软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(11):3374−3387 [doi: 10.13328/j.cnki.jos.005482]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

融合多维信息的主题自适应 Web API 推荐方法*

李鸿超, 刘建勋, 曹步清, 石 敏

(知识处理与网络化制造湖南省普通高校重点实验室(湖南科技大学),湖南 湘潭 411201)

通讯作者: 刘建勋, E-mail ljx529@gmail.com; 曹步清, E-mail:buqingcao@gmail.com

摘 要: 如何根据用户的自然语言需求描述自动生成或推荐用于解决问题的 Web API 服务集合,并辅助构建 Mashup,是业务流程管理者和服务组合者关注的热点之一.如何提高推荐的质量,是大家关注的焦点.为此,提出了一种融合多维信息的主题自适应 Web API 推荐方法 HDP-FM(hierarchical Dirichlet processes-factorization machines)为 Mashup 的创建推荐 Web APIs 集合.该方法以 Web API 的描述文档为语料库,利用 HDP 模型训练每个 Web API 的主题分布向量;其次,利用已生成的主题模型预测每个 Mashup 的主题分布向量,用于相似度的计算;最后,将 Mashup 之间的相似度、WebAPI 之间的相似度、Web API 的流行度和共现性作为因子分解机模型的输入,评分排序获取用于推荐的 Web APIs 集合.为了验证 HDP-FM 方法的性能,使用从 ProgrammableWeb 平台上爬取的真实数据进行多组实验,实验结果表明,HDP-FM 方法在准确率、召回率、*F*-measure 和 *NDCG@N* 等方面具有较好的性能.

关键词: Web API 推荐;HDP(hierarchical Dirichlet process);因子分解;Mashup 创建

中图法分类号: TP311

中文引用格式: 李鸿超,刘建勋,曹步清,石敏.融合多维信息的主题自适应 Web API 推荐方法.软件学报,2018,29(11):3374−3387. http://www.jos.org.cn/1000-9825/5482.htm

英文引用格式: Li HC, Liu JX, Cao BQ, Shi M. Topic-Adaptive Web API recommendation method via integrating multidimensional information. Ruan Jian Xue Bao/Journal of Software, 2018,29(11):3374−3387 (in Chinese). http://www.jos.org.cn/1000-9825/5482.htm

Topic-Adaptive Web API Recommendation Method via Integrating Multidimensional Information

LI Hong-Chao, LIU Jian-Xun, CAO Bu-Qing, SHI Min

(Key Laboratory of Knowledge Processing & Networked Manufacturing (Hu'nan University of Science and Technology), Xiangtan 411201, China)

Abstract: How to automatically generate or recommend a set of Web APIs for Mashup creation according a user's natural language description of requirement is a focus of attention among business process managers and services composition designers. A topic adaptive Web API recommendation method, HDP-FM (hierarchical Dirichlet processes-factorization machine), is proposed in this paper to recommend a set of Web APIs for Mashup creation. This approach firstly makes the Web API description document as a corpus, and trains a topic distribution vector for a Web API by the HDP model. It then predicts a topic distribution vector for a Mashup via the generated model, where the topic distribution vector is used to calculate the similarity. Finally, a factorization model is utilized to score and sort Web APIs by taking the similarity between Mashups, the similarity between Web APIs, the popularity of Web APIs and the co-occurrence of

* 基金项目: 国家自然科学基金(61872139, 61873316, 61572187); 国家科技支撑计划(2015BAF32B01); 湖南省自然科学基金(2017JJ2098)

Foundation item: National Natural Science Foundation of China (61872139, 61873316, 61572187); National Key Technology R&D Program of China (2015BAF32B01); Hu'nan Provincial Natural Science Foundation (2017JJ2098)

本文由面向智能制造的业务过程管理与服务技术专题特约编辑王建民教授、刘建勋教授推荐.

收稿时间: 2017-07-20; 修改时间: 2017-09-16; 采用时间: 2017-11-14; jos 在线出版时间: 2017-12-06

CNKI 网络优先出版: 2017-12-06 15:37:40, http://kns.cnki.net/kcms/detail/11.2560.TP.20171206.1537.028.html

Web APIs as inputs. A Mashup can be created based on these recommended Web APIs. To verify the performance of the HDP-FM method, a series of experiments are conducted on a real dataset crawled from the ProgrammableWeb platform. The results show that the HDP-FM method has a good performance over others in term of precision, recall, *F*-measure and *NDCG@N*.

Key words: Web API recommendation; HDP (hierarchical Dirichlet process); factorization machine; Mashup creation

Web 服务是一种以服务为导向的架构技术,该技术常被用来完成分布式和异构系统之间的自动化交互或链接业务流程.然而,功能单一的 Web 服务很难满足一些复杂多变的需求.为解决这一问题,一种区别于传统资源集成方案的新企业级应用开发技术 Mashup 被提出.该技术能够集成单一功能的服务(Web API 服务:使用 REST 风格、HTTP 协议、JSON 数据格式;可通过互联网使用的应用程序接口;具有易访问、可扩展、易开发与组合等诸多优点),构建多功能服务应用以适应用户的复杂请求.随着 Mashup 技术的广泛使用,出现了许多 Mashup 服务平台(如 ProgrammableWeb,IBM Mashup Center,Yahoo Pipe 等)以提供种类繁多的 Web API.在平台上,用户可以根据自身的需求选择性地调用 Web API 来创建满足相应需求的 Mashup 应用.然而,网络上发布的 Web API 服务越来越多(以 ProgrammableWeb 平台为例,截止 2016 年 12 月,已发布超过 15 500 个 Web API 服务接口),加之 Web API 描述文档非结构化,许多 Web API 功能相似但性能差异较大等一系列问题,使得从 Web API 服务库中选取开发者感兴趣的、适合的、高质量的 Web API 来构建 Mashup 应用变得越来越困难.

因此,如何根据 Mashup 构建者的自然语言表述的需求自动生成或推荐一个解决该问题的完整方案,是研究者自然而然的理想.但是在目前的人工智能的水平之下,想要完全实现这一目标有很大难度.一个可行的次优方案是“根据自然语言需求描述推荐可行的或者推荐可用于解决该问题的 Web API 任务集合以辅助用户构建 Mashup”.该解决思路目前倍受服务计算与业务过程管理领域研究人员的关注.

图 1 分别展示了开发者 1 所采用的传统的服务搜索方式以及开发者 2 所采用的“推荐可用于解决该问题的任务集合”方案下的 Mashup 创建的流程.开发者 1 根据需求描述,利用 Mashup 平台检索满足需求的特定功能的 Web API 来组建 Mashup 应用,然而 Mashup 平台上的检索功能仅仅是对用户需求的关键词的简单匹配,这会造成检索结果过于庞杂,包含大量符合功能需求的 Web APIs 和一些不符合功能需求的 Web APIs,缺少部分符合功能需求的 Web APIs,例如,检索的关键词为“image”,检索结果多达 1 121 个 Web APIs,一些具有该功能但描述为“picture”“photo”“album”的 Web APIs 没有被检索到,致使需要耗费大量的精力筛选服务质量高且具有组合关系的 Web APIs.而开发者 2 利用 Web API 推荐系统来获取解决问题的 Web API 服务集合,该系统能够利用除关键词以外的多维信息来理解 Mashup 需求文档,进而自动获取开发者感兴趣的、适合的、高质量的 Top-*N* Web API 服务集合,这一方案简化了 Mashup 构建流程,降低了 Mashup 开发难度.

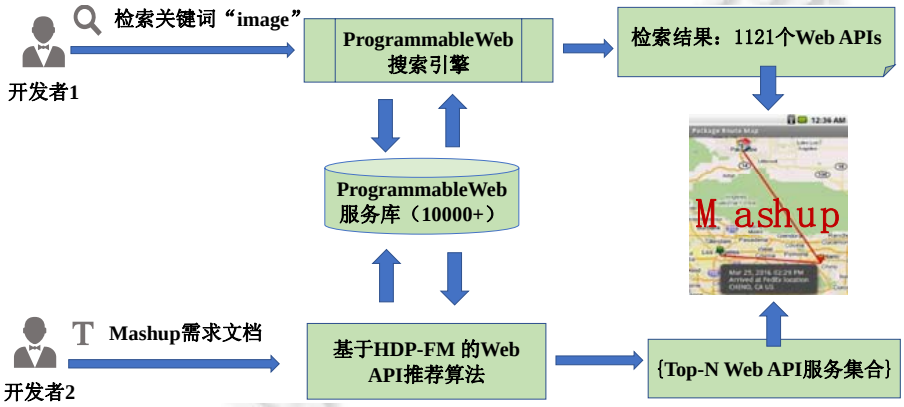


Fig.1 Two different processes for Mashup creation

图 1 两种不同 Mashup 创建流程

本文以此为动机,聚焦于“推荐用于解决问题的 Web API 服务集合以构建 Mashup”,提出一种融合多维信息

的主题自适应 Web API 推荐方法 HDP-FM.该方法首先利用 Hierarchical Dirichlet Processes(HDP)模型挖掘 Web API 描述文本的隐含主题;接着,利用已训练好的 HDP 模型预测 Mashup 描述文本的主题分布向量,用于相似度计算;最后,将 Mashup 与 Mashup 的相似度、Web API 与 Web API 的相似度、Web API 的流行度、Web API 的共现性作为因子分解模型的输入,来推荐 Top-*N* Web APIs 列表供目标 Mashup 创建者使用.

本文的主要创新点有如下几点.

(1) 利用 HDP 模型对 Web API 描述文档进行训练,获得隐含主题分布向量,并利用已生成的 HDP 模型预测 Mashup 主题分布向量;

(2) 采用因子分解模型,融合相似度、流行度、共现性等多维信息特征,实现 Web API 的 Top-*N* 评分推荐;

(3) 本文实验数据集使用从 ProgrammableWeb 平台上爬取的真实数据,一系列的实验结果表明,我们的方法具有较好的效用和效率.

本文第 1 节分析 Web API 推荐领域的相关工作.第 2 节对 HDP 模型进行介绍.第 3 节详细阐述融合多维信息的主题自适应 Web API 推荐方法.第 4 节介绍对比方法,给出实验结果,并对实验结果进行分析.最后一节对全文进行总结.

1 相关工作

随着应用 Web API 构建企业级 Mashup 应用愈来愈受到重视,目前 Web 服务领域已引入了许多推荐方法,并已经显示出其能够帮助用户找到有用的信息.纵观 Web 服务推荐研究领域中的方法,总体来说可分为如下 3 类:基于功能特性的 Web 服务推荐、基于非功能特性的 Web 服务推荐、混合特性的 Web 服务推荐.

(1) 基于功能特性的 Web 服务推荐.

该类型的推荐方法主要通过测量 Web 服务描述文档之间的相似度,以此来推荐最相似的 Web 服务来满足用户的需求[1−4].在我们以前的工作中[1],通过使用 Term Frequency-Inverse Document Frequency(TF-IDF)技术分析服务文本(WSDL 或 Web API 功能描述)以提高推荐的精度,该方法通过计算文本在词向量空间上的距离来测量文本之间的相似度.主题模型技术能够获取 Web 服务描述文档的潜在主题分布向量,进而挖掘 Mashup 描述文档与 Web API 描述文档之间的潜在语义关系.因此,一些研究者探索使用主题模型(如 latent Dirichlet allocation,简称 LDA[2])来进一步提高服务推荐的精度[3,4].Li 等人[3]利用 LDA 主题模型从 Web 服务的 WSDL 描述文档中获取 Web 服务隐含的功能特征.Chen 等人[4]利用 LDA 主题模型聚合标签数据与 WSDL 描述文档来提高服务发现的精确度.上述方法均利用相似度计算方法来计算用户需求与服务文档之间的匹配度(常用的相似度计算方法有 Jaccard 公式、余弦相似度公式等).然而传统的主题模型均存在这样一个限制,即,在训练生成描述文档的隐含主题向量分布前需要预先指定主题的个数.更重要的是,主题数会严重的影响最终的服务推荐的效果.通常的做法是不断调节主题数以寻找最佳的主题数,而这一过程需要反复多次训练主题模型.为了消除这一限制,本文探索使用 HDP 模型来获取 Web 服务的主题分布向量以推荐 Web APIs 集合给目标 Mashup.由 Teh 等人在 2004 年提出的 HDP 模型[5]是 LDA 模型的非参数模型推广,其为实现多文档之间共享无限多个聚类提供了解决途径.与传统的参数模型相比,HDP 模型的使用更加灵活,特别是应用于聚类问题时,该模型能够自动确定聚类数目和生成聚类中心的分布.正是由于 HDP 模型具有以上特点,使得该模型能够根据语料库自动确定相应的最优的潜在主题数目,并能很好地对新进的 Web 服务的主题分布做出预测.

(2) 基于非功能特性的 Web 服务推荐.

以非功能特性为基础的 Web 推荐方法主要关注于 Web 服务的质量或历史调用规律,采用协同过滤、矩阵分解等技术来预测 Web 服务的服务质量(QoS)或出现概率,推荐 Top-*K* 高质量的 Web 服务给用户[6−9].基于协同过滤的 Web 服务推荐[6,7],简单来说是利用兴趣相投或拥有共同经验的群体的喜好来推荐用户感兴趣的信息.Zheng 等人[6]利用协同过滤技术进行 QoS 预测,提出了一种用户协同机制,该机制能够从不同服务的用户行为中搜集有用的服务的 QoS 信息.在此基础上,Chen 等人[7]将位置信息和 QoS 信息结合,来构建区域模型.基于矩阵分解的 Web 服务推荐进一步提高 Web 服务推荐的精度[8,9],它是将 Web 服务的历史调用矩阵拆解为两个低维

矩阵,并利用这些子矩阵推荐 Web 服务.Lo 等人[8]使用基于位置规则的矩阵分解技术预测 Web 服务的 QoS.He[9]提出了一种基于位置的多层次矩阵分解模型来推荐 Web 服务.然而,不管是协同过滤还是矩阵分解,都不可避免地会出现矩阵稀疏性问题,且其输入的数据类型不具普遍性(过于单一),从而影响推荐的精度.Rendle 等人[10,11]提出了因子分解模型,就很好地解决了这一问题,该模型将任意长度、任意数量的特征向量作为输入,这就使得其能处理多维度的信息输入.即使矩阵稀疏,也不会影响推荐精度.因此,本文使用该模型来融合多维度信息,继而进一步提高服务发现质量.

(3) 混合特性的 Web 服务推荐.

由于混合特性的 Web 服务推荐方法能够结合功能特性和非功能特性以提高服务发现的精度,使得其成为时下的研究热点[12−17].Yao 等人[12]提出了一种新颖的混合 Web 服务的功能特征和 QoS 信息 Web 服务推荐方法.Cao 等人[13]探索出一种两层主题模型,该模型聚合了服务的文本信息和服务的网络信息去增强服务聚类的效果,进而提高 Web API 的推荐精度.Gao 等人[14]提出了一种基于“多样学习”的 API 推荐方法,紧接着,他们又提出了集合用户模型、服务、Mashup 和主题的服务推荐方法[15].针对 Mashup 的自动创建,Xia 等人[16]提出了一种具有分类意识的 API 聚类和分布式推荐方法,该方法使用一种扩展的 *K*-Means 聚类方法来对 API 进行聚类,接着开发了一种分布式的机器学习框架用来预测服务排序.Liu 和 Fulia[17]通过聚合用户、主题和服务潜在特征关系来提升服务推荐的质量.同样,本文的 HDP-FM 方法也是一种混合服务推荐法,既考虑了 Web 服务的功能特性又考虑了非功能特性(包括功能特性“Mashups 之间的相似度、Web APIs 之间的相似度”和非功能特性“Web APIs 的流行性和 Web APIs 的共线性”).

2 概率主题模型

概率主题模型是一系列旨在发现隐藏在大规模文档中的主题结构的算法,其中,LDA 是最为传统的概率主题模型,它能够提取文档的隐含主题,将文档从高维的词向量空间映射到低维的主题向量空间中.LDA[2]模型基于 3 点假设.

- 假设 1(词袋模型):一篇文档是由一组词构成的一个集合,词与词之间不考虑先后顺序关系;
- 假设 2:用于训练的文档集中文档不考虑顺序先后顺序;
- 假设 3:参数化的贝叶斯模型在训练时需预先指定聚类数目(主题数 *K*).

在 LDA 模型中,一篇文档可以包含多个主题,文档中的每个词都由其中的一个主题生成.给定特定的文档集和主题数 *K*,LDA 假设文档集中所有文档共享这 *K* 个主题,但每篇文档具有不同的主题分布,因此,整个模型的训练就是估计文档集中“文档-主题”分布 *H* 和“主题-词”分布 *F*.虽然使用 LDA 可以成功地学习语料库以生成主题,但是与大多数主题模型类似,其主题数 *K* 需要事先给定.更重要的是,主题数 *K* 的选取很大程度上会影响模型的训练效果.通常情况下,为了寻找最优主题数,需要不断调节主题数 *F*,反复训练多个 LDA 主题模型,至使模型的训练变得越发繁琐.为解决这一问题,我们探索 HDP 主题模型来获得文档主题分布.该模型为非参数贝叶斯模型,能够根据训练的语料库自动获取相应语料库的最佳主题数 *K*.该过程能够有效避免 LDA 模型的假设 3 所带来的限制.本文利用 HDP 模型对 Web APIs 的描述文档训练生成主题分布向量.例如 $WS^{a_i}=(t_1^{(a_i)},t_2^{(a_i)},...,t_K^{(a_i)})$,其中,$WS^{a_i}$ 表示 Web API “a_i”的主题分布向量,$t_2^{(a_i)}$ 表示 Web API “a_i”在第 2 个主题下的分布率.为了介绍 HDP 模型的工作原理,需要先引入一个重要概念“Dirichlet 过程模型(DP)”[18].DP 模型是非参数贝叶斯统计模型中的一种随机过程模型,且被广泛应用于文本相似度计算.在此给出 DP 模型定义.

定义 1(DP 模型)[18]. 假设 H_0 是测度空间 Θ 上的随机概率分布,参数 δ 是正实数,测度空间 Θ 上的概率分布 H 满足如下条件:对于测度空间 Θ 的任何划分(有限的或无线的)$R_1,R_2,\ldots,R_i$,都有公式(1)成立:

$$\{H(R_1),H(R_2),\ldots,H(R_i)\}\sim Dirichlet\{\delta H_0(R_1),\delta H_0(R_2),\ldots,\delta H_0(R_i)\} \quad (1)$$

则称 H 服从由基分布 H_0 和参数 δ 组成的 *Dirichlet* 过程 $H\sim(\delta,H_0)$;反之,如果 H 是一个根据 DP 得到的可测量的随机分布,则称 H_0 为 H 的基分布.

HDP 模型假设主题个数是无穷的,利用两层 DP 模型来获取服务描述文档主题分布向量.现在我们详细介

绍 HDP 模型的实现过程,图 2 为 HDP 模型生成服务描述文档主题分布向量的图模型,表 1 对 HDP 图模型中的符号进行了详细说明.

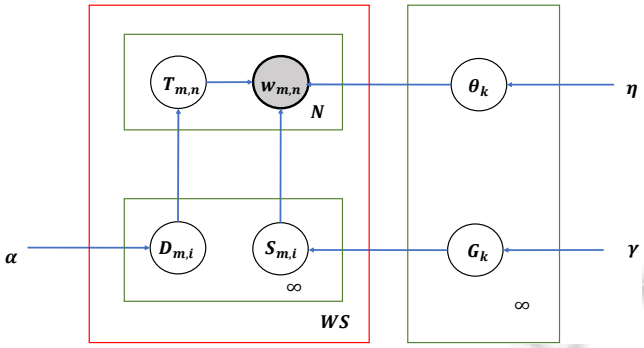


Fig.2 Graphic model of HDP

图 2 HDP 的图模型表示

Table 1 Symbolic description in HDP model

表 1 HDP 模型中的符号说明

符号	意义
α	生成文档 Stick-breaking 概率的 Beta 分布参数
γ	生成语料库的 Stick-breaking 概率 Beta 分布参数
η	生成隐含主题的 Dirichlet 分布参数
M	服务文档数
m	服务描述文档
N	序料库中的词总数
n	服务描述文档中的词(整个文档集)
θ_k	主题分布向量
G_k	语料库的 Stick-breaking 概率
$D_{m,i}$	文档的 Stick-breaking 概率
$T_{m,i}$	主题索引
$S_{m,i}$	服务层主题索引
$w_{m,n}$	服务描述文档 m 中的词
W	语料库中词的种类数

HDP 模型的实现过程具体描述如下:

(1) 初始化无限的主题分布向量θ_k,即允许 k 值为无穷大.例如θ_k~*Dirichlet*(η),该公式指对任意的 $k\in\{1,2,3,\ldots\}$均有θ_k满足以η为参数的 Dirichlet 分布,这一过程类似于 LDA 模型中的主题采样过程.

(2) 在整个文档集,抽样生成主题在语料库层的分布率 G_k,该过程的实现依赖于 *Bate* 分布以及参数γ.例如 G_k~*Beta*(1,γ),同样 $k\in\{1,2,3,\ldots\}$,实现这一过程依赖于 HDP 的 Stick-breaking 构造方法(Stick- breaking 构造方法是 HDP 模型采样的实现方法,用于指定主题概率的分布)[18],即

$$\sigma_k(G)=G_k\times\prod_{j=1}^{k-1}(1-G_k),\text{其中,}\sum_{k=1}^{\infty}\sigma_k(G)=1.$$

(3) 对每一个服务描述文档 m.

- 第 1 步,利用 *Multinomial* 分布函数抽样生成服务层主题索引,即 $S_{m,i}$~*Multinomial*($\sigma(G)$)公式中的 $i\in\{1,2,3,\ldots\}$,该步骤指根据语料库层的分布率抽样生成服务描述文档主题.需要注意的是,一个特定的服务描述文档抽样至一个特定的语料库子集,并不是整个文档集.
- 第 2 步,利用一个带有参数α的 Beta 分布函数来抽样生成服务描述文档的 Stick-breaking 概率,即 $G_{m,i}$~*Beta*(1,α)[19],同样 $i\in\{1,2,3,\ldots\}$.第 2 步是对第 1 步选取的主题进一步分配概率比例的过程.类似于 LDA 模型的主题分布抽样,不同的是,在 LDA 模型中,每一个主题至少会在一个描述文档中出现;但在 HDP 模型中,只有经过第 1 步选取的主题才会出现在描述文档中.

(4) 对每一个来至服务描述文档的词 n.

- 第 3 步,为每一个词分配主题 $T_{m,n}$~$Multinomial(\sigma(D_{m,i}))$,该过程指抽样生成服务描述文档 m 中的第 i 个词的主题.
- 第 4 步:使用第 3 步获取的主题,抽样生成词 $w_{m,n}$,即 $w_{m,n}$ ~ $Multinomial(\theta_{m,T_{m,n}})$.

潜在主题存在相互依赖与潜在主题个数允许无限,使得利用 HDP 模型推断(计算)覆盖全部潜在主题的先验概率分布变得越发困难.为解决这一问题,HDP 模型利用变分推断方法[20]来近似计算真实后验概率分布.其首先利用变分分布来分解潜在的主题,接着设置语料库级别的最大的主题数 $K_{\max}$ 与服务描述文档级别的最大主题数 l 来进一步控制条件分布范围.其变分分布的计算公式如下:

$$q(\theta,G,m,D,T)=(\prod_{k=1}^{K_{\max}} q(\theta_k \mid \lambda_k)q(G_k \mid \vartheta_k))(\prod_{m=1}^{M}\prod_{i=1}^{I} q(S_k \mid \psi_k)q(D_k \mid \gamma_k)\prod_{n=1}^{N}(T_{m,n} \mid \phi_{m,n})) \tag{2}$$

$$q(\theta,G,m,D,T)=q(\theta)q(G)q(m)q(D)q(T) \tag{3}$$

通过利用变分推断的正则化结果[20],我们可以获得最优的特征分布:

$$\ln q_j^* = \Delta_{i\neq j}[\ln p(W,Z)] + const \tag{4}$$

$$\Delta_{i\neq j}[\ln p(W,Z)] = \int \ln p(W,Z)\prod_{i\neq j} q_i dZ_i \tag{5}$$

其中,$Z=(\theta,G,m,D,T)$表示整个特征集合.变分推断是一个实时交互的过程,其利用公式(3)一次仅计算一个变分分布率,直到迭代收敛为止.综上可得:HDP 模型主要依据词频共现来聚类分组数据,且不依赖于参数 K(主题数);HDP 模型是具有两层 DP 模型的非参数贝叶斯模型,相比于基于简单概率分布的参数贝叶斯模型 LDA 来说具有较高的时间复杂度和空间复杂度.

3 方法概述

图 3 给出了 HDP-FM 方法的 Web API 推荐框架,此框架包括分为两个部分.

(1) 多维信息的构建.针对 Web API 和 Mashup 的描述文档,利用 HDP 模型以获得 Web API 和 Mashup 的主题分布,进一步利用增强余弦相识度公式[1]度量 Mashup 与 Mashup 之间的相似度、Web API 与 Web API 之间的相似度;针对 Web API 的 category 与历史调用次数,利用增强流行度公式计算 Web API 的流行度;针对 Web API 的组合关系,结合 Jaccard 相似系数公式,衡量 Web API 的共线性;

(2) 多维信息的融合.将第 1 部分获取的 Top-A 相似的 Web API、Top-M 相似的 Mashup、Web API 的流行度和 Co-occurrence(共现性)注入因子分解机模型,为目标 Mashup 的创建推荐 Top-N Web APIs.

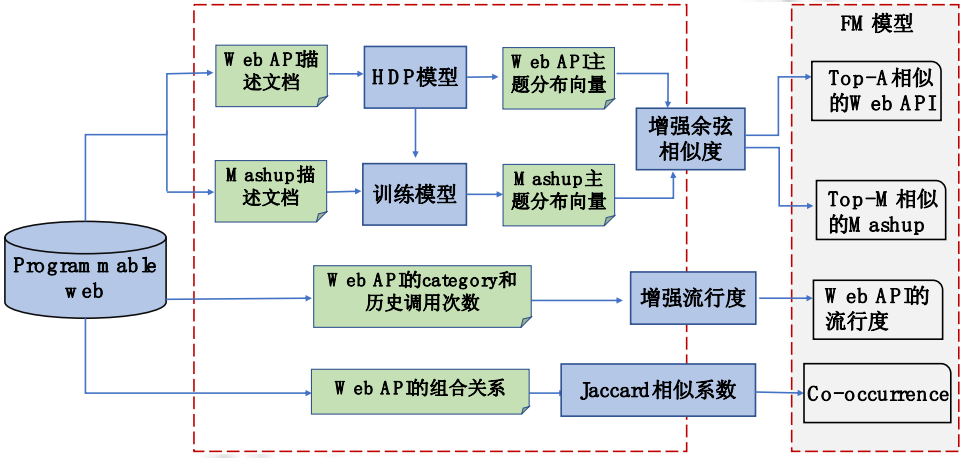


Fig.3 Framework of Web API recommendation by HDP-FM method

图 3 HDP-FM 方法的 Web API 推荐框架

3.1 相似度度量

相似度的度量是本文需要解决的核心问题,本文采用增强余弦相似度公式[1](6).HDP-FM 方法将 Web APIs 的描述文档作为 HDP 模型的输入,通过迭代抽样生成每一个 Web API 描述文档的主题分布向量;然后,利用已训练好的 HDP 模型为每个 Mashup 描述文档分配主题,以获得 Mashup 描述文档的主题向量;最后,通过公式(6)计算 Web API 与 Mashup 之间的相似度,同理可获取 Web API 与 Web API 之间的相似度以及 Mashup 与 Mashup 之间的相似度:

$$S(WS^{a_i},WS^{m_j})=\frac{\sum_{l=1}^{K}\xi\left(\frac{1}{e^{\mu\left|WS_l^{a_i}-WS_l^{m_j}\right|}}\right)WS_l^{a_i}\cdot WS_l^{m_j}}{\sqrt{(t_1^{(a_i)})^2+...+(t_K^{(a_i)})^2}\sqrt{(t_1^{(m_j)})^2+...+(t_K^{(m_j)})^2}} \tag{6}$$

其中,$WS^{a_i}=(t_1^{(a_i)},t_2^{(a_i)},...,t_K^{(a_i)})$ 与 $WS^{m_j}=(t_1^{(m_j)},t_2^{(m_j)},...,t_K^{(a_j)})$ 分别为 Web API "a_i"和 Mashup "m_j"的主题分布向量;$\xi(\cdot)$为惩罚值,用于对不相似的 Web API 进行惩罚,即主题向量之间对应位置元素差值越大,惩罚越强.当μ=0 时,该公式将退变为一般的余弦相似度公式.

3.2 Web API的流行度与共现性

如仅仅使用 HDP 模型来计算描述文本之间的相似度,其推荐精度不高.为了进一步提高推荐精度,我们引入 Web API 的流行度与共现性.Web API 的流行度能够很好地反映 Web API 的 QoS 信息,本文确定的 Web API 流行度计算公式如式(7)所示:

$$pop(a_i)=\frac{Fre(a_i)-\mathrm{Min}Fre(Category(a_i))}{\mathrm{Max}Fre(Category(a_i))-\mathrm{Min}Fre(Category(a_i))} \tag{7}$$

其中,$Fre(a_i)$为 Web API "a_i"被历史 Mashup 调用的次数,$Category(a_i)$表示与 Web API "a_i"具有相同领域属性的全部 Web API,Min$Fre(\cdot)$为历史上 Web API 被 Mashup 历史上最小的调用次数,Max$Fre(\cdot)$为历史上 Web API 被 Mashup 历史上最大的调用次数.

Web API 的共现性实际上是 Web API 组合关系的外在表现,HDP-FM 方法采用经典的 Jaccard 相似系数方法来计算 Web API 的共现性:

$$Co(a_i,a_j)=\frac{|a_i\cap a_j|}{|a_i\cup a_j|} \tag{8}$$

其中,$|a_i\cap a_j|$表示 Web API "a_i"和 Web API "a_j"被同一个 Mashup 调用的总次数,$|a_i\cup a_j|$表示 Web API "a_i"和 Web API "a_j"被历史 Mashup 调用的 Mashup 的总个数.

3.3 Top-*N* Web APIs推荐

因子分解机模型[10,11]吸取了支持向量机和矩阵分解模型的优点,其核心思想是:通过矩阵分解的方式将用户和物品的关联矩阵映射到同一个隐因子空间上,得到用户和物品在隐因子空间上的特征向量,通过计算用户特征向量和物品特征向量的内积,预测出用户对物品的评价.由于因子分解机模型能够在其模型中加入各种补充信息,从而能有效降低传统协同过滤算法中矩阵的稀疏性,进而优化特征组合的方式.此外,因子分解机模型理论基础坚实,参数优化方法完备.因此,本文利用该模型来融合多维度的信息,并预测 Mashup 与 Web API 的之间的链接关系或缺省值.

于是,一个二阶的因子分解模型被定义如下:

$$Y(X):=w_0+\sum_{i=1}^{n}w_ix_i+\sum_{i=1}^{n}\sum_{j=i+1}^{n}\langle v_i,v_j\rangle x_ix_j \tag{9}$$

利用模型参数 $w_0\in R,w\in R^n$ 和 $V\in R^{n*k}$,公式(9)可以被化简为

$$Y(X):=w_0+\sum_{i=1}^{n}w_ix_i+\frac{1}{2}\sum_{f=1}^{k}((\sum_{i=1}^{n}v_{i,f}x_i)^2-\sum_{i=1}^{n}v_{i,f}^2x_i^2) \tag{10}$$

其中,n 表示特征的长度(个数),w_0 表示初始偏移量,w_i 表示第 i 个特征的权重,x_ix_j 是指成对的特征变量之间的相

互作用,$\langle v_i,v_j\rangle$表示在因子分解模型中 Mashup x_i 与 Web API x_j 之间的相互影响,k 为因子分解矩阵维度.

利用上述的二阶因子分解模型,可以预测每一个 Web API 相对于目标 Mashup 的评分值,对评分值进行排序,便可获得最终的推荐列表.图 4 是利用因子分解模型为目标 Mashup 预测 Web API 评分的实例,训练数据集中的数据被分成两个部分:第 1 部分为输入特征向量集(feature vector X),第 2 部分是输出目标集(target Y).每一行为一个特征向量 x_i 以及相应的目标评分值 y_i.图 4 中,每一行从左到右依次为:

- 第 1 个矩阵(Box1)代表活动的 Web API(例如,x_i 行的 Box1 中的第 1 个值为 1,表示 A_1 为活动的 Web API);
- 第 2 个矩阵(Box2)代表活动的 Mashup(例如,x_i 行的 Box2 中的第 2 个值为 1,表示活动的 Web API A_1 历史上被 M_2 调用过);
- 第 3 个矩阵(Box3)表示与活动的 Web API 的最相似的 Top-A 个 Web API(例如,x_i 行的 Box3 中的第 2 个值为 0.3,表示活动的 Web API A_1 与 Web API A_1 的相似度为 0.3,且在活动的 Web API A_1 最相似的 Top-A 个 Web API 之中);
- 第 4 个矩阵(Box4)表示与活动的 Mashup 的最相似的 Top-M 个 Mashup(例如,x_i 行的 Box4 中的第 3 个值为 0.7,表示活动的 Mashup M_2 与 Mashup M_3 的相似度为 0.7,且在活动的 Mashup M_2 最相似的 Top-M 个 Mashup 之中);
- 第 5 个矩阵(Box5)表示与活动的 Web API 共现的 Web API 的共现值(例如,x_i 行的 Box5 中的第 3 个值为 0.5,表示活动的 Web API A_1 与 Web API A_3 的共现值);
- 第 6 个矩阵(Box6)表示活动的 Web API 流行度值.
- 最后,在训练集中向量 Y 中的值为 1,表示活动的 Web API 历史上被活动的 Mashup 调用过;0 表示未被调用过.在测试集中,向量 Y 中的值表示活动的 Web API 相对于活动的 Mashup 的预测评分.最终的推荐 Web API 集合是通过对预测评分的排序获得的.

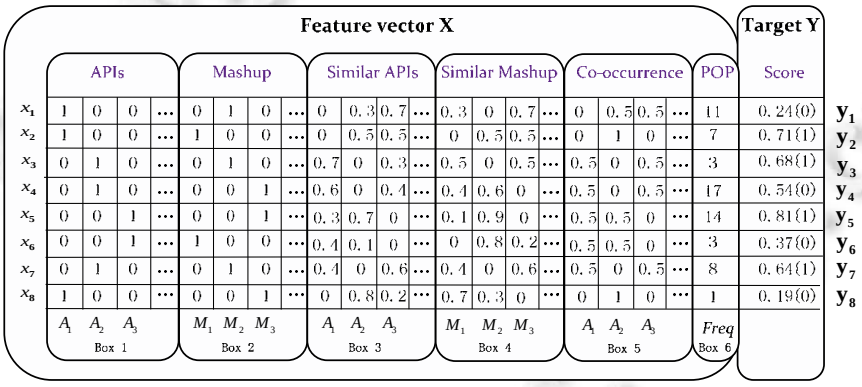


Fig.4 A case for predicting the score of Web API

图 4 预测 Web API 的评分实例

在应用因子分解模型对预测 Web API 评分之前,需要对观测数据集中的每一对(x_i,y_i)求出真实值与预测值之间误差之和,使其最小化,以求得最佳的参数$\Theta=(w_i,w,V)$集合,其中 $V\in R^{n*k}$ 为低秩矩阵,用于表示变量间的相互作用).为了寻找最优的模型参数,需要设计合适的损失函数 l:

$$Opt(S,\lambda)=\arg\min_{\Theta}\sum_{i=1}^{N}l(Y(x_i),Y')+\sum_{\theta\in\Theta}\lambda_{(\theta)}\theta^2 \tag{11}$$

其中,N 是训练集中实例的数量,$\lambda_{(\theta)}\in R_+$为正则化项的系数,$\sum_{\theta\in\Theta}\theta^2$ 表示参数集合的二范数.由于本文为 Binary 分类问题,因此,本文选取 Binary 分类问题常使用的 Logic 优化函数来优化损失函数.

$$l(Y,Y')=\log(1+\exp(-YY')) \tag{12}$$

对损失函数优化后,便需要选择合适的参数求解方法,常用的参数求解方法有:随机梯度下降(SGD)、马尔科

夫裴特卡罗法(MCMC)、交替最小二乘法(ALS).其中,SGD 最为常用且具有较快的训练速度,本文的参数求解也使用该方法.

4 实验设计

4.1 数据集选取与实验设置

数据集采用从 ProgrammableWeb 平台上爬取的真实数据,包含 6 673 个 Mashup,9 121 个 Web APIs 以及 13 613 个 Web API 和 Mashup 之间的链接.由于 HDP 模型主要依据词频共现来聚类分组数据,为了导出最相关的主题,我们对 Web APIs 和 Mashup 的描述文档做预处理.图 5 给出了名称为“Earthmine Flash Viewer”的 Web API 描述文档预处理的详细过程.

Web API 名称	Earthmine Flash Viewer
第 1 步:选取 Mashup 和 Web APIs 的描述文档作为原始文件,每一行代表一个 Mashup 或 Web API.	Integrate street-level images into your site via this Flash Viewer. Each image has longitude and latitude data, and you can do geo searches, keyword searches, and more. Contact for more information. Like the earthmine Flash Viewer API? Click the “Track this API” button on any profile page and never miss an API update, new app, or breaking news for that API again.
第 2 步:去除不合理的特征(包括,.;:=+?@%$|等).	Integrate street level images into your site via this Flash Viewer Each image has longitude and latitude data and you can do geo searches keyword searches and more Contact for more information Like the earthmine Flash Viewer API Click the Track this API button on any profile page and never miss an API update new app or breaking news for that API again
第 3 步:将所有字母均转换为小写字母.	Integrate street level images into your site via this flash viewer each image has longitude and latitude data and you can do geo searches keyword searches and more contact for more information like the earthmine flash viewer api click the track this api button on any profile page and never miss an API update new app or breaking news for that api again
第 4 步:去除停止词,并使用英语词典校验单词.	Integr street level imag site flash viewer imag ha longitud latitud data geo search keyword search contact inform earthmin flash viewer api click track api button ani profil api updat app break api
第 5 步:去除超高频且不具区分度的词(service,API)、词频为 1 的词.此类词既不具有区分度,且在训练语料库是严重消耗系统资源.	Integr street level imag site flash viewer imag longitud latitud geo search keyword search contact inform earthmin flash viewer click track button ani profil updat app break
第 6 步:使用 POSTagger 工具标注单词词性.	Integr/JJ street/NN level/NN imag/NN site/NN flash/VBP viewer/NN imag/NN longitud/NN latitud/JJ geo/NN search/NN keyword/NN search/NN contact/NN inform/VBP earthmin/JJ flash/NN viewer/NN click/VBP track/NN button/NN ani/NN profil/NN updat/NN app/NN break/NN
第 7 步:选取名词词性及动词词性的单词作为最终训练的语料库.	Street level imag site flash viewer imag longitud geo search keyword search contact inform flash viewer click track button ani profil updat app break
HDP 算法的输入文件格式(每一行为一个词频向量):描述文档词数 第 1 个词索引:第 1 个词的词频 第 2 个词索引:第 2 个词的词频...	24 533:149 266:319 57:1307 1056:59 444:186 930:69 266:319 459:180 519:152 10:3788 180:473 10:3788 109:770 9:4266 444:186 930:69 76:1005 35:1916 120:711 39:1837 61:1240 52:1514 4268:6 134:667

Fig.5 A case analysis of experimental pretreatment

图 5 实验预处理案例分析

- 实验环境设置:利用 Java 语言实现 HDP 算法,运行在一台 60G 内存的服务器;
- 实验中参数设置:HDP 算法迭代次数 Iter 为 3 000,HDP 模型参数 α=1, η=0.1, γ=1.5.实验中,Top-A 个 APIs 和 Top-M 个 Mashup 在因子分解机模型中分别设置为 10 和 20.实验中,Mashup 数据集被随机平均分成 5 个部分,其中一个部分为测试集,而另外 4 个部分边为训练集.

4.2 实验的评测指标

本文采用召回率、准确率、*F*-measure 和 *NDCG@N* 指标来评价以上 6 种方法性能的优劣,分别定义如下.

- 召回率反映的是推荐的相关 Web APIs 占所有相关 Web APIs 的比率,计算如公式(13)所示.

$$\text{召回率}=\frac{|R(A_i)\cap RM(A_i)|}{RM(A_i)} \tag{13}$$

- 准确率反映推荐的 Web APIs 集合中相关 Web APIs 所占的比例,计算如公式(14)所示.

$$准确率=\frac{|R(A_i)\cap RM(A_i)|}{R(A_i)} \tag{14}$$

- *F*-measure 是召回率与准确率的调和平均值,即

$$F\text{-measure}=\frac{2\times 召回率\times 准确率}{召回率+准确率} \tag{15}$$

其中,$R(A_i)$表示相关的 Web APIs(被目标 Mashup 真实调用的 Web APIs),$RM(A_i)$表示推荐的 Web APIs.

- *NDCG@N*(normalized discounted cumulative gain:归一化折损累积增益).在信息检索领域中,该方法是一种流行的的衡量排序质量的指标.本文用来衡量推荐列表中推荐 Web APIs 排名的优劣.*NDCG@N* 的值越高,说明 Web APIs 的推荐列表排序结果越好.即

$$DCG@N=\sum_1^N\frac{2^{rel_i}-1}{\log_2(1+i)} \tag{16}$$

$$NDCG@N=\frac{DCG@N}{IDCG} \tag{17}$$

其中,

- N 表示 Web APIs 的推荐个数;
- rel_i 表示第 i 个推荐的 Web API 的相关性得分:如果推荐的第 i 个 Web API 就是真实数据集中 Mashup 调用的 Web APIs,此时 rel_i=1;否则,rel_i=0;
- *IDCG*(ideal *DCG@N*),就是最大的 *DCG@N* 值(*DCG@N* 可以通过公式(16)计算得到),即为最优的推荐情况.

4.3 比较方法

- TF-IDF[1]:

利用词向量空间模型推荐与目标 Mashup 描述文档相似的 Web APIs.假设目标 Mashup m 的词向量空间为 $V^{(m)}$,第 i 个 Web API 的词向量空间为 $V^{(a_i)}$,则目标 Mashup m 与第 i 个 Web API 的文本相似度计算如公式(18)所示.

$$Sim(m,a_i)=\frac{V^{(m)}V^{(a_i)}}{\|V^{(m)}\|\|V^{(a_i)}\|} \tag{18}$$

最后,该 Web API 预测评分通过公式(19)计算得出.

$$Score(m,a_i)=pop(a_i)Sim(m,a_i) \tag{19}$$

其中,$pop(a_i)$为第 i 个 Web API 的流行度,可通过第 3.2 节中的公式(7)计算.

- E-LDA[3]

该方法使用 LDA 模型分别导出的 Mashup 和 Web API 的主题分布向量;接着,利用增强余弦相似度公式(6)来度量 Mashup 与 Web APIs 之间的文本相似度 $S(m,a_i)$;最后,将相似度高且流行的 Top-N 个 Web APIs 推荐给目标 Mashup.Web API 预测评分如公式(20)所示.

$$Score(m,a_i)=pop(a_i)S(m,a_i) \tag{20}$$

- E-HDP

类似于 E-LDA 推荐方法,该方法推荐相似度高且流行的 Top-N 个 Web APIs 给目标 Mashup.不同的是,该方法获取服务描述文档主题分布向量是由 HDP 模型导出.

- LDA-CF[6]

利用 LDA 模型与协同过滤方法共同获取候选的 Web API 集合,推荐候选集合中最流行的 Top-N 个 Web APIs 供 Mashup 创建使用.公式(21)给出了相应的推荐 Web APIs 的获取.

$$RecommendAPI=pop\text{Max}_N\{Cond_{i=i}^I(m,a_i)\} \tag{21}$$

其中, $Cond_{i=i}^{I}(m,a_i)$ 为 Mashup m 的候选 Web API 集合,$popp\text{Max}_N\{\cdot\}$表示取流行度最高的前 N 个 Web APIs.

- HDP-CF

类似于 LDA-MF 推荐方法的实现过程,在此基础上,使用 HDP 模型代替 LDA 模型训练服务描述文档的主题分布向量.

- LDA-FM

利用因子分解模型的优势,同时考虑文本相似度、Web API 的流行度与 Web API 的共线性,预测评分推荐 Top-N 个得分最高的 Web APIs 辅助 Mashup 创建.

- HDP-FM

利用因子分解模型融合文本相似度、Web API 的流行度、Web API 的共线性,预测评分推荐 Top-N 个得分最高的 Web APIs 辅助 Mashup 创建.与 LDA-FM 方法唯一的不同是,服务描述文本的主题分布向量由 HDP 模型训练获得.

4.4 实验评估

本小节我们首先对 HDP 模型的训练过程进行观测,接着探讨了主题 K 的选取对实验结果的影响,最后选取准确率、召回率、F-measure 和 $NDCG@N$ 这 4 种评价指标对多种方法进行比较.

(1) HDP 模型训练观测

HDP 模型在训练结束后会获得训练语料库的最优主题个数、每个主题下的词分布以及每一个 Mashup 和 Web API 的主题分布概率.表 2 展示了利用 HDP 模型获取的部分“主题-词”分布的情况,如,sport game score player 等词被分配到 Topic 14 中.表 3 展示了部分 Web API 描述文档在表 3 所给出的主题下的分布情况.如,Web API “FishingBuddy”在 Topic 14 上的分布率为 0.324059.表 4 给出了 HDP 模型的α,η,γ参数与主题数 K 随迭代次数变化的部分记录情况,不难发现,当迭代次数约达到 2 172 次时,主题个数不再发生变化,迭代过程趋于稳定,此时获取最优主题数 K=76.

Table 2 Some topics and their representative words learned from HDP

表 2 利用 HDP 模型获取的部分“主题-词”分布

主题	该主题下的词
Topic 14	sport game score player statistic football team new
Topic 26	call voice audio phone chat telephony record conference meet response
Topic 39	communication text language tool analysis extraction content word sentiment dictionary search
Topic 48	travel estate hotel restaurant deals offer website airport trip flight
Topic 70	location map weather latitude longitude area measurement

Table 3 Some topic distribution of Web APIs document

表 3 部分 Web API 描述文档的“文档-主题”分布

Web API 名称	Topic 14	Topic 26	Topic 39	Topic 48	Topic 70
FishingBuddy	0.324 059	0.018 313	0.013 059	0	0.012 987
360voice	0.318 004	0.016 112	0.009 914	0.007 115	0.012 545
iLime	0.015 484	0.321 454	0	0.155 458	0.001 245
eKlima	0	0.081 231	0.004 611	0.035 529	0.422 245
SharedBookshelve	0.012 454	0.012 154	0.015 54	0.017 454	0.013 254

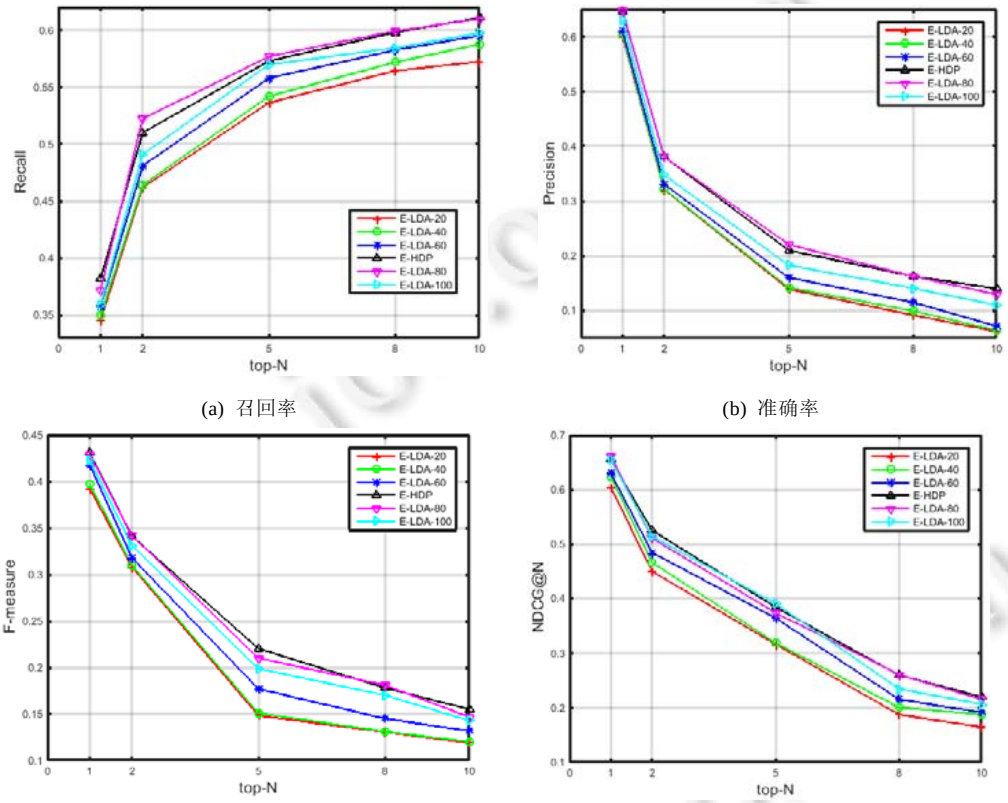
Table 4 Sample iteration record of parameters and the number of topic

表 4 参数与主题数迭代记录(部分)

迭代次数	K	α	η	γ
1	13	1.000 00	0.100 00	1.500 00
100	14	0.385 67	0.002 62	2.440 67
1 000	42	0.555 12	0.000 92	10.874 17
2 000	73	0.812 31	0.000 56	25.355 29
2 072	76	0.823 83	0.000 54	20.879 48
2 400	76	0.864 07	0.000 52	24.919 24

(2) 主题数 *K* 的影响

本组实验分别将 E-LDA 方法中的主题数 *K* 设置为 20,40,60,80,100 与 E-HDP 方法进行比较,E-HDP 方法的主题数 *K*=76 由 HDP 模型自动生成.图 6 中的纵坐标表示相应的评价指标(准确率、召回率、*F*-measure 及 *NDCG@N*)的值,横坐标为推荐 Web API 的数目,分别设置为 1,2,5,8,10.从图 6 中可以观察到:当主题数 *K* 在 20 至 80 之间时,E-LDA 的性能(准确率、召回率、*F*-measure 及 *NDCG@N* 的值)随着主题数 *K* 的增加而上升;当主题数 *K* 在 80~100 之间时,E-LDA 的性能呈现下降趋势.这说明当主题数 *K* 为 80 时,E-LDA 方法具有最优的性能.从图 6 中还可以发现,E-HDP 方法与主题数 *K* 为 80 时的 E-LDA 方法性能基本相同(准确率、召回率、*F*-measure 及 *NDCG@N* 的曲线相互缠绕近似重叠),证明 HDP 模型能够自动生成最优主题数.



(a) 召回率 (b) 准确率

(c) *F*-measure (d) *NDCG@N*

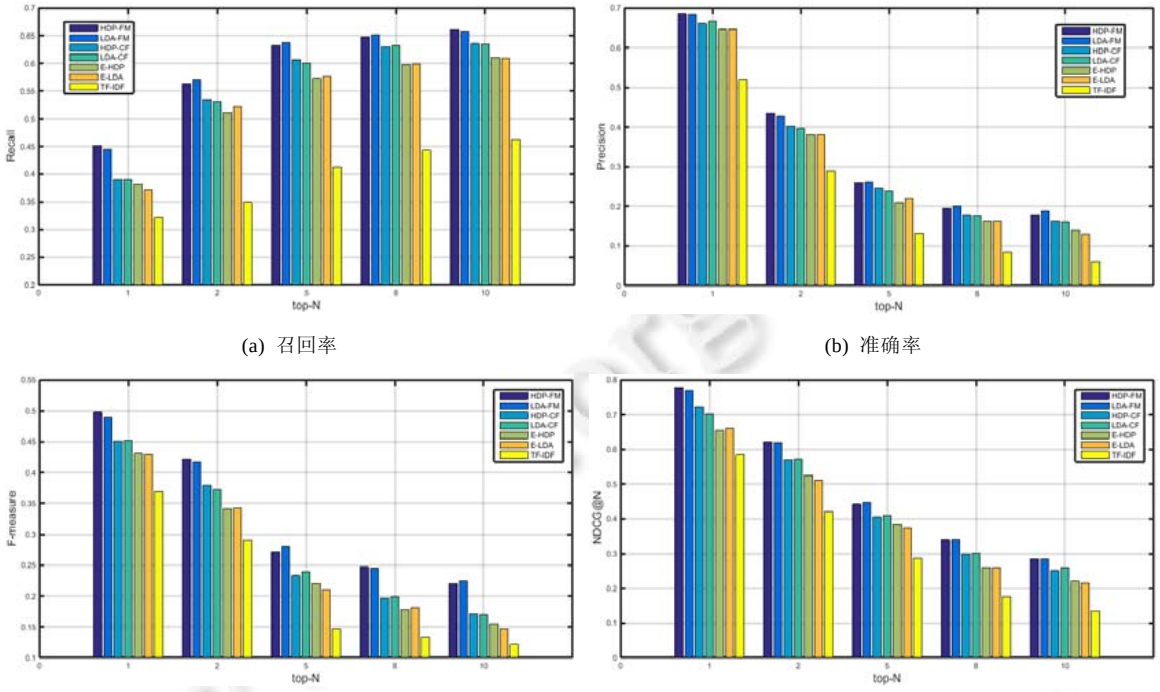
Fig.6 Impact of the number of topic on Web API recommendation

图 6 主题个数选取对 Web API 推荐的影响

(3) 推荐方法比较分析

本组实验将 HDP-FM 与第 4.3 节中介绍的方法进行比较,以此来说明 HDP-FM 具有良好的性能.与第 4.4 节情形(2)中的设置方式类似,图 7 中的纵坐标表示相应的评价指标的值,横坐标为推荐 Web API 的数目.在图 7 中我们不难发现,TF-IDF 方法的性能最差,因为 TF-IDF 方法仅仅利用词向量空间模型度量 Web APIs 与目标 Mashup 之间的相似度,该方法忽略了文档与文档之间的语义信息;其次,E-LDA 及 E-HDP 方法分别引入 LDA 模型和 HDP 模型,这两类主题模型均能挖掘描述文档的潜在语义信息(主题),因此其性能较之 TF-IDF 在相似度的计算上更为精确;再其次,LDA-CF 与 HDP-CF 方法利用协同过滤算法来增强 LDA 及 HDP 算法,以挖掘 Web API 与 Mashup 之间更深层次的链接关系,性能相比仅仅使用 LDA 及 HDP 模型又有所提升.但是 LDA-CF 与 HDP-CF 方法由于协同过滤算法的局限性,无法实现多维度信息的建模,FM 方法能够将多维度的信息作为输入且避免

稀疏性带来的影响,因此,LDA-FM 及 HDP-FM 方法在准确率、召回率、*F*-measure、*NDCG@N* 上表现最优.值得注意的是,实验中,凡是使用到 LDA 模型的方法均通过手动调节选取最优主题数,而涉及 HDP 模型的方法的最优主题数均由 HDP 模型自动生成.从图 7 中我们可以看出,HDP-FM 与 LDA-FM、HDP-CF 与 LDA-CF、E-HDP 与 E-LDA 均具有相似的性能,这进一步说明了 HDP-FM 方法的优越性.



(a) 召回率 (b) 准确率

(c) *F*-measure (d) *NDCG@N*

Fig.7 Recommendation performance comparison of various means

图 7 多种方法推荐的性能比较

5 总结与展望

本文着眼于“根据用户的自然语言描述需求推荐可行的或者推荐可用于解决问题的任务集合”,提出了 HDP-FM 方法用于推荐解决 Mashup 构建问题的 Web APIs 服务集合.HDP-FM 方法通过多维信息处理与多维信息融合两个阶段,将 Mashups 之间的相似度、Web APIs 之间的相似度、Web API 的共现性及流行度输入因子分解机模型,利用评分排序结果,为 Mashup 的创建推荐 Top-*N* Web APIs 作为推荐集合.本文的核心在于“利用 HDP 模型解决最优主题选取问题,利用因子分解机模型解决多维信息融合问题”.为了验证 HDP-FM 方法的有效性,本文分别采用了 4 种评价指标,比较了多种 Web APIs 推荐方法.一系列的实验结果均表明,HDP-FM 算法能够自动确定最优主题,且具有较高的推荐准确性.在未来的工作中,我们将继续探索机器学习技术来进一步提高 Web API 推荐的精度,如,可利用 LSTM(long short-term memory,长短期记忆网络)神经网络算法结合协同过滤、矩阵分解、因子分解机模型进一步提高服务发现的精度.

References:

[1] Cao B, Liu J, Tang M, Zheng Z, Wang G. Mashup service recommendation based on usage history and service network. Int’l Journal of Web Services Research, 2013,10(4):82−101.

[2] Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. Journal of Machine Learning Research, 2003,3:993−1022.

[3] Li C, Zhang R, Huai J, Sun H. A novel approach for API recommendation in Mashup development. In: Proc. of the IEEE 21st Int'l Conf. on Web Services. Anchorage, 2014. 289−296.

[4] Chen L, Wang Y, Yu Q, Zheng Z, Wu J. WT-LDA: User tagging augmented LDA for web service clustering. In: Proc. of the Int'l Conf. on Service-Oriented Computing. New York: Springer-Verlag, 2013. 162−176.

[5] Teh YW, Jordan MI, Beal MJ, Blei DM. Sharing clusters among related groups: Hierarchical Dirichlet processes. Advances in Neural Information Processing Systems, 2005,37(2):1385−1392.

[6] Zheng Z, Ma H, Lyu M, King I. Qos-Aware Web service recommendation by collaborative filtering. IEEE Trans. on Services Computing, 2011,4(2):140−152.

[7] Chen X, Zheng Z, Yu Q, Lyu MR. Web service recommendation via exploiting location and QoS information. IEEE Trans. on Parallel Distributed System, 2014,25(7):1913−1924.

[8] Wei L, Yin J, Deng S, Li Y, Wu Z. Collaborative Web service QoS prediction with location-based regularization. In: Proc. of the IEEE 19th Int'l Conf. on Web Services. Honolulu, 2012. 464−471.

[9] He P, Zhu J, Zheng Z, Xu J, Lyu MR. Location-Based hierarchical matrix factorization for web service recommendation. In: Proc. of the IEEE 21st Int'l Conf. on Web Services. Anchorage, 2014. 297−304.

[10] Rendle S. Factorization machines. In: Proc. of the IEEE 10th Int'l Conf. on Data Mining. Sydney, 2010. 995−1000.

[11] Rendle S. Factorization machines with libfm. ACM Trans. on Intelligent Systems and Technology, 2012,3(3):57−78.

[12] Yao L, Sheng QZ, Ngu AHH, Yu J, Segev A. Unified collaborative and content-based web service recommendation. IEEE Trans. on Services Computing, 2015,8(3):453−466.

[13] Cao B, Liu X, Rahman MM, Li B, Liu J, Tang M. Integrated content and network-based service clustering and Web APIs Recommendation for Mashup development. IEEE Trans. on Services Computing, 2017,3(22):1−14.

[14] Gao W, Chen L, Wu J, Gao H. Manifold-Learning based API recommendation for Mashup creation. In: Proc. of the IEEE 22nd Int'l Conf. on Web Services. New York, 2015. 432−439.

[15] Gao W, Chen L, Wu J, Bouguettaya A. Joint modeling users, services, Mashup and topics for service recommendation. In: Proc. of the IEEE 23rd Int'l Conf. on Web Services. San Francisco, 2016. 260−267.

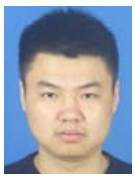
[16] Xia B, Fan Y, Tan W, Huang K, Zhang J, Wu C. Category-Aware API clustering and distributed recommendation for automatic Mashup creation. IEEE Trans. on Services Computing, 2015,8(5):674−687.

[17] Liu X, Fulia I. Incorporating user, topic, and service related latent factors into Web service recommendation. In: Proc. of the IEEE 22nd Int'l Conf. on Web Services. New York, 2015. 185−192.

[18] Pitman J. Combinatorial stochastic processes. Lecture Notes in Mathematics, 2006,1875(94):75−92.

[19] Ishwaran H, James LF. Gibbs sampling methods for stick-breaking priors. General Information, 2003,96(453):161−173.

[20] Abramson N, Braverman D, Sebestyen G. Pattern recognition and machine learning. IEEE Trans. on Information Theory, 2003,9(4): 257−261.



李鸿超(1993－),男,安徽六安人,硕士,CCF 学生会员,主要研究领域为服务计算,服务推荐.

曹步清(1979－),男,博士,副教授,CCF 专业会员,主要研究领域为软件工程,服务计算与云计算.

刘建勋(1970－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为服务计算与云计算,工作流管理的理论与应用.

石敏(1991－),男,软件工程师,CCF 学生会员,主要研究领域为服务计算,数据挖掘,人工智能.