

基于双向评价的服务信誉研究

Research on Service Reputation Based on Double-Side Evaluation

工程领域： 软件工程
作者姓名： 苗 旭
指导教师： 冯志勇 教授
企业导师： 陈列伟 正高工

天津大学智能与计算学部
二零一九年十一月

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得天津大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：荀旭

签字日期：2019年11月26日

学位论文版权使用授权书

本学位论文作者完全了解 天津大学 有关保留、使用学位论文的规定。特授权 天津大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：荀旭

导师签名：冯勇

签字日期：2019年11月27日

签字日期：2019年11月26日

摘要

服务信誉是用户对服务的综合信任程度，客正公正的服务信誉有助于建立用户与服务间的信任关系并推动服务生态系统发展。研究人员通过用户评论构建信誉系统，获取服务信誉，减少用户与服务之间的信息不对称，提高用户对服务生态系统的信任程度，辅助用户决策。然而，由于用户评论的自发性，通过用户评论构建信誉系统，获取服务信誉时存在三个无法忽视的问题，即评论数据稀疏、恶意用户内容、服务信誉计算方法选取。

为了计算服务信誉，本文提出了基于用户-服务的双向评价的服务信誉获取方法。第一，本文提出了基于用户评论质量的动态奖励机制，按照用户评论的数量与质量，调整用户评论的奖励，鼓励用户评论，提高评论数据量；第二，在一切皆服务（XaaS，X as a Service）的背景下，本文考虑到服务对用户的认知能力，提出构建服务对用户的评价体系，以提高用户评论的准确性。服务对用户的评价由客观评价与主观评价构成，客观评价由用户行为与服务运行状态决定，主观评价由用户评价决定；第三，基于 Chernoff bound，计算获取服务信誉时需要的最少用户数量，同时，借鉴众包中的质量评估方式，将用户信誉作为权重，利用 EM 算法计算服务信誉。

仿真实验结果表明，本文提出的方法，通过用户激励机制提高用户评论的数量，通过双向评价机制，降低了恶意用户的比例，提高了服务信誉的可信性和准确性。本文提出的方法，增强了用户对服务生态系统的信任，推动了服务生态系统的发展。

关键词： 服务信誉，用户评价，动态激励，恶意用户

ABSTRACT

Service reputation is the comprehensive trust among users and services. Fair service reputation can help users establish trust relationship with services. Further, it can promote the development of service ecosystem. Researchers build reputation system through user ratings to obtain service reputation, reduce information mismatch among users and services, improve users' trust of the service ecosystem, and assist users in decision-making. However, due to the spontaneity of user ratings, there are three problems that can't be ignored in when obtaining service reputation through reputation system, namely, sparse rating data, malicious user content and difficulty in choosing method to get calculating service reputation.

This paper proposes a method of obtaining service reputation based on user-service double-side evaluation. Firstly, this paper proposes a dynamic incentive mechanism based on user rating quality. It will adjust the incentive of users' rating according to the quantity and quality of users' rating. Secondly, in the context of X as a service (XaaS), this paper builds a service evaluation mechanism considering the cognitive ability of service to improve the accuracy of user feedback. The evaluation of service is composed of objective and subjective evaluation. objective evaluation is decided by user behavior and service running state while subjective evaluation is decided by user evaluation; Thirdly, based on Chernoff bound, we calculate the minimum number of users before obtaining service reputation. Meanwhile, we use the quality evaluation method in crowdsourcing for reference and calculate the service reputation using EM algorithm while user reputation is taken as weight.

The simulation results show that the proposed method can increase the number of user comments through user incentive mechanism, reduce the proportion of malicious users with double-side evaluation, and thus improving the credibility and accuracy of service reputation. The proposed method can enhance the trust of users to the service ecosystem and promote development of the service ecosystem.

KEY WORDS: Service Reputation, User Evaluation, Dynamic Incentive, Malicious Users

目 录

第 1 章 绪论	1
1.1 研究背景与意义	1
1.2 研究目标和内容	3
1.3 论文组织结构	4
第 2 章 相关工作综述	7
2.1 XaaS 概述	7
2.2 信誉系统概述	8
2.3 数据稀疏解决方法概述	8
2.3.1 众包中的数据稀疏问题	9
2.3.2 推荐系统中的数据稀疏问题	10
2.4 恶意用户生成内容过滤	11
2.4.1 不实评价过滤	11
2.4.2 垃圾评论过滤	12
第 3 章 动态奖励与用户-服务双向评价机制	15
3.1 基本概念与变量解释	15
3.2 基于用户评论质量的动态奖励机制	18
3.2.1 现有用户评级激励机制分析	18
3.2.2 激励机制的博弈分析	19
3.2.3 动态奖励机制	19
3.3 用户-服务双向评价机制	21
3.3.4 以服务为中心的数据获取方法	21
3.3.5 数据的修正与调整	23
3.3.6 服务评价	24
3.4 本章小结	25
第 4 章 基于双向评价的服务信誉计算	27
4.1 基于双向评价的用户识别与信誉值更新	27
4.1.1 用户行为分析	30
4.1.2 水军与不负责任用户识别	30
4.1.3 信誉值与权值更新	31
4.2 基于 Chernoff bound 的人数阈值计算	32

4.2.1 用户人数阈值应用场景分析	32
4.2.2 用户人数阈值计算算法	33
4.2.3 人数阈值计算合理性证明	34
4.3 基于带权 EM 算法的服务信誉评估	35
4.3.1 众包与用户评价比较	35
4.3.2 服务信誉评估	36
4.4 本章小结	36
第 5 章 实验和分析	39
5.1 实验整体流程	39
5.2 实验数据准备	40
5.2.1 用户特征模拟	40
5.2.2 服务特征模拟	42
5.2.3 实验参数设置	42
5.3 基于用户评论质量的动态奖励机制效果分析	43
5.3.1 最佳参数的获取和分析	43
5.3.2 用户激励机制效果评估总体流程	44
5.3.3 用户激励机制效果分析	45
5.4 基于用户信誉的服务信誉计算效果分析	46
5.4.1 可信用户识别准确率的对比和分析	46
5.4.2 用户人数计算	47
5.4.3 服务更新情况下的自适应对比和分析	47
第 6 章 总结和展望	49
6.1 工作总结	49
6.2 前景展望	49
参考文献	51
发表论文和参加科研情况说明	55
致 谢	57

第1章 绪论

本章首先介绍论文的研究背景与意义，接下来分析利用信誉系统获取服务信誉存在的问题以及挑战，并围绕这些问题提出本文的研究内容和目标，最后列出了本文的组织结构。

1.1 研究背景与意义

服务信誉不仅可以减少用户与服务之间的信息不对称，同时可以辅助用户决策，提高用户忠诚度。因此，很多公司或者组织都会构建服务信誉系统，比如 eBay 和亚马逊^[1]。用户评分是用户对服务功能和性能满意程度的综合度量，因此，信誉系统常常收集用户对服务的历史评分，然后这些评分经过信誉系统处理，汇总计算服务信誉。

一切皆服务（XaaS，X as a Service）的推广与发展，不仅推动了一系列服务的产生，同时，扩展了服务的概念，服务不仅仅包含传统的 Web 服务，XaaS 实际上是基础设施即服务（IaaS，Infrastructure as a Service）、平台即服务（PaaS，Platform as a Service）、软件即服务（SaaS，Software as a Service）等概念的推广，XaaS 使得服务的功能逐渐多元化。大量服务可供用户选择。如图 1-1 所示，服务可获取的数据主要包含两部分：以人为中心的感知数据和以服务为中心的感知数据。以人为中心的感知数据主要是用户可以直接获取的数据，以用户评论，评级等数据为代表。以服务为中心的数据按照数据来源的层次分为三部分：IaaS 中基础设施运行数据；PaaS 中容器、操作系统的运行日志；SaaS 中服务运行日志。

然而，用户和服务提供商的信息不对称性，使得用户在选择服务时，经常会面临一定风险，即服务的功能或者性能不能满足用户的需要，这不但会造成用户时间的浪费，更严重的，会导致用户正常业务运行失败。用户通过信誉系统，获取服务信誉，可以在一定程度上降低信息不对称的影响，并能减少用户使用具有欺骗性质的服务。利用服务信誉决定是否使用某个服务，已经成为一种普遍的方法。然而，服务信誉系统存在以下几个方面的问题。

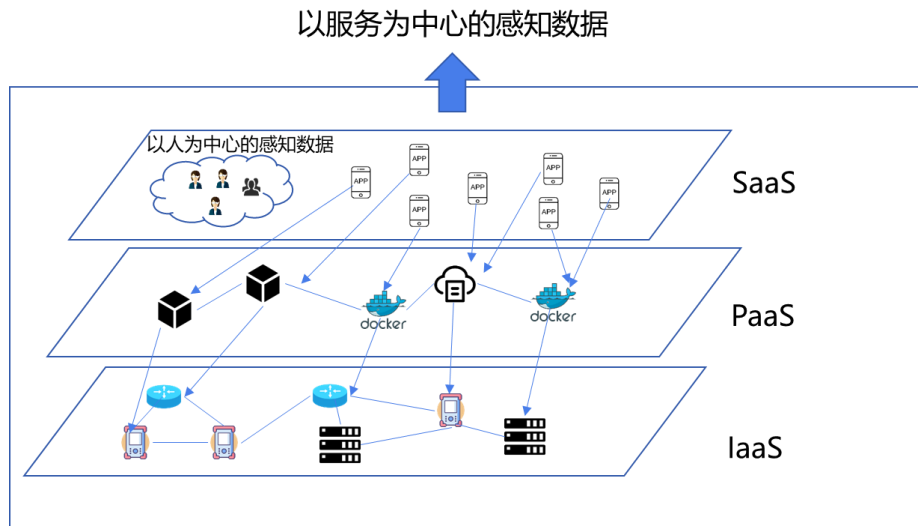


图 1-1 以服务为中心的感知数据

用户评级数据稀疏。数据显示¹，90%以上的用户会参考评级数据决定是否使用服务，然而，如图 1-2 所示，通过爬取的豌豆荚和 google play 的数据，分析来看，对服务进行评级的用户比例远小于用户下载数量的 1%。这不仅导致用于计算服务信誉的数据稀疏，还导致服务信誉的计算结果缺乏说服力。

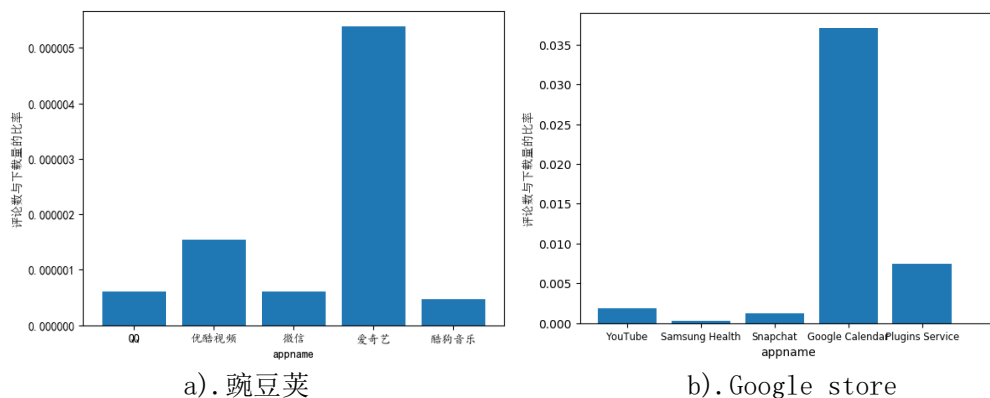


图 1-2 应用商店下载量 top5 的 APP 下载数与评论数量比例分析

用户评论质量参差不齐。由于用户偏好以及环境等其他因素的影响，很多用户对服务质量的判断会有偏差，有些用户甚至会随意对服务进行评价，让其他用户得到错误的信息，这不仅使得服务信誉缺乏可信度，还降低了用户对整个系统的信心。更糟糕的是，一些服务提供商为了自身利益，会雇佣用户提升自身信誉或者降低对手信誉。

¹ <https://www.podium.com/resources/podium-state-of-online-reviews/>

缺乏指标指导用户选择可信任的用户评论。直觉上讲，用户评论越多，计算出的服务信誉，越能反映一个服务的真实质量，然而，当用户的评级达到一个什么样的阈值时，得到的服务信誉是可信的，用户希望有一个指标能够反映当前评论的质量，以帮助用户选择服务。

总之，服务信誉在辅助用户决策，建立用户对服务的信任发挥着重要的作用。实现高质量高效率的信誉系统，获取服务信誉，可以帮助用户选择合适的服务，规避风险，提升用户体验，这对于服务生态系统的健康发展和长远运行都具有非常重要的现实意义。

1.2 研究目标和内容

本文为了调动用户的评论积极性，增加用户评价的数量，提高用户评价的质量，构建了以博弈论为基础的用户激励机制，鼓励用户进行评价；为了让用户评价更能反映服务的真实质量，提出用户-服务双向评价用户体系，结合服务对用户的主客观评价，过滤恶意用户评价；在保证用户评价可信的基础上，计算服务信誉需要的人数，以减小计算的复杂度，提出用户人数阈值计算方法，估计达到指定评价可信度所需要的最少人数；为了获取服务信誉，借鉴众包中的质量评估方法，利用 EM 算法计算服务信誉。

基于上述研究目标，本文的主要研究内容围绕以下三个问题展开：

1) 如何鼓励用户做出正确，反映服务真实质量的评论？用户如果不顾用户质量随机做出评论，很可能会误导其他用户，本文提出动态奖励机制，分阶段对用户奖励，给评论准确、评论时间较早的用户更多奖励，促使用户做出正确的评论。

2) 如何过滤恶意用户的评论？本文在用户对服务评价的基础上，提出服务对用户评论的机制，通过服务提供商收集的用户行为信息与自身运行状况给出的客观评价，同时，服务提供商对自身认知与用户的评价的比对给出的主观评价。对主客观综合评价较低的用户进行过滤。

3) 如何计算服务信誉？首先选择可信用户集合，由于用户评论的时序性，选择合适的用户数量不仅影响服务信誉计算的复杂度，同时，会影响服务信誉计算的准确度。本文提出了用户人数阈值计算方法，以最少的用户评论数量作为指标，计算达到指定评论可信度需要的用户人数，并利用 Chernoff bound 对其证明。同时，本文比较众包中质量评估与利用用户评价获得服务信誉的异同，借鉴众包的质量评估方式，将用户信誉作为权重，基于 EM 算法计算服务信誉。

框架的总体流程如图 1-3 所示：

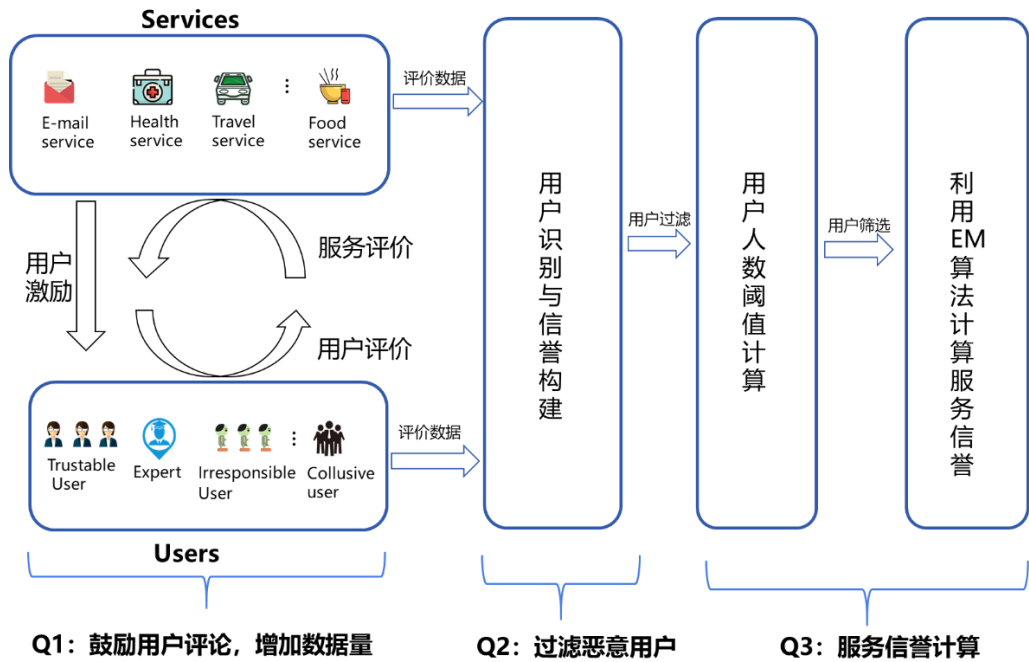


图 1-3 总体框架

框架的核心部分为：用户动态奖励机制，用户-服务双向评价，用户人数阈值计算方法，基于信誉的 EM 评估方法。框架的具体流程如下：

- 1) 奖励评价用户：在现实生活中，用户并不是对所有使用过的服务都会评价，因此，我们提出了对用户的动态奖励机制，按照用户评论的时间、用户评论的稳定性和用户评论的质量，对用户进行分段奖励。
- 2) 在用户使用服务的过程中，收集以人为中心的感知数据与以服务为中心的服务数据。其中包括从服务所在服务器，虚拟机，app 的运行数据。
- 3) 利用收集的信息，让服务对用户进行评价：结合主观因素与客观因素给出对服务对用户的评价。
- 4) 按照服务对用户的评价，分析用户行为，识别用户类型，过滤恶意用户评价，同时，构建用户的信誉。
- 5) 本文根据用户的评价准确率，以及期望的评价准确率，利用用户人数阈值计算方法，计算得到可靠评价所需的最少用户人数。
- 6) 基于 EM 算法，将用户的信誉作为权值，计算服务信誉。

1.3 论文组织结构

本文一共分为六个章节，其中，具体的章节结构如下：

第一章绪论主要介绍本课题的研究背景与意义，接着分析了当前信誉系统研究领域中存在的一些不足和弊端，围绕着这些问题，提出了本文的主要研究内容和目标，最后给出了论文的组织结构。

第二章主要介绍了当前国内外研究现状进行了综述，主要包括 XaaS 理念的提出，服务定义的变化，当前信誉系统的研究现状，以及信誉系统的重要意义，以及当前信誉系统存在的主要问题：数据稀疏与恶意用户。同时，总结了众包领域、服务推荐领域解决数据稀疏，和信誉系统当前如何过滤恶意用户与过滤恶意用户内容的方法。

第三章主要介绍了如何动态设定用户评论的奖励，以达到鼓励用户认真评价的目的；同时，介绍了服务如何基于收集的以用户和以服务为中心的感知数据，建立服务对用户的主客观评价。

第四章主要介绍了如何利用用户-服务双向评价，识别不同类型的用户，并建立用户信誉。同时，比较众包与用户评价的异同，说明了将众包中的质量评估方法应用于服务信誉的获取的原因，以及基于 Chernoff bound 的人数阈值计算过程与证明。最后，介绍了借鉴众包中质量评估的方式，利用 EM 算法计算服务信誉的过程。

第五章主要介绍了实验过程，与本文使用的数据集的来源，对用户与服务的特征进行了分析，并给出模拟方法。为了验证本文提出的动态激励机制的效果，对比了相同预算下，用户评论达到稳定的需要的时间和数量。将本文提出的方法与 HMRep 提出的方法在当前的数据集上进行实验，并将实验结果做了进一步的对比，表明本文提出的方法能够有效的识别恶意用户。最后，本文提出的基于 EM 算法的服务信誉计算方法，能够根据服务质量的变化改变服务信誉。

第六章主要总结了文章的研究内容和贡献，分析了当前研究的不足与待改进之处，并进一步展望了下一步的工作。

第2章 相关工作综述

2.1 XaaS 概述

XaaS 的提出,旨在降低服务提供商的运营成本,降低服务提供商的风险,利用 XaaS,服务的能力极大增强,并且分工明确,互不干扰,底层服务的变化,不会影响上层服务的运行^[2]。XaaS 提出的目的是按照用户使用的资源总量计费,减少服务提供商购买与维护基础设施的费用。就像乘坐飞机,不需要购置飞机,更不需要会开飞机一样^[3]。同时,因为 XaaS 架构下的服务器一般为云服务器,分布在各地,服务的抗灾能力与灾难恢复能力得到了极大的提高。

XaaS 实际上是基础设施即服务(IaaS, Infrastructure as a Service)、平台即服务(PaaS, Platform as a Service)、软件即服务(SaaS, Software as a Service)等^[4]。

IaaS 将基础设施作为服务, IaaS 提出之前,开发人员开发前,往往需要自己购置服务器,来存储服务产生的数据,为了能够将服务提供给其他用户,可能还需要购买网络设备,并且还需要管理服务器和网络设备。通过 IaaS,可以直接购买需要的服务器空间,并且无需负责设备的日常维护^[5]。

PaaS 将平台作为服务,在 IaaS 的基础上,开发人员不仅无需关心服务器,网络设备的安装部署,而且, PaaS 提供了基础的开发和运行环境,开发人员无需操心软件环境的部署安装。利用 PaaS,不仅可以降低运营费用,而且可以避免开发人员使用不同版本的环境带来的问题。同时,底层物理硬件的维护与更新则交给了 PaaS 提供商完成,开发人员可以根据需要自由提升所需资源的规模^[6]。

SaaS 将软件作为服务,在 PaaS 的基础上,无需开发人员。SaaS 的理念是用户无需购买整套软件,而是通过网络,购买线上服务,即可完成用户需求。同时,用户不需担心软件的安全问题,软件的运营,维护,升级,安全保障都由 SaaS 服务提供商来保证,并且 SaaS 具有高度的扩展性,灵活性^[7]。比较常见的将软件作为服务有谷歌云服务,微软的 office 365。

总之,XaaS 的出现,不仅提高了企业研发的灵活性,同时扩展了服务的定义,计算机领域的服务,不再单指 Web 服务,软件可以作为服务,平台可以作为服务,数据可以作为服务,内容也可以作为服务。服务的定义更加多元化,同时,服务的功能与应用范围也得到极大的扩展。

2.2 信誉系统概述

信誉系统在服务计算领域中应用已久^[8]。服务信誉不仅可以辅助用户决策,并且在一定程度上降低用户和服务提供商之间的信息不对称性^[9]。用户更倾向于购买服务信誉高的服务,信誉低的服务往往会通过减低自己的价格来获得用户的信任,因此,服务信誉往往影响到服务的价格制定^[10]。

收集用户对服务的二元评价<满意,不满意>,构建信誉系统是最常见的手段。eBay 现在就是通过收集用户的二元评价,统计用户对服务的满意,不满意个数和,它们的差,就是服务的信誉。这种方法计算简单,但通过 Van 等人^[11]的研究。这种方法帮助信誉高的服务提供商获得更多的用户,同时,信誉越高,服务的价格也越高,这种方法在一定程度上可以帮助用户提前获悉服务。

天猫、亚马逊等网站将用户对服务的评级进行累加求和取平均值来计算服务信誉。Jösang 等人^[12]提出了 Beta 信誉系统,通过统计更新 Beta 概率密度函数来计算信誉。考虑到评分者的可信性,Whitby 等人^[13]改进了 Beta 信誉系统,集成了一个阈值计算方法来确定评分者的报告可靠性,与简单求和系统类似,该系统只接受二进制评级。为了保护用户的隐私,Kersch-Baum 等人^[15]提出了一种对评分保密的信誉系统,通过设计令牌系统,减少恶意用户信誉系统的攻击。周等人^[20]提出了一种基于聚类的声誉评价体系,该体系根据消费者的评级比率将消费者分为诚实和不诚实两类,并利用 Dirichlet 分布来确定声誉。除了利用用户评级来获得服务信誉,还可以利用服务自身的属性,如 Qos 等来计算信誉。Fu 等人^[22]基于 Qos 一致性,构建服务的 Qos 本体,利用 Qos 的实际值与宣传值一致性的度量来计算服务信誉的可靠性,从而避免主观评价可能带来的欺骗行为。Mehdi 等人^[23]建立基于 Qos 的感知信任模型,利用两个统计分布,即 Dirichlet 和广义 Dirichlet 来估计 Web 服务的可信性。这些分布表示多个相关 QoS 度量的结果。前者在 qos 指标正相关的情况下使用,而后者在处理负相关的情况下使用。该方法有效的缓解了恶意反馈对服务信誉的影响。

总之,信誉系统在学术界,工业界都有广泛的应用。一个健壮,准确的信誉系统不仅能够帮助用户选择合适的服务,规避风险,同时,能够帮助服务提供商获得用户信任,提升用户对服务的忠诚度,获得巨大的经济效益。

2.3 数据稀疏解决方法概述

信誉系统中,经常会面临数据稀疏问题。所谓数据稀疏,是指用户与服务构成的二维矩阵,绝大多数值缺失或者为零的情况。造成这种数据稀疏的情况有很

多,在信誉系统中,主要有两类,一类是天然限制造成的数据稀疏,比如用户没有使用这种服务,因此,二维矩阵中此处的值自然是空值。第二类是用户使用了服务,却没有对服务做出评价,造成二维矩阵的缺失。如图 1-2 所示,通过我们爬取的豌豆荚和 google play 的数据,分析来看,对服务进行评级的用户比例远小于总用户数量的 1%。这不仅导致用于计算服务信誉的数据稀疏,这种数据稀疏,经常会导致方法分析出的结论缺乏说服力,因为数据并不是来自多数用户的反馈。

实际上,在众包、推荐系统等领域,也常常面临数据稀疏的问题。本章接下来将介绍众包和推荐系统中数据稀疏的解决办法。

2.3.1 众包中的数据稀疏问题

众包是将一个任务拆分成若干个小的任务,利用互联网等手段,通过大量工人来解决问题。工人指的是愿意完成这些任务,并获取一定报酬的志愿者。众包中经常会面临没有工人接受任务的问题,这和我们提到的数据稀疏问题,有些类似。在众包领域,服务发布方往往通过给予用户报酬来促进用户解决问题。如何设计合理的用户激励机制,促使工人接受任务,并高效的完成,一直是众包领域的研究热点。当前,众包环境下的激励机制的应用比较成熟。博弈论常用于设计激励机制以提升参与者的效益。Yang D 等人^[27] 设计了以任务发布者为中心的激励机制和以用户为中心的激励机制,以任务发布者为中心的工人激励机制是基于 stackelberg 博弈模型,给用户固定数量的奖励,以用户为中心的奖励机制中,基于拍卖模型设计用户奖励,用户会有一个最低的要求奖励。Wang P 等人^[28] 设计了一种激励契约,可以激励服务提供商提供服务请求者所偏好的 QoS 和价格。在激励契约的基础上,提出了一种有效服务选择的激励机制。在该机制中,服务请求者提供一组激励契约给服务提供者,然后基于他们对激励契约的响应引出他们的私人信息。迭代过程直到服务请求者最终获得满足全局 QoS 要求的解决方案为止。Adish 等人^[29] 利用在线学习中的遗憾最小化方法提出了 BP-UCB 机制,并结合采购拍卖和 muti-armed bandits 算法之间的关系设计预算可行、任务请求者效用接近最优且工作者激励相容的机制。另外,AdishS 设计了具有可靠保障的一套机制,它可以用在众包中进行定价。它能够请求者在复杂的市场中自动定价和分配任务。并为在给定预算下最大化任务数量以及给定固定任务数量最小化支付提出了具有持续竞争力的激励相容机制。

2.3.2 推荐系统中的数据稀疏问题

在推荐系统中，经常面临初始推荐时，用户评级数据不足，即所谓冷启动问题^[17]。冷启动问题在推荐领域由来已久，冷启动问题的常见解决方法是结合内容信息和协同过滤来进行混合推荐^[16]。Agarwal 等人^[18]提出了一个基于回归的潜在因素模型，以解决在项目特征存在的情况下，项目推荐的两个问题。通过低阶矩阵分解得到项目的潜在因素。Park 等人^[19]将冷启动问题转化为凸优化问题，而不是传统的矩阵分解问题，来改善冷启动带来的问题。Jamial 等人^[14]提出的 social MF 模型利用用户的社交关系，构造信任网络刻画用户的特征，用于保证冷启动的条件下推荐的准确性。Forsati 等人^[21]在融合信任关系和不信任关系的基础上，构建了基于矩阵分解的社会化推荐模型，从而大大改善了推荐的质量，改善了数据冷启动的问题。神经网络也被用来解决推荐问题中的冷启动问题，He 等人利用卷积神经网络，从数据中学习神经网络函数，然而，该方法需要大量的物品的信息才能实现冷启动条件下的物品推荐。Sang^[25]等人在线社交网络中（OSN），基于用户行为的动态性，考虑用户在不同社交媒体上的兴趣来进行推荐。

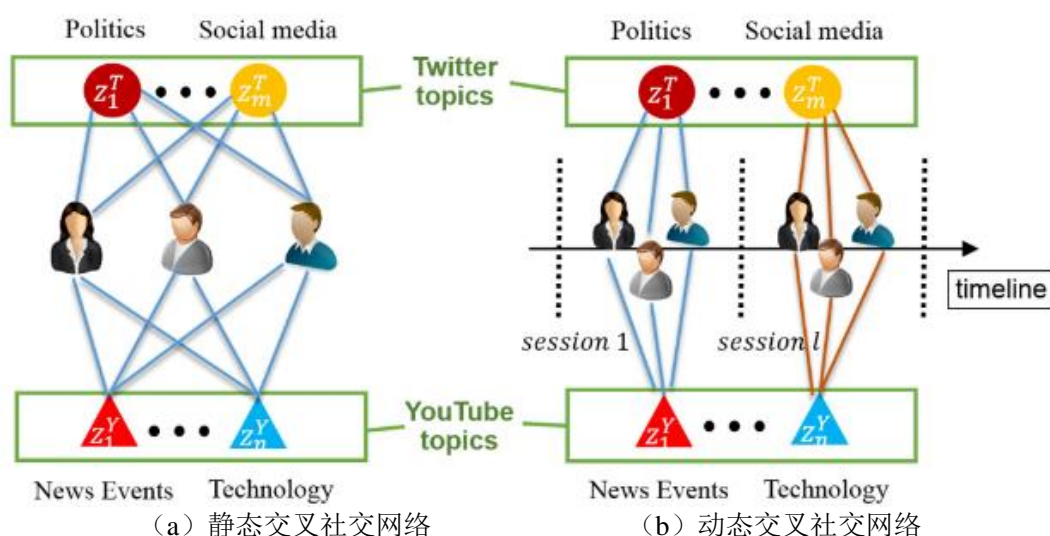


图 2-1 静态与动态交叉社交网络的对比。

如图 2-1 所示，传统的方法将用户在不同社交网络的行为看做一个整体，每个 OSN 中的用户行为都是不同的主题，但是在右侧的动态交叉社交网络中，按照时间段对用户的兴趣进行划分，可以发现，在每个小的时间段内，用户的行为模式是一致的，文章通过探究不同社交网络中共享的用户短时间内的行为关联，来解决推荐过程中的冷启动，即根据用户在当前社交网络上的兴趣推荐在另一个社交网络上感兴趣的话题，从而改善冷启动问题。朱宇等人^[26]使用 active learning,

基于物品的属性和用户的历史行为,设计了用户选择标准,通过利用用户的反馈,用户的历史评分以及物品属性,对用户的评分进行预测,有效的缓解了冷启动问题。Zhou^[42] 等人利用用户的隐藏信息来解决用户数据稀疏的问题。通过隐藏信息,对未对服务评价的用户赋予信任值,并提高服务排序稳定性的收敛速度。

2.4 恶意用户生成内容过滤

用户生成内容是指在电子商务平台上,用户在使用、消费服务或商品时产生的内容,其中,包括用户的评论,用户评级。一直以来,用户生成内容对于带动服务发展,提高服务的品牌效应,帮助服务改进用户体验都具有重要的作用。然而,用户的生成内容中,往往包含一些恶意、垃圾信息。如何去除用户生成内容中的恶意,不准确信息一直是研究者们关心的问题。国内外学者也利用不同的方法对恶意用户内容进行过滤。本文将恶意用户生成内容按照数值和文本分为两个部分:不实评级和垃圾评论。接下来将分别介绍当前不实评级与垃圾评论的过滤方法。

2.4.1 不实评价过滤

单明辉等^[33] 基于用户评价的相似度来判断评价是否可信,他们假设数据量越大,用户对服务的偏差越小,将用户对任意服务的可信度定义为用户好评占总评价的比例,并用平均偏离度描述两个用户对服务意见的偏差程度。以平均偏离度为指标,一个正常用户与另一个用户的偏差越大,该用户越有可能为恶意用户。因此,若一个用户与其他用户的平均偏离度越大,其评价可信度越低。

Wang^[34] 等人利用 CUSUM 方法进行检测,CUSUM 方法是工业中用于检测产品质量与标准质量偏差的函数,它来源于时序改变点检测理论,它在异常检测中有广泛的应用。它主要通过通过检测指定时间内反馈水平的和与正常情况下反馈水平的平均阶跃来发现恶意反馈 CUSUM 方法的优点在于能够以连续方式监测用户反馈,从而达到实时检测目的。然而,它在用户反馈波动较小时,作用比较明显,但是,一旦波动较大,它的检测能力就大大降低。

王尚广^[35] 等人提出反馈检查,反馈校正,反馈校正三个步骤减小服务选择的偏离度,提高服务信誉度评价的客观性。其中,通过构建用户档案记录用户的反馈历史,并将用户的反馈能力通过权值进行划分,针对用户偏好或者上下文的影响,文章通过层级模糊系统,通过上下文推理,计算不同用户在不同环境下的反馈和谐度,实质还是通过各项指标的相似性计算,然后将该用户反馈数据的权重计算其虚拟反馈,最后通过与用户实际反馈进行对比得到反馈冲突度,按照区

间范围进行不同的处理。针对恶意检测，文章同样利用了 CUSUM 方法，对恶意反馈进行检测。

Wang 等人^[40]也是利用用户评价的相似度来判断用户评价的可信度。不同的是，文章考虑用户评价的多维性，即用户对服务的评价不仅有总体的评价，还包含对服务不同属性的多维度的评价，比如，用户对服务的可用性，安全性等不同属性的评价。论文提出多维高可用服务信誉评价框架（HMRep, High-reliability Multi-faceted Reputation Evaluation Mechanism）。文章通过对比当前用户与其他用户在每个维度上的评价相似度，将评价相似度低的用户归为可疑用户，一旦可疑用户的评价相似度低于某一阈值，即被判定为恶意用户。恶意用户的评论即被舍弃。本文主要通过与 HMRep 框架的对比，来比较提出的模型在恶意用户识别能力的效果。

2.4.2 垃圾评论过滤

关于垃圾信息检测由来已久，最初的研究工作集中在垃圾邮件的检测上。Meda 等人^[31]根据 62 个用于检测垃圾邮件的特征，使用支持向量机在 Twitter 数据集上利用随机森林算法查找其中的恶意邮件。Wu^[32]等人提出了基于深度学习的层次分析法，通过 wordToVector 模型学习 tweet 的语法，并利用 Twitter 数据构造二值分类器，以便过滤恶意邮件。Tang 等人分析微博中垃圾邮件发送者的特征，选取微博内容相似度、平均链接数等 12 个特征，构建一个特征向量，采用集合学习的方法将多个基分类器结合起来，基于模糊逻辑的过采样和代价敏感的支持向量机被认为是在数据和算法层面上处理不平衡的数据。

Mishne 等人^[43]通过自然语言模型检测博客评论总的垃圾信息，他们利用机器学习的方法，推测恶意评论中的特定词汇出现的概率，同时，他们注意到垃圾评论经常会包含一些链接，将用户从当前页面导流到其他网站。Dong 等人^[44]利用自然语言处理的方法，检索亚马逊用户评论中二元语言模型（Bi-gram），并利用 Bi-gram 提取评论的主题，之后，利用主题的相关性来过滤垃圾信息。Li J 等人^[45]通过对用户文本评论中的情感词进行统计分析，从情感极性角度分析文本的语义，表明虚假评论文本中包含更多的情感词。除此之外，自然语言处理的相关方法也经常被用来判断虚假评论。Cristina 等人利用 NLP 的相关方法，首先消除不连续的文本，将非 ascii 字符消除，同时，提取不同评论的特征，使用监督学习的方法来判断当前评论是否为垃圾评论。

总之，研究人员主要利用两种方法对信誉系统的不实评价进行过滤：第一类，用户评级的相似性。第二类：利用机器学习的方法对用户评级分类。然而，这两种方法都没有考虑随着 XaaS 的提出，服务自身功能的增强，服务比以前可以获

得更多用户行为信息，因此，本文提出利用服务对用户进行评价，同时，正如前文所述，当前利用用户评论获取用户信誉，存在评论数据稀疏的问题，因此，本文提出了基于用户评论质量的动态奖励机制，鼓励用户评论，提高用户评论数量。

第3章 动态奖励与用户-服务双向评价机制

如图 3-1 所示，本章主要内容分为两部分：基于用户评论质量的动态奖励机制与用户-服务双向评价机制。基于用户评论质量的动态奖励机制通过用户的评论时间与评论质量对用户的评论进行动态奖励，以达到增加提高评论质量，反映用户评论质量的目的。用户-服务双向评价机制通过收集以用户和以服务为感知中心的数据，在对数据进行一定修正后，让服务通过用户的评论、用户的行为和服务自身运行状态分别对用户给予主观和客观评价，并将它们结合，即为服务给用户的评价。

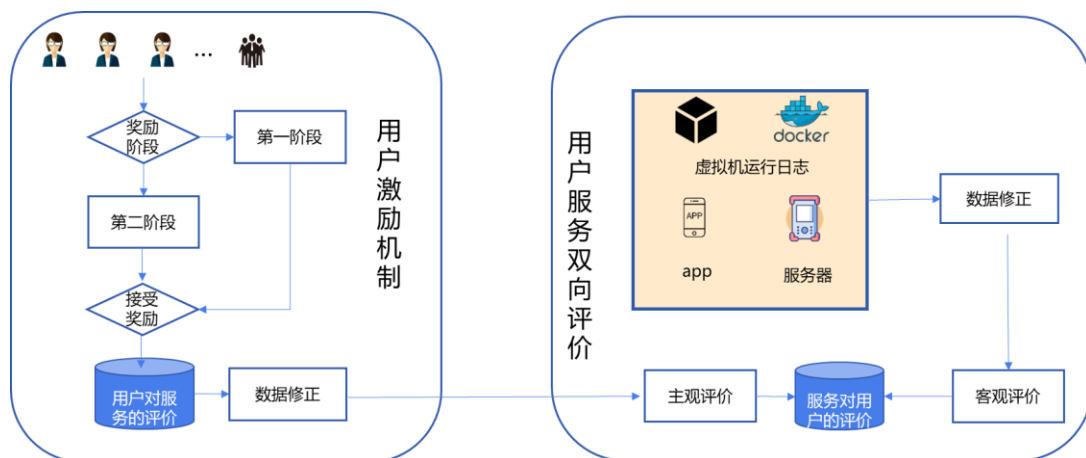


图 3-1 基于双向评价的用户识别与信誉计算流程

3.1 基本概念与变量解释

为了便于理解，本节将对文中的一些定义及变量进行解释。

定义 1（以用户为感知中心的数据）：以用户为感知中心的数据在群体智慧研究中的定义为利用个体或者社区共同感知信息以形成知识片段的新的感知模式。在本文中，指的是用户可以直接获得的数据，主要为用户评论数据。

定义 2（以服务为感知中心的数据）：以服务为感知中心的数据在本文中指的是通过服务收集的所有数据，它包含以用户为感知中心的数据，为了区分，本文设定以服务为感知中心获得的数据是用户不可直接获得的数据。

本文用到的变量与其含义如表 3-1 所示：

表 3-1 变量符号及解释

变量符号	变量定义
U	用户集合，存储所有用户信息
CU	可信用户集合
MU	恶意用户集合
COU	水军用户集合
IU	不负责任用户集合
u_i	用户 i , $u_i \in U$
S	服务集合，存储所有服务信息
UID	用户 ID，唯一标识
registerTime	注册时间
usedTime	用户对某一服务的使用时间
friendCount	用户的好友数量
feedbackCount	用户的评论数量
rank	用户等级
us	服务使用时间
win	时间窗口的值
diffCR	当前时间和注册时间的差值
cost	用户在时间窗口内使用服务的总花销
errorCount	服务在时间窗口内日志中错误的个数
τ	固定值，用来调控用户好友总数与用户的好友总数增长值对用户的影响。
$score_i(s_j)$	用户 i 对服务 j 的评分调整值
\overline{score}_i	用户 i 的评分均值
C	常数，控制用户评分调整的幅度
oe_i	服务对用户 i 的客观评价值
se_i	服务对用户 i 的主观评价值
$S_{m \times 8}$	用户的服务感知数据组成的矩阵
$ES_{m \times n}$	服务对用户的评价矩阵
K	阈值
r_i	用户 i 的信誉值
Cycle	轮次，其中第 k 轮的时间窗口为 win
$R_{m \times n}^k$	第 k 轮用户对服务的评级矩阵

rs_j	服务 j 的信誉
--------	----------

定义 3: 对于 s_j , 评级 r 出现的次数为 $h_j(r, k)$, 它的计算方法如下所示: 其中 $I(score_i(s_j) = r)$ 是判断用户 i 给服务 j 的评级是否是 r , 如果是, 次数加 1, 否则, 次数不变。

$$h_j(r, k) = \sum_{i=1}^m I(score_i(s_j) = r) \quad (3-1)$$

定义 4: 对于 s_j , 评级 r 的相对频率为 $f_j(r, k)$

$$f_j(r, k) = \frac{h_j(r, k)}{\sum_{r' \in [minScore, maxScore]} h_j(r', k)} \quad (3-2)$$

定义 5: 服务评级的相对频率分布, $\vec{f}_j(k)$ 表示服务 j 在用户对其第 k 次评价时的相对频率分布, 其第 p 个元素表示评级 r 的相对频率

$$\vec{f}_j(k)[j] = f_j(r, k) \quad (3-3)$$

本文将部分数据以表的形式存储, 本文用到的表及其含义如表 3-2 所示:

表 3-2 相关表名与其含义

表名	定义
User	用户属性表, 以 UId 为标识
friendship	用户的好友表
session	用户 Session 表, 用户每次使用服务, 存储用户信息
review	用户评论表

由图 3-2 可知, 传统的服务信誉计算方法是通过用户评价矩阵, 汇总用户对服务的满意度, 计算服务的信誉度。本文是通过服务对用户的评价, 先计算用户的信誉, 过滤到一部分恶意用户的评价, 之后, 再计算服务的信誉。可以看出, 本文提出的方法的优势在于:

- 1) 相比用户对服务的评价存在稀疏性问题, 由于服务会对每个评价过它的用户进行评估, 因此, 不存在稀疏性问题。
- 2) 为了减少用户评论数据稀疏, 本文提出基于用户评论质量的动态奖励机制, 增加用户评论数量。
- 3) 服务对用户的评价数据来源广, 可以利用服务的主客观评价过滤恶意用户, 获得更为准确的服务评价。

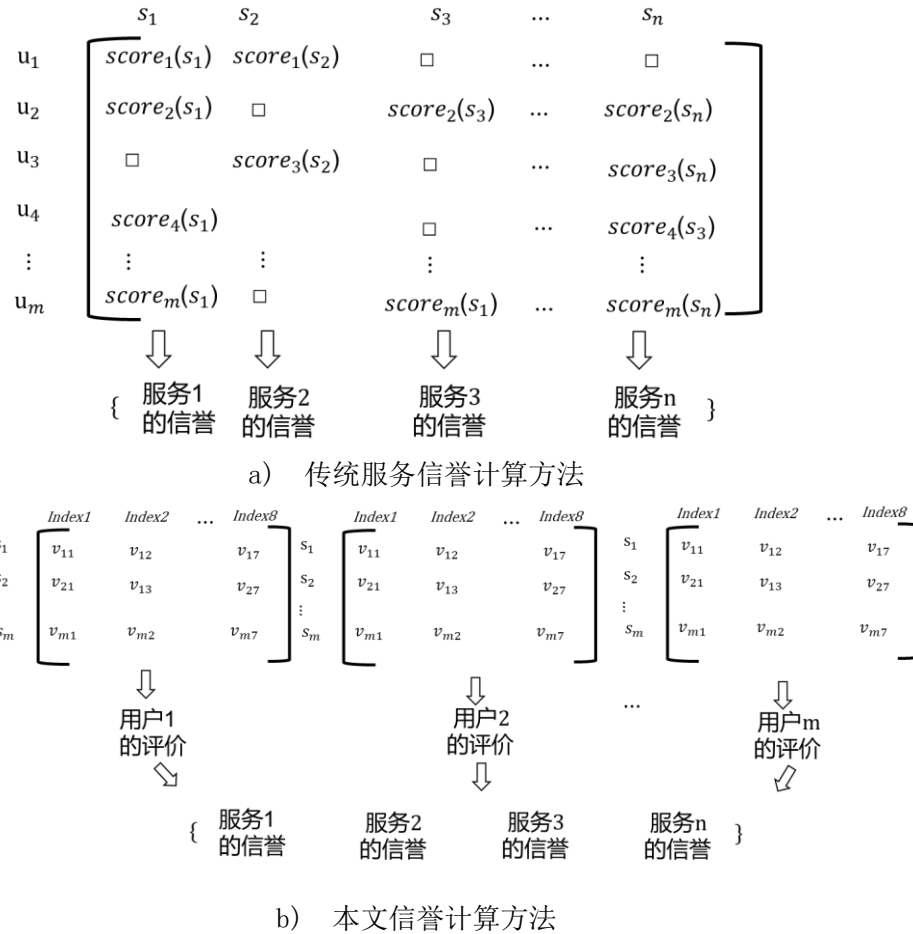


图 3-2 传统服务信誉计算方法与本文信誉计算方法的比较

3.2 基于用户评论质量的动态奖励机制

考虑图 1-2 出现的用户使用服务的数量远远大于用户评论服务的数量。主要是因为用户对服务进行评价，属于用户自发行为，服务提供商无法强迫用户对服务做出评价。本文基于博弈论构建用户激励机制，意在刺激用户尽早的给出高质量的用户评价，鼓励用户评论，方便其他用户决策。减轻评论数据稀疏的问题。

3.2.1 现有用户评级激励机制分析

用户是评价内容的生产者，同时，也是评价内容的使用者，用户需要借用其他用户对服务的评论来判断当前服务的可靠性。在无奖励机制的情况下，用户对服务进行评价，无法获得任何奖励。然而，用户对服务评价会花费时间、精力，一般而言，理性用户不会对服务做出评论，只有利他型用户会在没有奖励的情况下对服务做出评价；如果用户做出评论，服务给出固定数量的奖励：在这种情况下，如果用户评论获得的奖励大于他的时间成本，理性用户会对服务评论。然而，

在这种情况下，由于奖励固定，用户随意评论与认真评论的收益相等，用户很可能不会对服务随意评论，导致最终的服务评论未能反映服务的真实情况。为了提高用户评论的积极性，促使用户尽早做出评论，方便服务（提供商）吸引更多用户，本文提出基于用户评论质量的动态奖励机制。

3.2.2 激励机制的博弈分析

首先，博弈的双方分别为：服务和用户。其中，服务的收益为用户认真评论带来的收益减去支付给用户的激励费用。用户的收益为评论服务，服务给的奖励与时间成本的差值。当前，服务的可选策略为不设置奖励，固定奖励，动态奖励。假设服务生态系统中的用户都为理性用户，用户策略的集合为不评论，随意评论，认真评论。本文假设理性用户会从自己利益最大化的角度选择合适自己的策略。

假设服务（提供商）给用户的最大奖励为 A ，用户认真评论所用的时间成本为 C ，认真评论对服务的收益为 P ，随意评论的成本为 C' ($C' < C$)，随意评论对服务的潜在收益为 P' ($P' < P$)，动态奖励额为 A' ($C < A' \leq A$)，则用户与系统之间的博弈矩阵为表 3-3 所示：

表 3-3 用户-服务博弈矩阵

	不评论	认真评论	随意评论
无激励	(0, 0)	($P, -C$)	($P', -C'$)
统一激励	(0, 0)	($P - A, A - C$)	($P' - A, A - C'$)
动态激励	(0, 0)	($P - A', A' - C$)	($P' - A', -C'$)

括号中的第一个数字代表服务的收益，后面的数字代表用户的收益。利用划线法对博弈进行均衡分析，可以得出，当服务对用户无激励政策时，用户不评论的收益最大，当服务对所有用户的奖励都一样的时候，用户随意评论的收益最大，当采用动态奖励的方式时，用户认真评论的收益最大，同理可得，服务在用户不评论时，采用任意策略收益相同。当用户认真评论时，服务采用动态奖励收益最大，当用户随意评论时，服务不对用户激励收益最大。由上述分析可知，动态鼓励，认真评论是服务和用户的最佳策略。

3.2.3 动态奖励机制

如图 3-3 所示，用户可得的奖励按照评论的不同阶段来计算，本文将用户获得的奖励分为服务提供的奖励(SA, Service Award)与用户实际获得的奖励(RA, Real Award)。

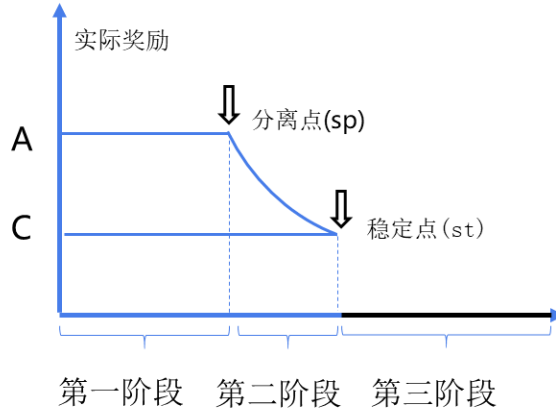


图 3-3 用户动态奖励函数

第一阶段：由于服务上线，或者更新的初期，我们难以知道服务的真实质量，所以，在服务的真实质量没有达到稳定状态之前，无论用户的评论质量如何，如公式 3-1 所示，用户评论的激励统一设置为 A 。

$$SA = RA = A, k < b \quad (3-4)$$

第二阶段：为了将用户的奖励与评论质量统一起来，本文按照评论质量与评级的稳定状态对用户评论进行奖励。用户获得的奖励如公式 3-2 所示，其中， ES_{ij} 是用户 i 收到的服务 j 的评价，关于 ES_{ij} 的计算详见算法 2。

$$RA = ES_{ij} * SA_{ij} \quad (3-5)$$

SA_{ij} 是用户 i 根据当前用户评级稳定程度获得奖励，其中， SA_{ij} 的计算方法如公式 3-3 所示

$$SA_{ij} = \left(\frac{\varphi - v_i(c, k)}{\varphi - v_i(b, k)} * (A - C) + C \right) \quad (3-6)$$

其中， C 是预估的用户认真评价的成本， A 是服务对单个用户提供的最高奖励， φ 为常数，用来表示服务评级稳定时的常量值。 $v_i(c, k)$ 表示评论计算窗口大小的值为 k 时，第 c ($c > k$) 个用户评论后，当前服务评级的稳定性，它的计算方法如公式 3-4 所示：

$$v_i(c, k) = \frac{\sum_{p=k-w+2}^k \text{sim}(\vec{f}_j(p) - \vec{f}_j(p-1))}{k-1} \quad (3-7)$$

计算窗口大小为 k ，则服务在收到 c 次评价后 ($c > k$) 次评价后，服务 i 的评级变化 $v_i(c, k)$ ， $v_i(c, k)$ 的值越大，表明用户对服务的评级越稳定。其中， sim 函数用来计算向量的相似度，本文直接用欧式距离公式计算， $\vec{f}_j(p)$ 的计算见定义 5。

第三阶段：由于用户对服务的评价已经达到稳定，除非服务更新，服务信誉不会变化。因此，在达到第三阶段后，将不会对用户进行奖励。

3.3 用户-服务双向评价机制

由定义 1, 2 可知，服务不仅可以通过对比自身运行状况与用户评价的一致性，还可以通过分析用户行为数据来评估用户评价的可信性。在用户与服务交互的过程中，服务对用户的评价将帮助服务生态系统筛选出合理的用户群体，提高系统中用户群体评价的公正性。基于这个理念，本文提出“用户-服务”双向评价模型来提高服务信誉计算的可靠性，利用服务对用户的评估，过滤恶意用户的评价，调整正常用户因为偏好等原因产生的误差，用户服务双向评价模型图 3-4 所示，它主要包括两个部分，用户评价服务与服务对用户的评价。几个环节：以服务上和以用户为中心的数据获取，数据的修正与调整，服务评价等过程。

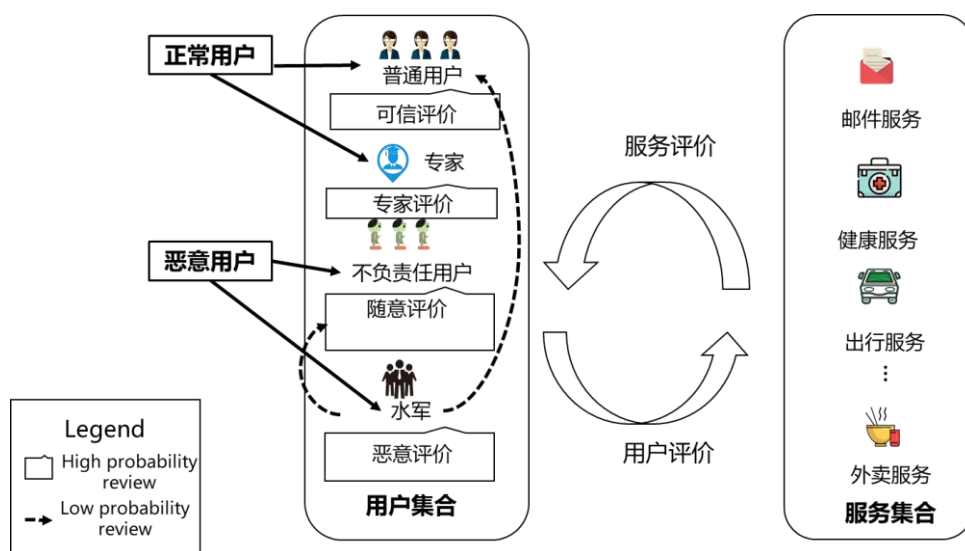


图 3-4 用户-服务双向评价模型

3.3.4 以服务为中心的数据获取方法

本文在用户评级的基础上，利用服务收集的用户行为数据与自身运行数据构建服务对用户的评价机制。如前文所述，服务需要收集的数据分为两种：以人为感知中心的数据和以服务为感知中心的数据。以人为感知中心的数据主要以用户评级和评论数据为主，收集过程比较简单，这里不再赘述。本文收集的以服务为中心的感知数据是一个八元组： $\langle \text{UID}, \text{registerTime}, \text{usedTime}, \text{friendCount}, \text{feedbackCount}, \text{errorCount} \rangle$ 分别代表用户的注册时间，使用服务的时间，用户的

好友数量, 用户的评价次数, 服务出错个数。以服务为中心的数据用户行为数据的建立与收集方法如算法 3-1 所示:

算法 3-1: 以服务为中心的数据获取方法

输入: U: 用户集合, S: 服务集合

输出: $S_{m \times 8}$: 以服务为中心的感知数据构成的矩阵

```

1: when u enter service ecosystem: // 当用户进入服务生态系统
2:   Insert(UID, createTime) into User
3: when user i is followed by other user j and user I have followed user j:
4:   update friendship set friendCount = friendCount + 1 where UID = i
5:   update friendship set friendCount = friendCount + 1 where UID = j
6: when detect session of user:
7:   Insert into session(UID, SID, startTime, expireTime)
9:   if current time < startTime + expireTime:
10:    if user operate:
11:      update Session set expireTime = expireTime + delayTime
12: when s encounter an error:
13:   Insert into serviceLog(SID, errorMessage)
14: when user review:
15:   update review set score = userscore where UID = user.UID
16: for u in User:
17:   registerTime ← select createTime as registerTime from user where
                        UID = u.UID
18:   usedTime ← select sum(expireTime- startTime) as usedTime from
                        session where UID = u.UID group by UID
19:   feedbackCount ← select count(review) from review where UID = u.UID
20:   errorCount ← select count(SID) from serviceLog and session where
                        serviceLog .SID = session.SID and session.UID = u.UID
21:    $S_{m \times 7} \leftarrow \langle \text{UID, registerTime, usedTime, friendCount, feedbackCount,}$ 
                        errorCount>
22: end for
23: return  $S_{m \times 7}$ 

```

算法 3-1 的 1 到 15 行是收集通过服务评价所需的数据,第 2 行是在服务进入生态系统后,记录用户的进入时间,并用 UID 作为用户的唯一标识,第 3-5 行是两个用户互相关注时,将两个用户的好友各自增加 1,6-11 行是在用户操作服务时,记录用户的使用时间,本文利用用户的登录状态 session 来判断是否在使用服务,当用户一段时间没有使用服务,那么注销用户的 session。12-13 行,是用来表示服务时,记录运行时出现的错误。16-21 行是统计以服务为中心的数据,22 行是所有用户的服务感知数据组成的矩阵 $S_{m \times 8}$ 。

3.3.5 数据的修正与调整

服务对用户评价之前,需要对收集的数据进行整理、标准化、去除不同的单位对评价的影响。本文分别按照以人为中心和以服务为中心的感知数据对 3.4.1 中收集的数据中进行处理。

以人为中心的数据主要以用户评级数据为主。考虑到不同用户的评分尺度不同,即使两个服务的评分相同,评分代表的用户意见不一定相同。

表 3-4 用户评价实例

	用户 A	用户 B
服务 C	4	
服务 D		4
服务 E	3	5
服务 F	3	5
服务 G	3	5
平均分	3.25	4.75

如表 3-4 所示,用户 A 与用户 B 分别对服务 C、D 评价,通过用户 A, B 的历史记录可知,用户 A 属于严格型用户,用户 B 是宽松型用户。虽然用户 A、B 对服务 C、D 给出的评价等级是相同的,但是评分严格的用户 A 给出的分数要比评价宽松的用户 B 给出的分数权值更高。为了降低不同用户的评分尺度对用户的影响。本文对每个用户的评分按照其评分的最大值和最小值出现的频率进行归一化处理。

$$score_i(s_j) = \begin{cases} minScore + (score_1(s_j) - minScore) \frac{\overline{score_1} - Min + C}{score_i - Min + C} & \text{if } \overline{score_1} < score_i \\ maxScore - (maxScore - score_1(s_j)) \frac{maxScore - \overline{score_1} + C}{maxScore - score_i + C} & \text{else} \end{cases} \quad (3-8)$$

以服务为中心的感知数据是一个八元组,对矩阵的每一行元素(除 UID 以外)进行标准化处理,以消除不同单位的影响。如公式 3-9 所示:其中, s_{ij} 是用户 i 第 j 项属性的数据值。

$$S_{ij} = \frac{s_{ij} - \min\{s_{1j}, \dots, s_{nj}\}}{\max\{s_{1j}, \dots, s_{nj}\} - \min\{s_{1j}, \dots, s_{nj}\}} \quad (3-9)$$

3.3.6 服务评价

基于以人和以服务中心的感知数据,将服务对用户的评价划分为主观评价与客观评价。服务对用户的评价算法流程如算法 3-2 所示。将收集到的用户数据矩阵 $S_{m \times 8}$, 用户评价表 R 作为参数,首先,计算服务对用户的客观评价,它的主要计算方法是利用熵权法求得收集数据的权重,然后求得每个用户的客观评价。其中,第 3 到 6 行是利用熵权法求权重的过程。其次,根据用户对服务的评价,服务判断评价与自身质量是否符合,给出主观评价。最后,将主客观评价结合,其中权重初值设为 0.5。4.1.3 节将详细讲解权重如何根据服务主观评价的准确性改变。

算法 3-2: 利用服务对用户评估算法

输入: S : 服务集合; R : 用户评级

输出: $ES_{m \times n}$: 服务对用户的综合评价矩阵, $SES_{m \times n}$: 服务对用户的主观评价矩阵,

1. **for each** s_j **in** S :
2. **for each** $s_{ij} \in S_{m \times 7}$ **do**
3. $p_{ij} \leftarrow \frac{s_{ij}}{\sum_{i=1}^m s_{ij}}$
4. $e_j \leftarrow -k \sum_{i=1}^n p_{ij} * \ln(p_{ij})$
5. $d_j \leftarrow 1 - e_j$
6. $w_j \leftarrow \frac{d_j}{\sum_{j=1}^m d_j}$
7. $oe_{ij} \leftarrow \sum_{j=1}^m w_j * p_{ij}$
8. $se_{ij} \leftarrow \text{giveScore}(s_j, r_i)$
9. $SES_{m \times n} \leftarrow se_{ij}$
10. $ES_{ij} = w * se_{ij} + (1 - w) * oe_{ij}$
11. **end for**
12. **end for**
13. **return** $ES_{m \times n}$

3.4 本章小结

本章首先在 3.1 对模型中使用的定义与变量进行阐述, 3.2 节介绍了动态激励机制的构造方法与理论基础。3.3 节中介绍了如何构建用户服务-双向评价机制, 最后, 3.4 节对本章的内容进行了总结。

第4章 基于双向评价的服务信誉计算

通过第三章我们可以知道，通过构建用户-服务双向评价机制，利用服务对用户的评价，可以建立用户的信誉体系。然而，本文的最终目的是获取可信的用户评价，进而计算服务信誉，反映服务质量，辅助用户决策。如图 4-1 所示，基于用户信誉的服务信誉计算整体流程如下：

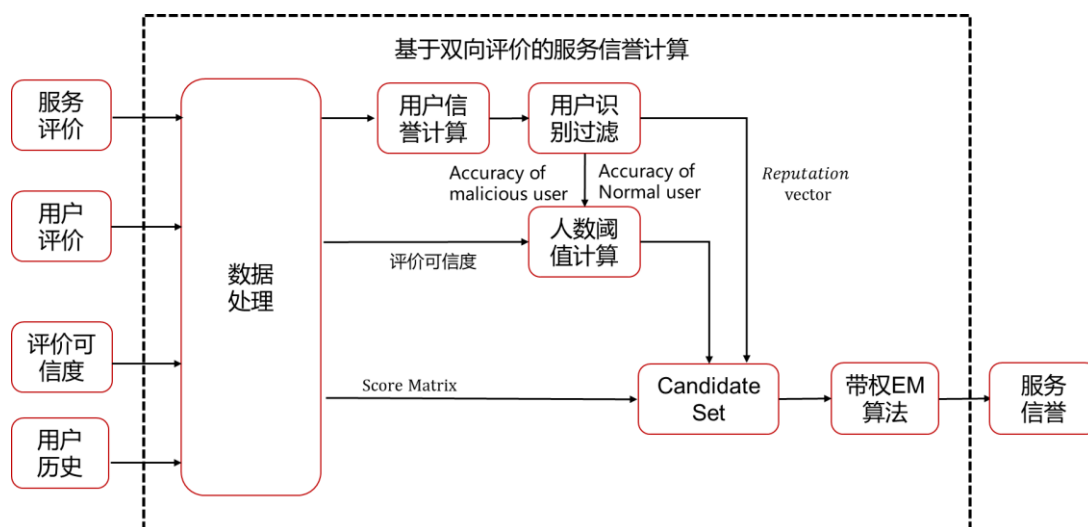


图 4-1 基于双向评价的服务信誉计算

- 1) 将用户评价，服务评价，期望达到的评价可信度，以及用户的历史数据作为输入。利用用户-服务双向评价机制中计算出的服务对用户的评价矩阵，计算用户信誉，并将低信誉的用户过滤，获得可信用户集合。
- 2) 根据可信用户的历史评价记录，计算他们的评价准确率，并利用 Chernoff bound 计算用户人数阈值，计算达到指定准确率所需的最少用户人数，缩短获得服务真实质量所需的时间。
- 3) 对用户评价进一步筛选后，借鉴众包中的质量评估方法，将用户信誉作为权值，使用 EM 算法计算服务信誉。

4.1 基于双向评价的用户识别与信誉值更新

由 3.1 可知，本文建立的用户-服务双向评价机制，相比单纯利用用户评论建立信誉评价机制，以服务为中心的感知数据的来源广且密集。本文利用服务对用

户的评价识别用户。应当注意的是，服务对用户的评价未必准确，为了解决服务评价偏差问题，本文通过两方面来保证服务评价的准确性，第一，在利用服务评价构建用户信誉之前，根据服务评价的偏差，过滤掉一部分服务评价。第二，服务对用户的评价是由主观评价与客观评价组成，每经过一段时间，通过核定服务主观评价的准确性，调整服务主观评价的权重。

算法 4-1 是用户类型识别算法，用以区分正常用户和恶意用户。本文假设多数服务会给用户正确的评价，算法 4-1 的第 1 行，是过滤对用户偏差较大的服务，具体的过滤方法如算法 4-2 所示，算法的 2-13 行是通过迭代的方式，获取服务对用户的综合评价。其原理是计算服务用户评价集合的质点，使得这个点距离所有评价的距离最小。14 行是根据设定的阈值，对低信誉的用户过滤，并将其加入到恶意用户集合中。

算法 4-1: 用户类型识别算法

输入: ES: 服务对用户的评价集合

输出: 正常用户集合, 恶意用户集合

```

1. Filter(ES)
2. for each  $u_i \in ES_{m*n}$  do
3.    $r_i^{start} \leftarrow \text{average}(es_i)$ 
4.    $r_i^{end} \leftarrow r_i^0 + 1$ 
5.   While true do:
6.     if  $r_i^{end} - r_i^{start} < \nabla$ :
7.       break:
8.     else:
9.        $\text{mid} \leftarrow r_i^{start} + (r_i^{end} - r_i^{start}) / 2$ 
10.      if  $(\text{distance}(r_i^{start}, es_i) < \text{distance}(r_i^{end}, es_i))$ 
11.         $r_i^{end} \leftarrow \text{mid}$ 
12.      else:
13.         $r_i^{start} \leftarrow \text{mid}$ 
14.      if  $r_i^{end} > K$ :
15.        CU.add( $u_i$ )
16.      else:
17.        MU.add( $u_i$ )
18.      end if
19. return CU, MU

```

算法 4-2 主要利用服务评分的偏差来判断某个服务是否可信。对于任意用户 i ，服务 j 对它的评价等于

$$es_{ij} = r_i + deviation(s_j) \quad (4-1)$$

那么，对于任意两个服务对用户的评价 es_{ij} ， es_{ik} ，他们对用户的评价偏差值应为：

$$\begin{aligned} es_{ij} - es_{ik} &= r_i + deviation(s_j) - (r_i + deviation(s_k)) \\ &= deviation(s_j) - deviation(s_k) \end{aligned} \quad (4-2)$$

如果服务 s_j 与服务 s_k 的评价都是根据用户的评价质量来对服务进行评价，那么，两个服务 s_j 与 s_k 的偏差应该较小，反之，如果 s_j 与 s_k 有一个为根据用户评价质量来对用户评价，那么偏差就会较大，因此，本文利用服务评价的偏差来过滤服务评价。算法 4 的核心是通过公式 4-2 计算当前服务与其他服务的偏差度。其中 $|S|$ 是用户 i 评论的所有服务数量。 $diff$ 是服务对用户 i 的评价偏差。

$$diff = \sqrt{\frac{\sum_{j \neq k} (es_{ij} - es_{ik})^2}{|S| - 1}} \quad (4-3)$$

如果偏差度大于某一指定阈值 Δ ，该服务评价将被过滤出去。

算法 4-2：服务过滤算法

输入： ES：服务对用户的评价集合

输出： ES：可信服务评价集合

1. **for** each $es_j \in ES_{m \times n}$:
 2. $diff \leftarrow \sqrt{\frac{\sum_{j \neq k} (es_{ij} - es_{ik})^2}{|S| - 1}}$
 3. **if** $diff > \Delta$
 4. Filter(es_j , ES)
 5. **else:**
 6. continue:
 7. **end if**
 8. **end for**
 9. **return** ES
-

4.1.1 用户行为分析

用户对服务评价时,由于背景和专业知识不同,加上自身偏好和环境的影响,使得评价的质量难以保证。用户行为的不确定性导致用户评价质量难以控制,其他用户在参考用户评价时,首先需要判定评论用户所属类型以判断用户给出的评价是否可信,因此,有必要根据用户行为对用户类型分析,为用户决策提供参考。

结合 Wang 等人的工作^[40],按照用户评价与服务质量的一致性,本文将用户分为两类:正常用户与恶意用户。其中,正常用户又可以按照领域认可度分为普通用户和专家用户。相较于专家而言,普通用户的评价专业度与可信度要低于普通用户。恶意用户按照是否受人雇佣可以分为不负责任用户与水军用户。不负责任的用户是指不受他人雇佣,对于服务随机评价的用户。水军用户是指受人雇佣,对特定服务给出极值评价的用户(极大值提高雇佣者的声誉,极小值降低竞争者的声誉)。

4.1.2 水军与不负责任用户识别

本文通过 DBSCAN 聚类算法对所有恶意用户聚类。用户数量最大的簇为水军用户,剩余的是不负责任用户,如算法 4-3 第 11 到 12 行所示。DBSCAN^[37]算法是基于密度的聚类算法,其不需要事先指定聚类的数目就可以把距离较近位置的点聚类成一组。使用该算法来对用户进行聚类,可以避免 k-means 等聚类时需要指定选择初始聚类中心和聚类簇数 k 对聚类结果影响。

算法 4-3: 水军与不负责任用户识别算法

输入: R: 服务对用户的评价集合, MU: 恶意用户

输出: COU: 水军集合, IU: 不负责任用户集合

```

1.  for u in MU:
2.      ru ← select score from R where UID = u.UID
3.      Score[u.UID] ← ru
4.  end for
5.  Cluster = DBSCAN(Score)
6.  COU ← getClusterMaxCount()
7.  IU ← MU - CS
8.  return COU, IU

```

4.1.3 信誉值与权值更新

用户的信誉值更新方式如下：其中 λ 是用户历史信誉所占的比重， r_i^{end} 是用户指定时间窗口下，所有用户评价过的服务对用户的评价。

$$r_i = (1 - \lambda) * r_i + (\lambda * r_i^{end}) \quad (4-4)$$

为了保证服务主观评价的准确性，减少服务评价的偏好用户，当评级达到稳定的时候，要对服务的主观权重进行更新。本文利用算法 4-4 对服务评价用户的主客观权重更新。第 1-6 行是判断当前用户对服务 j 评论是否达到稳定。第一行在给评级次数 c 下，本文利用评级的变化程度来判断用户评级是否达到稳定。计算方法如公式 3-7 所示， $v(c, k)$ 的值越大，则说明评级稳定性越高，否则越小。算法的第 3 行和第 4 行分别计算某个评级的稳定性与频率，具体可见定义 3.4。将稳定时的评级记为 R_r 。第 3 行是计算服务 s_j 的评级 score 出现的频率，第 4 行是服务 s_j 出现评级 score 的相对频率。第 7 行开始服务对用户的评价矩阵，计算服务 j 评价偏差。第 8 行是计算服务主观评价的误差，它的计算方法如公式 4-5 所示，它的本质是通过判断服务对用户的主观评价可靠性是否大于用户对用户的评价，如果可靠，则加重服务对用户主观评价的权重。

$$ese_j = \sum_{i=1}^m w * DI(R_r, se_i) \quad (4-5)$$

算法 4-4: 服务评价权值更新

输入: $ES_{m \times n}$: 用户对服务的评价矩阵, $SES_{m \times n}$: 服务对用户的主观评价矩阵

输出: w : 服务主观权值

1. **while** ($v(c, k) \neq 1$)
 2. **for** r in $ES[j]$:
 3. $h_j(r, k) = \sum_{i=1}^m I(score_i(s_j) = r)$
 4. $f(r, k) = \frac{h_j(r, k)}{\sum_{r' \in [minScore, maxScore]} h_j(r', k)}$
 5. **end for**
 6. **end while**
 7. **for** i in $SES_{m \times n}$:
 8. $ese_j \leftarrow w * DI(R_r, se_{ij}, es_{ij})$
 9. $eo_{ej} \leftarrow eo_{ej} + (1 - w) * DI(R_r, se_{ij}, es_{ij})$
 10. $w \leftarrow 1 - \frac{(ese_j)}{ese_j + eo_{ej}}$
 11. **return** w
-

可以利用公式 4-6 判断服务对用户的主观评价可靠性是否大于用户对用户的用户评价。算法 4-4 的第 10 行是对服务主观评价的更新。

$$DI = \begin{cases} 1 & \text{if } |es_{ij} - R_r| > |se_{ij} - R_r| \\ 0 & \text{else} \end{cases} \quad (4-6)$$

4.2 基于 Chernoff bound 的人数阈值计算

4.2.1 用户人数阈值应用场景分析

服务生态系统中，经常会有新的服务产生，同时，为了满足用户需求，修改 bug，服务会对自身进行服务更新。如图 4-2 所示，当服务生态系统中服务产生或者更新时，用户使用了新产生的，或者更新的服务，会对服务做出评价。应当注意到，用户对服务的评价是一个时间序列，并非同时到达。用户数量的选取，不仅影响到服务的信誉的计算可靠性，同时，也影响计算的复杂度。直觉上，用户的评论数量越多，这些评论所反映的服务可信度就越能接近服务的真实可信度，评论越可信，用户越有可能参照评论决定是否使用服务，对于服务提供商来说，往往希望用户尽早使用服务，一方面，用户尽早使用服务，并对服务评价，以便吸引更多的用户来使用服务，另一方面，有些服务，是有时效限制的，一旦超过某个时间节点，这个服务的获益就大幅度减小（服务生态系统中很快会有类似的服务出现，新发布的服务处于垄断地位）。用户往往在等其他用户的评论达到一个稳定的水平，才决定是否使用服务，因此，如何选取合适用户人数的阈值，保证服务信誉能反映服务质量的同时，减少计算量，具有重要意义。



图 4-2 应用场景

4.2.2 用户人数阈值计算算法

服务进入生态系统或者更新的时候，服务生态系统需要一个指标告知用户当前用户的评价可信度。本文将用户人数作为阈值，计算达到指定的评价准确率所需的最少用户数。

首先，根据用户的历史评价，计算用户的评价准确率均值，计算公式如 4-6 所示，其中 \bar{r}_j 是服务 j 的评价均值， n 是用户评价的服务总数， m 是用户总数。本文通过计算当前用户对服务的评价与服务真实质量的偏差，计算用户的评价的准确度。

$$\mu = \frac{\sum_{i=1}^m \frac{\sum_{j=1}^n \frac{|score_{ij} - \bar{r}_j|}{1 \text{Max}(\bar{r}_j - \text{minscore}, \text{maxscore} - \bar{r}_j)}}{n}}{m} \quad (4-7)$$

本文利用投票策略，来计算用户对服务评价的可信度。投票策略指的是超过半数的人认为某一观点正确，那么最终大家就都认同这一观点。本文以用户历史评价准确率为权重，计算当前用户对服务评价的可信度。其具体做法是利用公式 4-7，其中， C 是期望的用户评价可信度， μ 是用户评价准确度均值。

$$n = \frac{-\ln(1-C)*2*(\mu)}{(\mu-1/2)^2} \quad (4-8)$$

用户人数的阈值计算如算法 4-5 所示。

算法 4-5: 最少人数计算

输入: U : 用户集合; C : 评价可信度; S : 服务集合; r : 用户评级

输出: num:最少用户数量

```

1: for each  $u_i \in U$  do
2:    $uc_i \leftarrow 0$ 
3:   for each  $s_j \in S$  do
4:      $\bar{r}_j \leftarrow \text{select average(rank) from } R \text{ where } s = s_j$ 
5:      $\text{minscore} \leftarrow \text{select min(rank) from } R \text{ where } s = s_j$ 
6:      $\text{maxscore} \leftarrow \text{select max(rank) from } R \text{ where } s = s_j$ 
7:      $uc_i \leftarrow uc_i + (r_{ij} - \bar{r}_j) / \text{Max}(\bar{r}_j - \text{minscore}, \text{maxscore} - \bar{r}_j)$ 
8:   end for
9:    $m \leftarrow \text{selectcount}(r) \text{ from } R \text{ where } u = u_i$ 
10:   $E(u_i) \leftarrow uc_i / m$ 
11:  $\mu \leftarrow \mu + E(u_i)$ 

```

```

12: end for
13:  $\mu \leftarrow \mu / |U|$ 
14:  $\text{num} \leftarrow \frac{-\ln(1-C)*2*(\mu)}{(\mu-1/2)^2}$  //将 $\mu$ ,  $C$  带入公式 4-8
15: return num

```

4.2.3 人数阈值计算合理性证明

由算法 4-5 可知，用户人数的阈值计算核心是公式 4-8。本节将针对公式 4-8 进行证明。用户人数阈值计算的公式是建立在两个假设的基础上：1. 用户的平均评价准确率大于 50%；2. 用户评价是独立的。关于用户人数阈值计算的证明如下所示。

服务产生或更新后，系统中至少有 $n/2$ 个用户给出的评价是准确的概率计算公式如下：其中 p_i 是用户 i 对服务给出正确评价的概率。

$$P_{\frac{n}{2}} = \sum_{\xi \subseteq U, |\xi| \geq \lceil n/2 \rceil} \prod_{u_i \in \xi} p_i \prod_{u_j \notin \xi} (1 - p_j) \quad (4-9)$$

之后，我们计算用户给出正确评价的期望值：

$$\begin{aligned}
 E(P_{\frac{n}{2}}) &= E\left(\sum_{\xi \subseteq U, |\xi| \geq \lceil n/2 \rceil} \prod_{u_i \in \xi} p_i \prod_{u_j \notin \xi} (1 - p_j)\right) \\
 &= E\left(\sum_{k=\lceil n/2 \rceil}^n \left(\sum_{\xi \subseteq U, |\xi|=k} \left(\prod_{u_i \in \xi} p_i \prod_{u_j \notin \xi} (1 - p_j)\right)\right)\right) \\
 &= \left(\sum_{k=\lceil n/2 \rceil}^n \left(\sum_{\xi \subseteq U, |\xi|=k} \left(\prod_{u_i \in \xi} E(p_i) \prod_{u_j \notin \xi} E(1 - p_j)\right)\right)\right) \quad (4-10)
 \end{aligned}$$

令 $E(p_i) = \mu$ ，那么上述的用户期望值为：

$$\begin{aligned}
 E(P_{\frac{n}{2}}) &= \left(\sum_{k=\lceil n/2 \rceil}^n \left(\sum_{\xi \subseteq U, |\xi|=k} \left(\prod_{u_i \in \xi} \mu \prod_{u_j \notin \xi} (1 - \mu)\right)\right)\right) \\
 &= \left(\sum_{k=\lceil n/2 \rceil}^n \left(\sum_{\xi \subseteq U, |\xi|=k} (\mu^k (1 - \mu)^{n-k})\right)\right) \\
 &= \sum_{k=\lceil n/2 \rceil}^n C_n^k \mu^k (1 - \mu)^{(n-k)} \quad (4-11)
 \end{aligned}$$

根据 Chernoff bound²得到：

$$\sum_{k=\lceil n/2 \rceil}^n C_n^k \mu^k (1 - \mu)^{(n-k)} \geq 1 - e^{\left(-\frac{1}{2\mu} n(\mu-1/2)^2\right)} \quad (4-12)$$

1. ² https://en.wikipedia.org/wiki/Chernoff_bound

假设要求评价的准确率为 C , 只要使得上式大于 C 即可。由此, 可以计算出, 所需的最少人数为: $\frac{-\ln(1-C)*2*(\mu)}{(\mu-1/2)^2}$ 。

当生态系统中都是可信用户时, 可以通过公式 4-7 对所需人数阈值计算。然而, 如前文所述, 生态系统中, 常常会存在恶意用户。恶意用户可能会通过伪装获得信任, 在某个时间突然发动攻击, 对服务的信誉造成损害, 因此, 我们需要考虑一些正常用户突然变成恶意用户的情况。系统中的用户类型为 $r = \{u_n, u_w\}$, 那么随机选择一个用户 i 对服务的评价准确率为:

$$P(r_i) = P(r_i | u_n) * P(u_n) + P(r_i | u_w) * P(u_w) \quad (4-12)$$

我们假设至少一半以上的正常用户给出的评价是正确的, 用户为正常用户时对服务给出正确的评价的可能性为 p_n , 用户为水军时给出答案的正确性为 p_w , 那么, 基于可信用户的历史数据可得 $E(p_n) = \xi$, 恶意用户的历史数据可得 $E(p_w) = v$, 假设系统中正常用户的比例为 α , 那么系统中非正常用户的比例 $1 - \alpha$, 注意 $E(u_i) = E(P(r_i)) = \xi * \alpha + v * (1 - \alpha)$, 最后得出所需最少的用户人数为:

$$n = \frac{-\ln(1-C)*2*(\xi*\alpha+v*(1-\alpha))}{(\xi*\alpha+v*(1-\alpha)-1/2)^2} \quad (4-13)$$

4.3 基于带权 EM 算法的服务信誉评估

4.3.1 众包与用户评价比较

将任务发布到众包平台, 利用群体智慧解决任务, 如表 4-1 所示, 其实与利用用户评价构建信誉系统与许多相似之处。用户评价是一种特殊的众包, 其本质也是利用群体智慧解决特定任务。具体到信誉系统, 任务是获得服务的真实评价, 服务提供商就是任务发布方。不同的是, 一般用户评价服务是没有报酬的, 但在 3.2 中, 提出的用户激励机制, 实际上减小了众包与信誉系统中用户评价的区别。

表 4-1 众包与用户评价对比

	众包	用户评价
行为	雇主发布任务, 众包平台的工人决定是否接受任务。	用户使用任务, 之后, 用户自行决定是否对服务评价。
用户类型	高质量工人, 低质量工人。	正常用户, 恶意用户
评价准确	工人类型决定	用户类型决定
报酬	有	无
任务类型	判断标准由任务类型决定, 一般不固定	判断标准不固定

4.3.2 服务信誉评估

利用 EM 算法对工人的质量进行评价是众包中最常见的方法。本文考虑利用 EM 算法估计服务的真实质量。EM 算法是一种迭代优化策略，由于它的计算方法中每一次迭代都分两步，期望步（E 步）与极大步（M 步），所以算法被称为 EM 算法。由于 EM 算法的初始值设置常常会影响到最终结果，因此，本文将用户信誉作为权值，提高初始值设定的可信度。

本文主要考虑以下场景：用户使用服务后，对服务做出评价，假设服务生态系统中共有 m 个用户， n 个服务， m 个用户对 n 个服务做出评价，未做评价的服务按照 svd 算法对评分矩阵进行填充，其中 $S_{mn} = \{s_{ij}\}$ 表示用户给服务的评分矩阵。 π_{ij}^k 表示用户 k 的混淆矩阵，即服务信誉为 i ，用户将其标注为 j 的可能性，以 $V_{n*1} = V_j (v_n \in [\minScore, \maxScore])$ 表示每个服务的真实评价，具体的计算方法如下：

- 1) 将用户信誉作为权值赋值给每个用户，并设定服务质量的初始值。
- 2) E 步，定义 ϕ_j^s 为第 j 个服务的信誉为 s 的概率， ϕ_j^s 的定义如下：

$$\phi_j^s = P(v_j = s | S_{mn}, \pi) \quad (4-10)$$

以用户信誉作为权值，计算后验概率 ϕ_j^s

$$\phi_j^s = \frac{p * \beta_j}{p * \beta_j + (1-p) * u_j} \quad (4-11)$$

其中， p_s 表示服务信誉为 s 的平均期望概率， $\beta_j = \prod_{i=1}^m (1 - \pi_i^1) s_{ij} \pi_i^{1-s_{ij}} * R_i$ ， $u_j = \prod_{i=1}^m (1 - \pi_i^0) s_{ij} \pi_i^{0-s_{ij}} * R_i$

- 3) M 步，通过用户评价结果与预估的服务质量的初始值，可以得到用户的误差矩阵。

$$\pi_{ij}^k = 1 - \frac{\sum_{j=1}^n \sigma_j * s_{ij}}{\sum_{j=1}^n \sigma_j} \quad (4-12)$$

通过以上几个步骤获得每个用户评价正确的后验概率 ϕ_j ，并计算后验概率 P 。重复迭代，直至函数收敛。最终可以获得用户的评价误差矩阵，以及服务信誉 V_{n*1} 。

4.4 本章小结

为了解决用户评价质量的预测，4.1 节利用用户-服务双向评价机制建立用户信誉，对用户识别与过滤，4.2 节介绍了用户人数阈值计算方法，结合用户的评价准确率，在新的服务产生或者服务更新的时候，估计获得可靠评价所需的最少

用户人数，并对其应用的场景以及计算方法，数学原理进行了详细的阐述，该模型可以在服务更新或者新的服务进入服务生态系统时，估计达到可靠的用户评价需要最少的用户人数。4.3 节中比较众包与用户评价的区别，分析将众包中的质量评估方法应用到服务信誉的可行性。之后，将常用于评估众包质量的 EM 算法应用到服务信誉的获取中，并将用户信誉作为权重，迭代计算得出服务信誉。

第5章 实验和分析

本章主要对前面提出的模型进行实验和分析。在 5.1 中，介绍了本文实验环境与数据来源，对实验模拟的用户数据与服务数据的特征进行了分析。5.2 中介绍了本文使用的评价指标。5.3 中确定了实验的最佳参数。5.4 中对用户奖励机制的效果，进行了分析，同时，通过用户-服务双向评价过滤恶意用户后，利用 EM 算法对服务信誉在服务质量变化时的效果进行分析。

5.1 实验整体流程

实验的整体流程如图 5-1 所示：

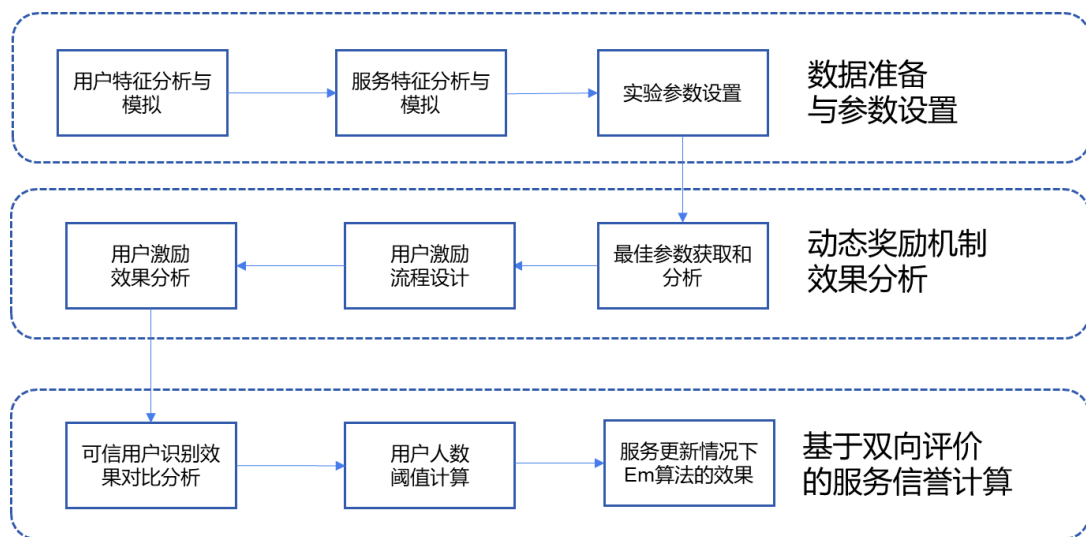


图 5-1 实验整体流程

- 1) 数据准备与参数设置。由于缺乏真实数据，本文对实验所需数据进行仿真模拟，分析用户和服务的特征，并按照这些特征对用户和服务的行为进行模拟。之后，给出实验中的参数设置。
- 2) 动态奖励机制效果分析。先分析当前实验中参数选取对实验的影响，之后，分析用户激励机制的效果，比较了在无激励机制下、固定激励下，动态激励下达到同样服务信誉所需的成本，以及服务信誉达到稳定需要的时间对比。

3) 基于双向评价的服务信誉计算效果分析。首先, 比较基于双向评价机制与 HMRep 方法的可信用户识别效率。之后, 计算获取可信用户所需的最少人数。最后, 考虑在服务质量改变的情况下, 基于 EM 算法计算服务的信誉是否能反映服务质量的变化。

5.2 实验数据准备

当前, 限于用户行为数据多是企业内部数据, 且用户评分数据有限。许多现有的方法^{[3][4][13]}运用模拟的数据对模型效果进行验证。类似的, 本文使用模拟数据对模型效果进行校验。实验过程中, 主要在刻画用户特征的基础上, 模拟用户的服务的评价, 以及刻画服务特征基础上, 模拟服务对用户的评价。本文所有实验均是使用 python 编写, 并在 windows10 系统下执行。实验环境为 Dell 台式机, 8G 内存, i7 处理器, 3.6GHZ。

5.2.1 用户特征模拟

本文在模拟用户时, 考虑用户的多样性。将用户分为三类: 正常用户, 不负责任的, 水军。实验中设计的正常用户的特征与评价函数如下表所示:

表 5-1 正常用户行为描述与模拟函数

行为描述	模拟行为的函数
评价: 虽然正常用户收到偏差影响, 对于服务的评价的描述可能会有一定偏差, 但是正常用户的评价时围绕服务的真实质量, 因此, 我们用正态分布函数来描述正常用户的评价随用户的变化	$s = N(\mu; \sigma^2)$ 其中, s 是用户对服务的评级, μ 代表服务的真实信誉, σ 代表用户评分与服务真实信誉之间的偏差,
使用时间: 正常用户的使用时间, 由程豪等人 ^[41] 的研究可知, 符合对数正态分布曲线	$t = \ln N(\mu; \sigma^2)$, 其中, μ 是正常用户的平均使用时间, σ 代表服务的热度的倒数, 热度越低, 服务的使用时间越少。 μ 的初始值设置为 2, σ 初始值在初始化服务时随机生成
注册时间, 为了简化计算, 我们将所有正常用户的注册时间都定为系统初始化时	$registerTime = 0$, $registerTime$ 代表用户注册使用服务的时间。
用户的好友数量: 以六度分割理论为基础, 任意一个正常用户都可以通过最多六次, 查找到其他正常	本文使用 gephi 来模拟初始的正常用户的分布。

用户

用户在服务上花费的金额:

$$P(\text{money}) = cm^{(-r)}$$

其中, m 用户的使用金额的比例, $P(\text{money})$ 是使用金额的概率。

不负责任用户的特征与评价函数如下表 5-2 所示, 不负责任用户与正常用户的主要区别在于对服务评价的随意性。

表 5-2 不负责任用户的行为描述与模拟函数

行为描述	模拟行为的函数
评价: 不负责任用户的评价是按照服务规定的评分范围随机给出, 与服务的真实质量无关	$s = \text{random}(\text{minScore}, \text{maxScore})$ minScore 是服务规定的用户可以给出的最低评级, maxScore 是服务规定的用户可以给出的最高评级, random 是随机函数, 使得不负责任用户的评价在最低到最高评级中任意选择

水军的行为特征和模拟函数如表 5-3 水军的行为描述与模拟函数表 5-3 所示:

表 5-3 水军的行为描述与模拟函数

行为描述	模拟行为的函数
评价: 由于受到服务提供商的雇佣, 水军的评价具有指向性, 有受雇佣提高自身评价的水军, 会给服务给出最高分, 同样, 也有受雇佣给竞争对手差评的水军, 会给服务最差的评价	$s = \begin{cases} \text{maxcore} & \text{提升信誉} \\ \text{minxore} & \text{降低信誉} \end{cases}$
使用时间: 水军用户的使用时间往往小于正常用户	$t = \text{InN}(\mu; \sigma^2)$ μ 的初始值设置为 1, 小于正常用户的使用时间, σ 初始值在初始化服务时随机生成 $\text{registerTime} = 0$
注册时间, 为了简化计算, 我们将所有水军用户的注册时间都定为系统初始化时	
用户的好友数量:	使用 gephi 来模拟初始的水军用户的分布。
水军的账号往往是临时注册的, 因此, 水军的好友数量比较低, 我们设置水军的平均好友数量为 2	
用户在服务上花费的金额:	$\text{money} = c$ money 是水军在服务上的花费, 为固定

值，服务所要求的最低费用。

5.2.2 服务特征模拟

服务对用户进行评价，如前文所述，分为主观评价与客观评价，主观是由服务根据用户的评论给出的，本文假设，服务对自身的质量有准确的认识。服务通过判断用户的评价与自身实际的质量的对比，给出评价。

表 5-4 服务的特征与描述函数

行为描述	模拟函数
服务评价：服务按照用户对自身评价与实际质量的偏离为标准	$\zeta S_k^i = \begin{cases} \max \text{ score} - ws_k - \tau_j(ws_k) , & T(ws_k) \geq \tau_j(ws_k) \\ \max \text{ score}, & T(ws_k) < \tau_j(ws_k) \end{cases}$
故障数量：本文使用泊松分布模拟服务出现故障的概率	$\text{errorCount} = \frac{e^{-\lambda} \lambda^k}{k!}, \text{其中 } k \text{ 是服务出现故障的次数, } \lambda \text{ 是服务平均故障次数, 与服务质量成反比。}$

5.2.3 实验参数设置

仿真实验中，我们设置用户人数为 1000，服务数量为 50。为了保证实验效果的可信，我们主要选取了以下几个场景：

- 可信用户数量较高，800 个可信用户,200 个恶意用户，（HCU）
- 可信用户数量中等，600 个可信用户，400 个恶意用户（MCU）
- 可信用户数量低，500 个可信用户，500 个恶意用户（LCU）

其中，服务的质量范围从 1-5 进行变化，其中，质量为 1 的服务是表现最差的服务。为了让仿真实验更好的反映现实环境，本文按照应用商店中服务的评价分布作为服务生态系统中不同质量的服务分布。如图 5-2 所示，是本文爬取的 ios 商城的 app 评分分布：

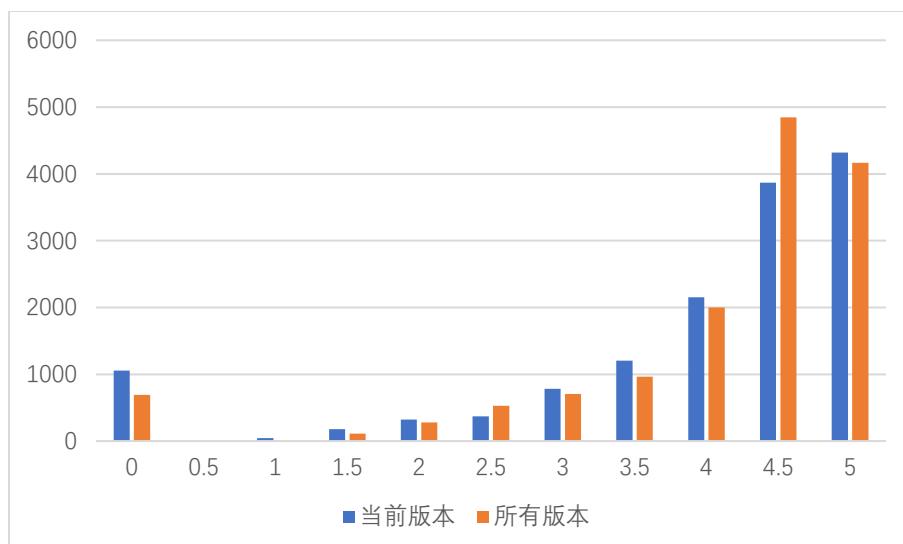


图 5-2 ios 商城中不同版本 app 评分分布

本文将 ios 商城中 app 的评分分布作为服务生态系统中服务的质量分布，设置服务的质量分布为质量为 5 的服务 30 个，质量为 4 的服务 6 个，质量为 2, 3 的服务 4 个，质量为 1 的服务 6 个。

5.3 基于用户评论质量的动态奖励机制效果分析

5.3.1 最佳参数的获取和分析

基于服务对用户的评价，可以得到用户对应的信誉值 $reputation$ ，当用户的信誉值 $reputation > K$ 时，则认为是可信用户。为了找到合理的 k 值，使得 F1-score 最高。我们选取用户信誉值从 4-5 变化，本文将利用 F1-score 作为评价指标。评价指标为 F1-Score，F1-score 是模型准确率和召回率的加权和，兼顾了模型的准确率和召回率，定义如下：

$$\begin{aligned}
 F1 - score &= 2 * Precision * Recall / (Precision + Recall) \\
 Precision &= |DCU \cap CU| / (|DCU \cap CU| + |DCU \cap (COU \cup IU)|) \\
 Recall &= |DCU \cap CU| / (|DCU \cap CU| + |(DIU \cup DCOU) \cap CU|)
 \end{aligned} \quad (5-1)$$

我们使用 CU, COU 以及 IU 分别代表可信用户，水军以及不负责任用户，让 DCU, DCOU 以及 DIU 分别代表利用基于用户-服务双向评价机制检测出的可信用户，水军以及不负责任用户。

在上述三个场景下，设定水军和不负责任的用户的比例为 50%（我们在三种场景下水军比例的变化，得到类似的结果）。如图 5-3，展示了 3 个不同场景下 F1-score 的变化情况，当 K 值设置在 4 到 4.4 之间，F1-score 最高，即模型的效

果最好；在 K 值大于 4.4 之后，F1-score 降低，这是由于可信用户自身受到环境和主观的偏好的影响，对于服务的评价存在一定偏差导致的。同时由于可信用户中存在一定数量的专家用户，他们对服务的评价准确，因此，积累了较高的信誉值，当 K 值超过 4.4 之后，专家用户基本可以被识别出来，因此，F1-score 最终维持在一定水平。因此，为了取得最好效果，本文的后续实验，都是在 K 值为 4.4 下进行的。

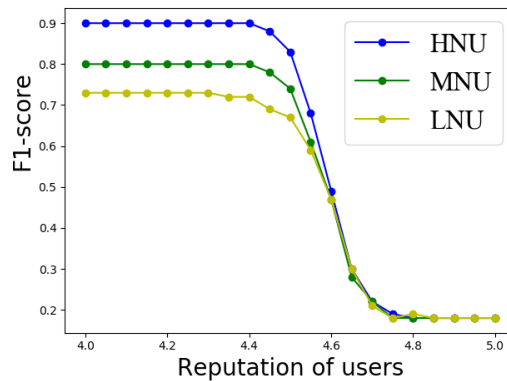


图 5-3 三种场景下 K 值选取对服务质量的影响

另一方面，我们可以看到在三种场景下，HNU 下的 F1-score 最高，即可信用户数量多的时候，F1-score 更高，这是因为尽管可信用户更容易获得高的信誉值，但由于按照信誉值来判断可信用户，很多信誉值低的可信用户被过滤了出去，因此，模型的召回率不高，导致最终的 F1-score 不高。

5.3.2 用户激励机制效果评估总体流程

为了将基于用户评论质量的动态奖励机制应用于信誉系统中，本文设计了一系列实验对动态奖励机制进行分析，主要的流程如下。

- 1) 服务根据当前自身的评级数量与用户的评论质量，判断当前用户的评级是否达到稳定，如果没有达到稳定，则对评论服务的用户进行奖励，设置奖励值。
- 2) 理性用户通过观察服务奖励值，判断当前奖励值是否值得进行标记
- 3) 服务判断当前自身所处的奖励阶段，计算评价服务的用户应得的奖励值，并将奖励值付给用户，如果服务的评级稳定性处于第一阶段，直接给予最高奖励，如果处于第二阶段，则需要计算服务对用户的评价，给予用户奖励
- 4) 更新服务的最高奖励，并且将更新后的奖励告知用户
- 5) 如果服务的评级仍未达到稳定，则转回第二步。

在动态用户奖励机制中，最关键的是确定激励分段点，如果分段点选择的过早，那么最终的服务评级稳定性过差。如果分段点选择的过晚，那么用户评级已

经趋于稳定，相对于第一阶段，时间太短，动态奖励机制退化为统一奖励机制，为了达到在保证用户评论质量的同时，减少支出，必须选择合理的分段点。利用用户人数阈值计算方法，我们可以获取达到指定准确率所需的最少用户人数。在实验中，本文设定希望达到的用户评价准确率为 90%，在计算使用服务的平均准确度后，通过 4.2 节的用户人数阈值计算方法，可以计算需要的用户人数。

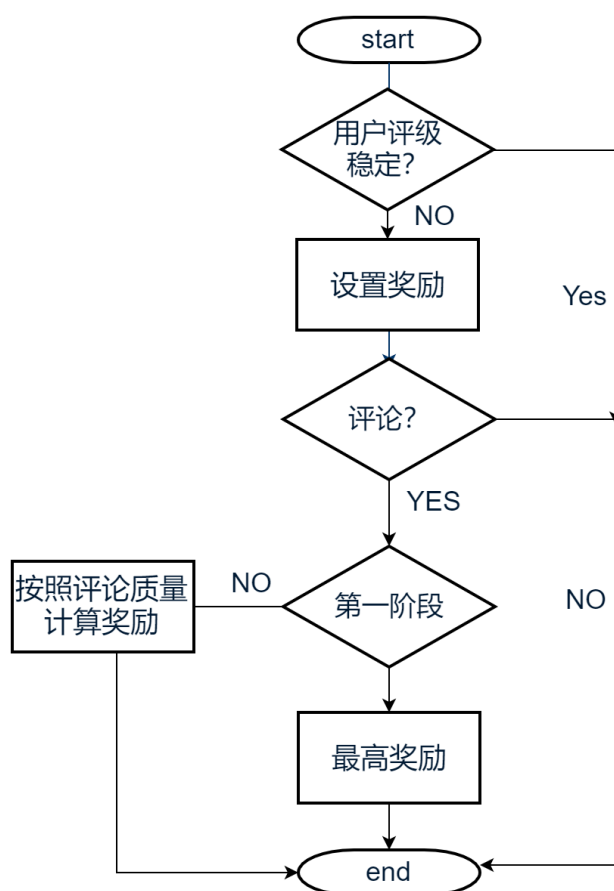
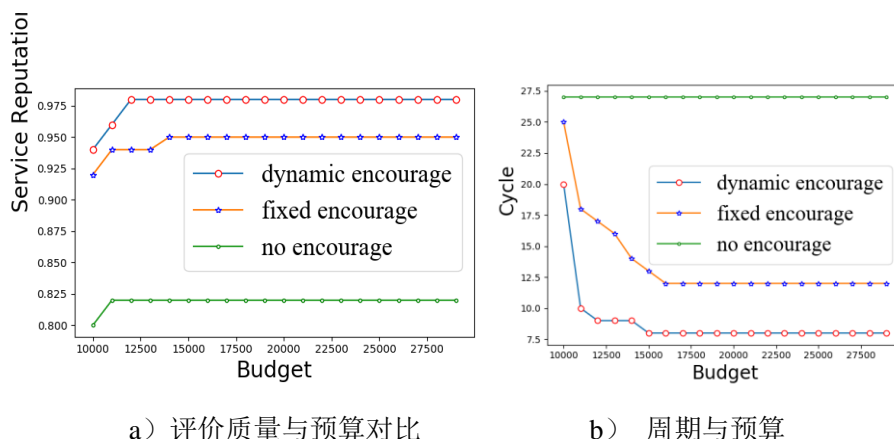


图 5-4 评估流程

5.3.3 用户激励机制效果分析

本文在考量用户激励机制的效果的时候，主要几个方面进行比较，一、在达到相同评价质量（评价稳定性）的条件下，比较在没有用户激励机制、固定用户激励、动态用户激励机制条件下，所需的用户数量；二是比较在相同预算条件的条件下，三种激励机制的标记质量。用户激励机制效果的模拟过程中，由 3.2 的分析可知，没有激励的情况下，用户不会对服务做出评价，一般而言，对服务做出评价的用户，主要是利他型用户，即不要报酬也会做出评价的用户。在实验中，按照爬取的谷歌和豌豆荚数据中用户下载量与评论量的比例，在无用户激励条件下，设置评论比例为 100:1，即 100 个正常用户中有 1 个用户会对服务进行评论。本文通过轮盘算法随机选择正常用户对服务评论。在固定奖励的情况下，按照 3.2

节的分析，用户倾向给出随机评价。本文设置评论质量比例为 1:1，即一半的用户会给出随机评价，一半的用户会给出高质量评价。在动态奖励机制条件下，用户会按照所处的时间阶段，给出不同质量的评价。



a) 评价质量与预算对比

b) 周期与预算

图 5-5 动态激励机制分析

在服务提供商相同的预算下，用户的评价质量。如图 5-5 a) 所示，无激励机制的条件下，用户的评价质量偏低。动态激励机制下，用户的评价质量要高于固定激励条件下用户的评价质量。

在服务提供商相同的预算下，用户达到稳定的评价质量需要的周期。如图 5-5 b) 所示，无激励机制的条件下，用户的评价质量达到稳定与用户的预算无关。而在动态奖励机制的条件下，用户评价达到稳定的周期要小于固定奖励下需要的周期。

5.4 基于用户信誉的服务信誉计算效果分析

5.4.1 可信用户识别准确率的对比和分析

为了测试用户-服务双向评价机制识别可信用户类型的效果，我们在上述三种场景下，不断调整系统内恶意用户中水军的比例。即当恶意用户的比例等于 50%，表明恶意用户中，一半是不负责任用户，一半是水军。展示了三种场景下，基于用户-服务双向评价机制和 HMRep^[40] 对恶意用户的识别效果。如图 5-6 所示：基于用户-服务双向评价机制在三种场景下都能保持较高的 F1-score 值，最高达到 95%。这主要是由于利用服务对用户的评价，相比 HMRep 算法，一方面：用户激励机制增加了用户评论的数量，在一定程度上解决了评论数据稀疏的问题；另一方面，基于双向评价对恶意用户识别，可用数据广。

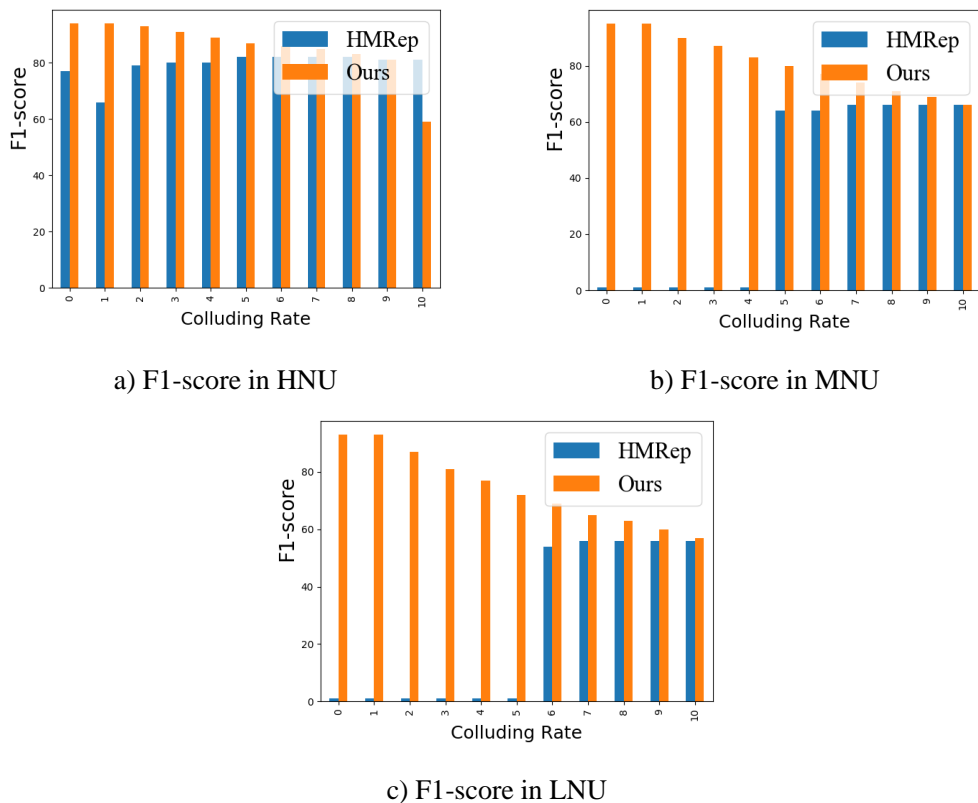


图 5-6 三种场景下本文方法与 HMRep 方法恶意用户识别效果的对比

5.4.2 用户人数计算

在服务对用户评价的基础上，利用用户信誉过滤恶意用户集合后，利用公式 4-8 计算需要的最少用户人数，本文随机选取三次实验中所需的最少用户人数。

表 5-5 最少用户人数计算

用户平均准确率	评价准确率	最少人数
62%	90%	70
64%	90%	31
66%	90%	30

5.4.3 服务更新情况下的自适应对比和分析

考虑本文提出的模型在服务的质量变化时，服务信誉是否能反映服务信誉的变化。我们在 HCU 场景中进行了测试（其他场景类似）。随机挑选一个服务，服务的质量在最低质量和最高质量之间摆动。具体来说，在前 5 轮，服务质量为 1，然后上升到 5，之后重复这一过程，我们考虑两个场景：

- 1) 从 5 到 1 摆动，服务质量周期性由高变低；
- 2) 从 1 到 5 摆动，服务质量周期性由低变高。

从图 5-7 可以看出，本文提出的方法对服务质量的改变非常敏感，尤其是当服务质量较差时，得到的服务信誉与服务质量的变化基本一致。

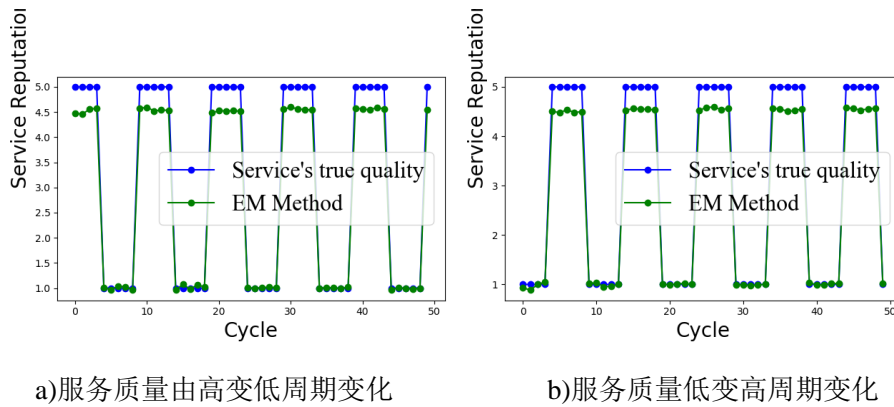


图 5-7 服务质量变化时服务信誉变化趋势对比

当服务的质量从高到底或者从低到高变动时，正常用户和专家用户会服务的评价会随之改变，但他们的信誉值不会受到一个服务质量的改变而变化，因此，即使服务的质量发生变化，基于 EM 算法的服务信誉计算方法能迅速的监测到服务的变化，并改变服务信誉。

第6章 总结和展望

本章 6.1 节总结了论文的研究内容与主要贡献，本章 6.2 节阐述了本文存在的不足，为未来的工作改进提供参考。

6.1 工作总结

服务信誉不仅可以帮助用户决策，增强用户对服务的信任，而且在提高用户忠诚度等方面也发挥着不可替代的作用。随着 XaaS 的迅速推广，服务的定义得到极大扩展，服务可获得更多数据。但是在利用信誉系统获取可信服务的过程中，一直存在几个问题：用户评价数据不足，恶意用户给出评价减低评价可信性的问题。针对以上几个问题，本文的主要贡献如下：

1) 本文提出对用户评价进行动态奖励的机制，对认真评论的用户给予适当奖励，通过实验结果可以看出，在相同的预算条件下，比没有奖励的机制得到的评价质量更高，需要的时间也更少。

2) 本文提出建立用户-服务双向评价机制，在传统的用户评级的基础上，利用服务对用户的主观与客观评估，建立用户信誉度，过滤恶意用户，提供评价反馈的公正性。本文提出的模型的有效性已经通过实验得到了充分验证。实验结果表明我们的模型可以有效识别恶意用户。

3) 为了在服务更新或者产生的时候，及时告知用户当前评价的准确率，将用户人数作为指标，基于 Chernoff bound，计算获得可靠的服务信誉需要的最少人数。

4) 在建立用户信誉评价体系之后，借鉴众包领域中，利用 EM 算法评价工人工作质量，本文将用户信誉作为权值，将用户的评价准确度作为隐变量，计算服务信誉。

6.2 前景展望

本文虽然在计算服务信誉过程中，对数据稀疏与恶意用户等问题上进行了深入的探讨与研究，并做了一些优化与改进，但本文仍然存在一些不足：

1) 本文建立的用户-服务双向评价模型, 是在服务提供商将自身设备的部署等操作交予云服务平台进行, 这样服务生态系统才能获取到运行在其中的服务的运营数据, 本文提出的框架适用于部署在云服务平台上的服务, 对于自建服务器的服务, 本文提出的效果可能不是很明显。

2) 本文提出的框架, 需要进一步在大数据环境下进行验证, 并考虑服务评价用户的多样性, 尤其是服务提供商之间串通的情况下, 当前的框架的表现情况。

3) 本文中考虑的用户评论, 主要以用户评级为主, 未来可以将用户评论的文本数据通过机器学习的方法, 转化成向量, 加进本文的框架中, 以帮助过滤恶意用户。

参考文献

- [1] 韩燕波, 王桂玲, 刘晨等. 互联网计算的原理与实践 [M]. 北京: 科学出版社, 2010.
- [2] Duan Y, Fu G, Zhou N, et al. Everything as a service (XaaS) on the cloud: origins, current and future trends[C]. 2015 IEEE 8th International Conference on Cloud Computing. Piscataway, NJ: IEEE, 2015: 621-628.
- [3] 张卫. 基于 XaaS 的制造服务链形成与应用研究[D]. 浙江大学, 2011.
- [4] Praveen Sawkar. 为何一切业务都转变为“万事皆服务”(XaaS)模式?[J]. 网络安全和信息化, 2018(09): 24-26.
- [5] Bhardwaj S, Jain L, Jain S. Cloud computing: A study of infrastructure as a service (IAAS)[J]. International Journal of engineering and information Technology, 2010, 2(1): 60-63.
- [6] Paraiso F, Haderer N, Merle P, et al. A federated multi-cloud PaaS infrastructure[C]. 2012 IEEE Fifth International Conference on Cloud Computing. Piscataway, NJ: IEEE, 2012: 392-399.
- [7] Philip J, Shah D. Implementing Signature Recognition System as SaaS on Microsoft Azure Cloud[M]. Data Management, Analytics and Innovation. Berlin: Springer, Singapore, 2019: 479-488.
- [8] Standing S, Standing C, Love P E D. A review of research on e-marketplaces 1997–2008[J]. Decision Support Systems, 2010, 49(1): 41-51.
- [9] Wang Y, Lin K J. Reputation-oriented trustworthy computing in e-commerce environments[J]. IEEE Internet Computing, 2008, 12(4): 55-59.
- [10] Assemi B, Schlagwein D. RETRACTED: Provider feedback information and customer choice decisions on crowdsourcing marketplaces: Evidence from two discrete choice experiments[J]. 2016.
- [11] Van Der Heide B, Johnson B K, Vang M H. The effects of product photographs and reputation systems on consumer behavior and product cost on eBay[J]. Computers in Human Behavior, 2013, 29(3): 570-576.
- [12] Josang A, Ismail R. The beta reputation system[C]. Proceedings of the 15th bled electronic commerce conference. 2002, 5: 2502-2511.
- [13] Whitby A, Jøsang A, Indulska J. Filtering out unfair ratings in bayesian reputation systems[C]. Proc. 7th Int. Workshop on Trust in Agent Societies. 2004, 6: 106-117.

- [14] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks[C]. Proceedings of the fourth ACM conference on Recommender systems. New York, NY: ACM, 2010: 135-142.
- [15] Kerschbaum F. A verifiable, centralized, coercion-free reputation system[C]. Proceedings of the 8th ACM workshop on Privacy in the electronic society. New York, NY: ACM, 2009: 61-70.
- [16] Park S T, Pennock D, Madani O, et al. Naïve filterbots for robust cold-start recommendations[C]. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY: ACM, 2006: 699-705.
- [17] Schein A I, Popescul A, Ungar L H, et al. Methods and metrics for cold-start recommendations[C]. Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY: ACM, 2002: 253-260.
- [18] Agarwal D, Chen B C. Regression-based latent factor models[C]. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY: ACM, 2009: 19-28.
- [19] Park S T, Chu W. Pairwise preference regression for cold-start recommendation[C]. Proceedings of the third ACM conference on Recommender systems. New York, NY: ACM, 2009: 21-28.
- [20] Zhou X, Matsubara S. Towards robust reputation system based on clustering approach[C]. 2015 IEEE International Conference on Services Computing. Piscataway, NJ: IEEE, 2015: 33-40.
- [21] Forsati R, Mahdavi M, Shamsfard M, et al. Matrix factorization with explicit trust and distrust side information for improved social recommendation[J]. New York, NY: ACM Transactions on Information Systems (TOIS), 2014, 32(4): 17.
- [22] Fu X, Zou P, Jiang Y, et al. Qos consistency as basis of reputation measurement of web service[C]. The First International Symposium on Data, Privacy, and E-Commerce (ISDPE 2007). Piscataway, NJ: IEEE, 2007: 391-396.
- [23] Mehdi M, Bouguila N, Bentahar J. Trust and reputation of web services through qos correlation lens[J]. IEEE Transactions on Services Computing, 2015, 9(6): 968-981.
- [24] He X, Liao L, Zhang H, et al. Neural collaborative filtering[C]. Proceedings of the 26th international conference on world wide web. International World Wide Web Conferences Steering Committee. New York, NY: ACM, 2017: 173-182.
- [25] Sang J, Yan M, Xu C. Understanding Dynamic Cross-OSN Associations for Cold-Start Recommendation[J]. IEEE Transactions on Multimedia, 2018, 20(12): 3439-3451.
- [26] Zhu Y, Lin J, He S, et al. Addressing the item cold-start problem by attribute-driven active learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2019.

- [27] Yang D, Xue G, Fang X, et al. Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones[J]. IEEE/ACM Transactions on Networking (TON), 2016, 24(3): 1732-1744.
- [28] Wang P, Du X. QoS-aware service selection using an incentive mechanism[J]. IEEE Transactions on Services Computing, 2016, 12(2): 262-275.
- [29] Singla A, Krause A. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms[C]. Proceedings of the 22nd international conference on World Wide Web. New York, NY: ACM, 2013: 1167-1178.
- [30] Zheng X, Zhang X, Yu Y, et al. ELM-based spammer detection in social networks[J]. The Journal of Supercomputing, 2016, 72(8): 2991-3005.
- [31] Meda C, Bisio F, Gastaldo P, et al. A machine learning approach for Twitter spammers detection[C]. 2014 International Carnahan Conference on Security Technology (ICCST). Piscataway, NJ: IEEE, 2014: 1-6.
- [32] Wu T, Liu S, Zhang J, et al. Twitter spam detection based on deep learning[C]. Proceedings of the australasian computer science week multiconference. New York, NY: ACM, 2017: 3.
- [33] 单明辉, 贡佳炜, 牛尔力, 等. RulerRep: 一种基于偏离度的过滤不实评价新方法[J]. 计算机学报, 2010, 33(7): 1226-1235.
- [34] Wang S, Zheng Z, Wu Z, et al. Reputation measurement and malicious feedback rating prevention in web service recommendation systems[J]. IEEE Transactions on Services Computing, 2014, 8(5): 755-767.
- [35] 王尚广, 孙其博, 杨放春. Web 服务选择中信誉度评估方法[D]. 中国科学院软件研究所| 中国计算机学会, 2012.
- [36] 陈翔, 徐佳, 吴敏, 等. 基于社会行为分析的群智感知数据收集研究[J]. 计算机应用研究, 2015, 32(12): 3534-3541.
- [37] Birant D, Kut A. ST-DBSCAN: An algorithm for clustering spatial-temporal data[J]. Data & Knowledge Engineering, 2007, 60(1): 208-221.
- [38] Zhang Y, Chen X, Zhou D, et al. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing[C]. Advances in neural information processing systems. Cambridge, MA:MIT Press, 2014: 1260-1268.
- [39] 刘志娟. 考虑工作者信誉的众包质量控制方法[D]. 大连理工大学, 2018.
- [40] Wang M, Wang G, Zhang Y, et al. A high-reliability multi-faceted reputation evaluation mechanism for online services[J]. IEEE Transactions on Services Computing, 2016.
- [41] 程豪, 吕晓玲, 钟琰, 范超, 赵昱. 基于智能手机大数据的 APP 使用时间分布规律研究[J]. 数学的实践与认识, 2018, 48(19): 158-164.

- [42] Zhou X, Lin D, Ishida T. Evaluating reputation of web services under rating scarcity[C]. 2016 IEEE International Conference on Services Computing (SCC). Piscataway, NJ: IEEE, 2016: 211-218.
- [43] Mishne G, Carmel D, Lempel R. Blocking Blog Spam with Language Model Disagreement[C]. Chiba, CHB: AIRWeb. 2005, 5: 1-6.
- [44] Dong R, Schaal M, O'Mahony M P, et al. Topic extraction from online reviews for classification and recommendation[C]. Twenty-Third International Joint Conference on Artificial Intelligence. 2013.
- [45] Jindal N, Liu B. Opinion spam and analysis[C]. Proceedings of the 2008 international conference on web search and data mining. New York, NY: ACM, 2008: 219-230.

发表论文和参加科研情况说明

申请的专利：

[1] 冯志勇，苗旭，陈世展，薛霄，一种基于多源数据的用户评价可靠性判定方法，中国，申请（专利）号：201911021986.X，2019-10-25

参与的科研项目：

[1] 国家科技部重点研发项目：健康养老跨界服务应用示范，项目编号：2017YFB1401200，2017.12-2020.11.

[2] 国家自然科学基金重点项目：价值驱动的软件服务持续迭代演化机理与关键技术，项目编号：61832014，2019.01-2023.12.

[3] 国家自然科学基金面上项目：社会移动云环境下的 APP 自主协同及个性化混搭机制研究，项目编号：61572350，2016.01-2019.12.

致 谢

研究生的学习即将结束，在撰写论文的过程中，我的建模能力，表达能力都得到了极大的提升。

在这里感谢我的导师冯志勇教授，研究生期间，冯教授不仅对我的研究方向，研究内容提出了许多宝贵意见与建议，而且注重理论与实践相结合。同时，感谢组内的陈世展老师，薛霄老师，他们在我的科研过程中，付出了很多心血，提出了很多建设性的意见，为我论文的选题，内容的确定提供了思路。在建模过程中，几位老师各抒己见，为我提供了不同的研究重点以供我参考，在我遇到问题的时候，帮我出谋划策，答疑解惑，再次对他们真诚的感谢。

同时，还要感谢我身边的同学们，感谢姜亚南学长，肖建茂师兄，周哲远，么红帅，郭思莹以及学弟学妹们的支持与帮助。研一的时候，姜学长不仅在学习上给我细心的帮助，帮我解决编程时遇到的问题，而且在生活中给我很大的帮助。研二的时候，肖建茂师兄经常和我讨论我的论文内容，在我撰写本文的过程中，也给予了极大的支持。在学习的过程中，难免遇到各种磕磕碰碰，在这个过程中，因为各位同学老师的帮助，开导，我得以克服一个有一个困难。唯一美中不足的是，自己的研究工作未能发表论文。这是研究生涯最大的遗憾。在天津大学求学的日子里，我切实的感受到大学里同学之间友好，互助的氛围，为自己是一名天大的学生感到自豪，同时，也为自己能够遇到这么多的优秀的天大学子而感到荣幸。在以后的日子里，希望能够和大家保持联系，天涯比邻。

我还要感谢我的父母和家人，研究生期间，他们在物质上，精神上都给予了我极大的支持，没有他们的帮助，我不可能有机会来到天大，成为一名天大的研究生，希望父母可以永远身体健康。

再次感谢所有帮助过我的人，希望他们万事顺利！