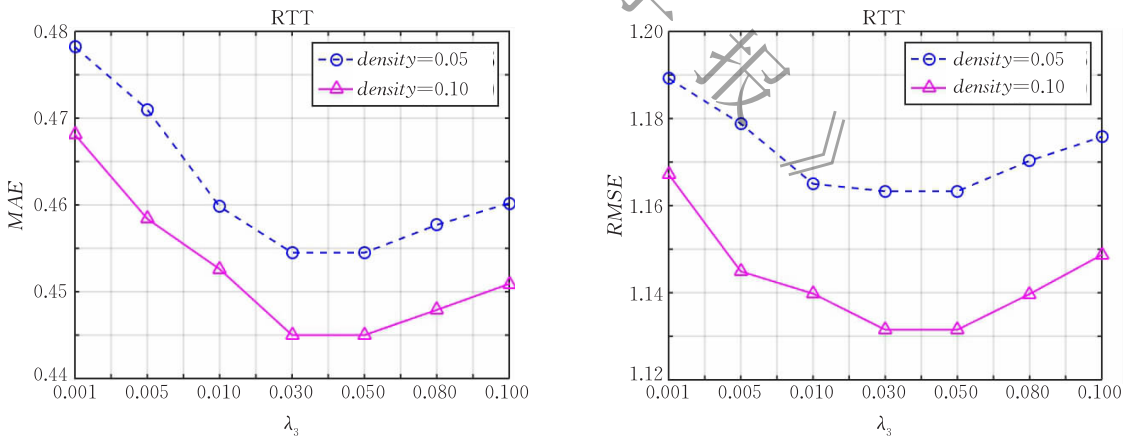


图 11 λ_3 对预测性能的影响

6 总 结

本文针对基于 QoS 的 Web 服务推荐,在经典因子分解机的基础上,通过添加位置信息提出了一种位置感知的因子分解机模型和 Web 服务 QoS 预测方法.该方法先利用用户和 Web 服务的网络位置信息,来为当前用户(服务)查找相似邻居;然后将当

前用户、当前服务与相似邻居的关联性结合到因子分解机中,从而为用户预测他未调用过的 Web 服务的 QoS 值,最终能够向用户推荐具有较高 QoS 值的服务.实验结果表明,通过使用因子分解机模型和引入位置信息,QoS 预测性能与以往的方法相比有一定提高.而且因子分解机具有近乎线性的预测时间复杂度,不仅能解决数据稀疏、冷启动等问题,同时还解决了传统协同过滤算法时间效率较低的问题.

致 谢 作者非常感谢审稿专家为提高本文质量提出的建设性意见,同时要特别感谢编辑部老师为本文的发表所做的全部工作!

参 考 文 献

- [1] Wang Hai-Yan, Yang Wen-Bin, Wang Sui-Chang, Li Si-Rui. A service recommendation method based on trustworthy community. *Chinese Journal of Computers*, 2014, 37(2): 301-311(in Chinese)
(王海艳, 杨文彬, 王随昌, 李思瑞. 基于可信联盟的服务推荐方法, 计算机学报, 2014, 37(2): 301-311)
- [2] Shao L, Zhang J, Wei Y, et al. Personalized QoS prediction for Web services via collaborative filtering//*Proceedings of the 2007 IEEE International Conference on Web Services*. Salt Lake City, USA, 2007: 439-446
- [3] Shao Ling-Shuang, Zhou Li, Zhao Jun-Feng, et al. Web service QoS prediction approach. *Journal of Software*, 2009, 20(8): 2062-2074(in Chinese)
(邵凌霜, 周立, 赵俊峰等. 一种 Web Service 的服务质量预测方法. 软件学报, 2009, 20(8): 2062-2074)
- [4] Zheng Z, Ma H, Lyu M R, et al. WSRec: A collaborative filtering based Web service recommender system//*Proceedings of the 7th International Conference on Web Services*. Los Angeles, USA, 2009: 437-444
- [5] Chen X, Liu X D, Huang Z C, et al. RegionKNN: A scalable hybrid collaborative filtering algorithm for personalized Web service recommendation//*Proceedings of the 8th International Conference on Web Services*. Miami, USA, 2010: 9-16
- [6] Zheng Z, Ma H, Lyu M R, et al. Collaborative Web service QoS prediction via neighborhood integrated matrix factorization. *IEEE Transactions on Services Computing*, 2013, 6(3): 289-299
- [7] He P, Zhu J, Zheng Z, et al. Location-based hierarchical matrix factorization for Web service recommendation//*Proceedings of the 21st International Conference on Web Services*. Alaska, USA, 2014: 297-304
- [8] Lo W, Yin J, Li Y, et al. Efficient Web service QoS prediction using local neighborhood matrix factorization. *Engineering Applications of Artificial Intelligence*, 2015, 38: 14-23
- [9] Tang M, Jiang Y, Liu J, Liu X F. Location-aware collaborative filtering for QoS based Web service recommendation//*Proceedings of the 19th International Conference on Web Services*. Hawaii, USA, 2012: 202-209
- [10] Liu J, Tang M, Zheng Z, et al. Location-aware and personalized collaborative filtering for Web service recommendation. *IEEE Transactions on Services Computing*, 2016, 9(5): 686-699
- [11] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 2005, 17(6): 734-749
- [12] Herlocker J, Konstan J A, Riedl J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 2002, 5(4): 287-310
- [13] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms//*Proceedings of the 10th International Conference on World Wide Web*. Hong Kong, China, 2001: 285-295
- [14] Yu Q, Zheng Z, Wang H. Trace norm regularized matrix factorization for service recommendation//*Proceedings of the 20th International Conference on Web Services*. Santa Clara, USA, 2013: 34-41
- [15] Jiang Y, Liu J, Tang M, et al. An effective Web service recommendation method based on personalized collaborative filtering//*Proceedings of the 9th International Conference on Web Services*. Washington, USA, 2011: 211-218
- [16] Wu J, Chen L, Feng Y, et al. Predicting quality of service for selection by neighborhood-based collaborative filtering. *IEEE Transactions on Systems, Man, and Cybernetics*, 2013, 43(2): 428-439
- [17] Chen Lei, Yang Geng, Chen Zheng-Yu, et al. Web service QoS prediction via matrix completion with structural noise. *Journal on Communications*, 2015, 36(6): 49-59(in Chinese)
(陈蕾, 杨庚, 陈正宇等. 基于结构化噪声矩阵补全的 Web 服务 QoS 预测. 通信学报, 2015, 36(6): 49-59)
- [18] Ma Y, Wang S, Hung P C K, et al. A highly accurate prediction algorithm for unknown Web service QoS value. *IEEE Transactions on Services Computing*, 2015, 9(4): 511-523
- [19] Chen X, Zheng Z, Liu X, et al. Personalized QoS-aware Web service recommendation and visualization. *IEEE Transactions on Services Computing*, 2013, 6(1): 35-47
- [20] Lo W, Yin J, Deng S, et al. Collaborative Web service QoS prediction with location-based regularization//*Proceedings of the 19th International Conference on Web Services*. Hawaii, USA, 2012: 464-471
- [21] Song Zhi-Bo. Web Service Recommendation Based on the Network Location [M. S. dissertation]. Hangzhou Dianzi University, Hangzhou, 2014(in Chinese)
(宋智波. 基于网络位置的 Web 服务推荐技术研究[硕士学位论文]. 杭州电子科技大学, 杭州, 2014)
- [22] Xu J, Zheng Z, Lyu M R. Web service personalized quality of service prediction via reputation-based matrix factorization. *IEEE Transactions on Reliability*, 2016, 65(1): 28-37
- [23] Wang S, Ma Y, Cheng B, et al. Multi-dimensional QoS prediction for service recommendations. *IEEE Transaction on Services Computing*, 2016, Doi: 10.1109/TSC.2016.2584058
- [24] Rendle S. Factorization machines//*Proceedings of the 10th IEEE International Conference on Data Mining*. Sydney, Australia, 2010: 995-1000
- [25] Rendle S. Factorization machines with libFM. *ACM Transactions on Intelligent Systems and Technology*, 2012, 3(3): 57
- [26] Rendle S, Gantner Z, Freudenthaler C, et al. Fast context-aware recommendations with factorization machines//*Proceedings of the 34th ACM SIGIR Conference on Research and Development in Information Retrieval*. Beijing, China, 2011: 635-644
- [27] Oentaryo R J, Lim E P, Low J W, et al. Predicting response in mobile advertising with hierarchical importance-aware factorization machine//*Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. New

York, USA, 2014: 123-132

[28] Yu Qiu-Hong. Relationship Recommendation in Social Networks Based on Factorization Machines[M.S. dissertation]. Beijing University of Posts and Telecommunications, Beijing, 2013(in Chinese)
(余秋宏. 基于因子分解机的社交网络关系推荐研究[硕士学位论文]. 北京邮电大学, 北京, 2013)

[29] Zheng Z, Wu X, Zhang Y, et al. QoS ranking prediction for cloud services. *IEEE Transactions on Parallel and Distributed Systems*, 2013, 24(6): 1213-1222

[30] Xue G R, Lin C, Yang Q, et al. Scalable collaborative filtering using cluster-based smoothing//*Proceedings of the 28th ACM SIGIR Conference on Research and Development in Information Retrieval*. Salvador, Brazil, 2005: 114-121

[31] Zheng Z, Zhang Y, Lyu M R. Investigating QoS of real-world Web services. *IEEE Transactions on Services Computing*, 2014, 7(1): 32-39

[32] Zheng Z, Zhang Y, Lyu M R. Distributed QoS evaluation for real-world Web services//*Proceedings of 2010 IEEE International Conference on Web Services*. Miami, USA, 2010: 83-90

[33] Lee D D, Seung H S. Algorithms for non-negative matrix factorization//*Proceedings of the 2001 Conference on Neural Information Processing Systems*. Vancouver, Canada, 2001: 556-562

[34] Salakhutdinov R, Mnih A. Probabilistic matrix factorization//*Proceedings of the 2011 Conference on Neural Information Processing Systems*. Granada, Spain, 2011: 880-887



TANG Ming-Dong, born in 1978, Ph.D., professor. His current research interests include services computing, software engineering and data mining.

ZHANG Ting-Ting, born in 1992, M.S. Her current research interests focus on service computing.

YANG Ya-Tao, born in 1992, M.S. His current research interests focus on service computing.

ZHENG Zi-Bin, born in 1982, Ph.D., associate professor, Ph.D. supervisor. His current research interests include service computing and big data analysis.

CAO Bu-Qing, born in 1979, Ph.D., associate professor. His current research interests include service computing and service-oriented software engineering.

Background

Web services have become the mainstream components for building large-scale and distributed software applications. To build reliable service-based software applications, it is critical to use high-quality Web services. However, the quality of services (QoS) is usually unknown to a Web service user, for the QoS values offered by service providers are missing in most cases. Moreover, QoS values of Web services are typically related to specific users, i. e., different users are likely to perceive different QoS values on the same Web service. In this situation, personalized Web service QoS prediction has become an important issue, and has raised a lot of interests from the scholars in Web service evaluation and recommendation.

Aiming at discovering and recommending high-quality Web services for a user, a dozen of studies have been carried out to address the above Web service QoS prediction problem. Most of the state-of-the-art approaches employ collaborative filtering techniques to predict Web service QoS values, and they are typically divided into two categories: memory-based and model-based. Traditional collaborative filtering techniques, however, usually suffer from the difficulties such as data sparseness and cold start issues, which probably lead to inaccurate prediction results in practice, and their time efficiency is usually quite low. Though, new collaborative

filtering techniques, such as matrix factorization, have been proposed to tackle these issues, there is still much room for improvement. For example, matrix factorization models usually have square time complexity for its model equation computation, which may cause inefficiency for large-scale recommender systems. Moreover, adding extra features to matrix factorization models is yet a non-trivial task.

This paper proposes a QoS-aware Web service recommendation approach using a location-aware factorization machine model. The factorization machine model has shown its greater effectiveness and efficiency in prediction, compared with the other collaborative filtering techniques. This work is the first to introduce it to personalized Web service QoS prediction. Furthermore, this work takes advantage of the locations of users and Web services, and integrates them with the classic factorization machine, so that the QoS prediction performance can be further improved. The experiments using a real Web service dataset validate the proposed approach. This work is partly supported by the National Natural Science Foundation of China under Grant No.61572186, No.61572188 and No.61402168. QoS-aware Web service recommendation is among the key research issues of these projects, and this work is intended to address it.