

基于随机块模型的多层次服务生态系统演化分析

刘明义 涂志莹 徐晓飞 王忠杰

(哈尔滨工业大学 计算学部, 哈尔滨市 150001)

摘要 随着云计算、边缘计算和移动计算等新技术的快速发展, 可用服务的数量和多样性呈现出爆炸式增长。服务之间快速增长的合作、竞争关系催生了“服务生态系统”这一新现象。复杂多变的市场环境使得服务的变化频率越来越高, 因此服务生态系统的演化研究是当前一个重要的研究问题, 发现其演化过程中蕴含的潜在演化模式以帮助服务提供者和市场监管者做出决策。为此, 服务生态系统演化分析成为了软件服务工程领域的研究热点之一, 许多的研究工作依托当前流行的开源服务生态系统 *ProgrammableWeb* 展开。然而, 现有服务生态系统演化分析视角比较单一, 分析内容局限于服务生态系统宏观的网络拓扑性质, 这种分析方式极大的限制了服务生态系统演化分析的应用场景。本文首先利用带权重的度修正随机块模型对服务生态系统进行建模, 基于随机块模型中的参数变化, 从服务生态整体、服务社区和服务个体三个层次对服务生态系统的演化进行了全面的分析, 提出了面向服务生态整体层面的演化点发现算法、面向服务社区层面的社区演化事件检测算法、面向服务个体的个体发展阶段划分算法。基于一个典型的 Web 服务生态系统 *ProgrammableWeb* 的真实数据集进行了实验, 实验结果验证了本文方法的有效性。

关键词 服务生态系统; 演化分析; 随机块模型; *ProgrammableWeb*; API

中图法分类号 TP311.5

Multi-level Service Ecosystem Evolution Analysis based on Stochastic Block Model

LIU Ming-Yi TU Zhi-Ying XU Xiao-Fei WANG Zhong-Jie

(Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China)

Abstract With the rapid development of new technologies such as cloud, edge and mobile computing, the number and diversity of available services has exploded. The rapidly growing collaboration between services have given rise to a new phenomenon called "service ecosystems". The complex and changing market environment make services change more and more frequently, so the study of service ecosystem evolution is an important research issue to discover the potential evolutionary patterns embedded in its evolution to help service providers and market regulators to make decisions. The analysis of service ecosystem evolution has become one of the hot research topics in the field of service computing, and a lot of research works are carried out based on the currently popular service ecosystem *ProgrammableWeb*. However, existing research on service ecosystem evolution analysis often considers only a single level, and the analysis is limited to the macroscopic network

本课题得到国家重点研发计划资助项目(No. 2018YFB1402901)、国家自然科学基金资助项目(No. 61772155, 61832004, 61832014)资助。刘明义, 男, 1995年生, 博士生, CCF学生会会员, 主要研究领域为服务计算、知识图谱。E-mail: liumy@hit.edu.cn。涂志莹, 男, 1983年生, 博士, 副教授, CCF会员, 主要研究领域为软件工程、服务计算、知识工程、企业业务建模。E-mail: tzy_hit@hit.edu.cn。徐晓飞, 男, 1962年生, 博士, 教授, CCF会员, 主要研究领域为服务计算与软件服务工程、大服务与务联网、企业智能计算、企业业务建模。E-mail: xiaofei@hit.edu.cn。王忠杰 (通讯作者), 男, 1978年生, 博士, 教授, CCF会员 (提供会员号), 主要研究领域为服务计算、软件工程。E-mail: rainy@hit.edu.cn。

topology of service ecosystem, which greatly limits the application scenarios of service ecosystem evolution analysis. In this paper, we first model the service ecosystem using a weighted degree-corrected stochastic block model, and based on the parameter changes in the stochastic block model, we conduct a comprehensive analysis of the evolution of the service ecosystem at three levels: service ecosystem as a whole, service community and service individual, and propose an evolution point discovery algorithm for the service ecosystem as a whole, a community evolution event detection algorithm for the service community level, and an individual development stage classification algorithm for the service individual. The algorithm of individual development stage division is proposed. Experiments are conducted based on a real dataset of a typical Web service ecosystem *ProgrammableWeb*, and the experimental results verify the effectiveness of the method in this paper.

Key words *Service Ecosystem; Evolution Analysis; Stochastic Block Model; ProgrammableWeb; API

1 引言

随着云计算、边缘计算和移动计算等新技术的快速发展以及面向服务架构 (Service-Oriented Architecture, SOA) 的广泛应用, 越来越多的服务以互联网作为其发布和运行平台, 以 API 的形式向外暴露其功能, 客户端或其他服务通过标准化的协议对其进行访问。多个 API 可以容易的组合成更大粒度的应用程序, 称为服务聚合 (mashup) [1], 这些 mashup 可以为用户创造和提供各服务所无法提供的独特新价值。API 之间这种不断增长的“合作-竞争关系”催生了“服务生态系统”, 也称为 API 经济[2], 对传统互联网生态的产生了深远的影响。它逐渐弱化了传统业务领域的概念, 不断破除着领域间的壁垒, 让跨界合作和跨界整合变得更加普遍。

API 经济使服务提供者在跨界整合和创新上变得更容易, 从而创造了越来越多的新应用。此外, 随着用户逐渐使用这些新应用, 他们可能会提出新的需求, 这不仅进一步加快了创新进程, 同时也加剧了市场竞争。服务提供者需要对用户需求和偏好的变化保持敏感, 不断推出新的服务或更新已有服务。当服务发生变化时, 相应的服务生态系统也会随之发生变化, 有必要对服务生态系统的演化开展研究, 从不同的角度发现其变化的规律, 从而预测未来演化路径。从商业角度来看, 演化分析可以帮助服务提供商和市场监管者了解服务生态系统的演化模式, 协助其做出科学的决策以推动服务个体与服务生态系统的可持续和健康发展。例如, 通过演化分析, 服务提供商可以了解竞争对手的合作策略, 发现流行的市场演化趋势, 使他们能够针对稍纵即逝的创新机会调整自身的业务策略, 从而保持或增强服务的竞争力; 对于市场监管者来说, 理解服务生态系统的演化规律可以更好的履行其监管引导的职责, 从而帮助服务生态健康稳定的发展。

另一方面, 从服务计算的学术研究角度, 服务生态系统演化分析可以从数据中挖掘可解释的先验知识, 为其他下游任务提供支持, 如服务推荐、服务发现和服务组合, 从而加快服务开发的速度[3][4]。

ProgrammableWeb 是世界上最大的在线 API 商店平台, 它收集了大量的第三方 API 和 mashups, 每天都有新的 API/mashup 在上面发布、已有的 API 从中消亡、多个 API 合作形成新的 mashup。因此, 它可被看作一个典型的 Web 服务生态系统。此外, *ProgrammableWeb* 还是服务计算领域的一个标准数据集, 作为真实互联网生态系统的典型代表, 它被用于服务计算领域的诸多经典研究问题, 特别是服务推荐[5][6][7]、服务发现[8][9]、服务演化分析[10]和服务质量 (Quality of Service, QoS) 预测[11][12]等。截至 2021 年 1 月 9 日, *ProgrammableWeb* 收集了 23,881 个 API 和 7,973 个 mashup, 每一个 API 和 mashup 都包含了如创建日期、类别、简介和活动状态等信息。

目前, 已有一些研究从不同的角度开展 *ProgrammableWeb* 服务生态系统的演化分析。例如, Weiss 等人[13]通过统计 *ProgrammableWeb* 服务生态系统中可用服务数量随时间的变化情况来判断服务生态系统的健康状况。Adeleye 等人[14]基于 API 服务之间的关系将 *ProgrammableWeb* 服务生态系统组织成一个复杂网络图, 通过观察这个复杂网络的度分布的变化来断言 *ProgrammableWeb* 服务生态系统遵循幂律分布并基于幂律进行服务发现。Huang 等人[15]通过统计服务生态系统中例如密度、聚类系数等网络性质来为服务推荐提供先验知识。Fokaefs 等人[16]则关注服务生态中服务个体的演化, 通过比较 API 服务的接口描述的变化来探索 API 服务个体的演化。然而, 大部分已有研究停留在对 *ProgrammableWeb* 服务生态系统的简单网络拓扑性质的浅层次统计分析上, 且演化分析往往只关注服务生态整体、服务社区或服务个体中的一个层次, 尚缺少一套完善的方法能够同时从多个层次

对服务生态系统进行更全面的演化分析，以扩展服务生态演化的应用场景，支持商业和政务辅助决策等任务。

ProgrammableWeb 服务生态系统是一个持续演化的复杂网络系统，该网络系统中不同的服务由于功能的相似而聚成了服务社区，而系统的演化行为则可以由服务社区内部和服务社区之间的行为变化来描述。随机块模型[17]恰好可以对这样的复杂网络系统进行建模。从而，本文提出了一种基于随机块模型的多层次服务生态系统演化分析方法。该方法首先基于 API 及 API 之间的共现关系将 *ProgrammableWeb* 服务生态系统建模为动态网络，然后利用本文提出的一种带权重的度修正随机块模型对 *ProgrammableWeb* 服务生态系统进行数学形式化描述，将所需探究的服务生态整体、服务社区以及服务个体的行为参数化。随后，基于服务生态系统的参数化表达，对服务生态整体、服务社区和服务个体三个层次的演化问题分别给出了求解方法，包括：在服务生态全局层面，提出了一种全局演化点检测算法；在服务社区层面，提出了一种服务社区演化事件判定算法；在服务个体层面，提出了一种服务个体演化阶段划分算法。最后，在真实的 *ProgrammableWeb* 数据集上进行了实验，证实了本文方法的可行性。

本文的主要创新点如下：

1) 提出了一种利用带权重的度修正随机块模型对 *ProgrammableWeb* 服务生态系统进行建模的方法，有效的将服务生态系统中多层次信息统一到一个模型中。

2) 从服务生态整体、服务社区以及服务个体三个层次对服务生态系统进行了演化分析，对传统研究仅关注服务生态系统整体演化或仅关注服务个体演化的不足做了有效补充。

3) 提出了服务生态系统的整体演化点发现方法、服务社区的演化事件判别方法、服务个体演化阶段划分方法，对传统方法仅基于基础网络拓扑特征开展演化分析进行了深化。

本文章节安排如下：第 2 节总结了现有服务生态系统研究；第 3 节给出了 *ProgrammableWeb* 服务生态系统的网络表示及基于随机块模型的服务生态系统生成表示，并给出了相关参数的求解过程；第 4 节介绍了利用随机块模型中的参数从服务生态整体、服务社区、服务个体三个层次进行演化分析的方法；第 5 节给出了在真实数据集上的实验结果；第 6 节总结全文。

2 相关工作

近年来，像 *ProgrammableWeb* 这样的服务生态系统的演化分析已被广泛研究。现有的研究集中在单个服务状态的变化和服务网络拓扑结构的变化

上，这些研究旨在帮助开发者选择合适的服务整合到他们自己的应用程序中。这些研究的最终目标是提供先验知识，帮助解决各种传统的服务计算问题，如服务推荐和服务标签预测[18]。

例如，Wang 等人 [19] 通过研究 *ProgrammableWeb* 服务生态系统的网络和聚类属性，发现了 mashup 社区的用户行为模式，并且他们认为 mashup 社区具备无尺度的特征，频繁使用的 API 会吸引大量用户。Weiss 等人[13]研究了 mashup 生态系统的结构及其随时间的增长，并得出结论：i) mashups 在 API 上的分布遵循幂律和 ii) mashup 的复杂性继续增加。Huang 等人[15]提出了一个三阶段的网络预测方法（NPA），以研究整个 *ProgrammableWeb* 服务生态系统的使用模式和演化痕迹，用于演化感知的服务推荐。Yang 等人[20]在他们的工作中考虑了服务生态系统的演化，通过利用隐含狄利克雷分布（latent Dirichlet allocation, LDA）和时间序列预测来提取服务演化模式，然后用这些模式来指导服务推荐。Bai 等人[21]开发了一个定制的主题模型来挖掘服务生态系统的有效表征，解决了服务演化和信息稀少的问题，而 Gan 等人[22]提出了一种使用深度神经网络进行服务多标签推荐的新颖方法。Lyu 等人[23]提出了一个用于 Web 服务生态系统可视化的三层视图模型，该模型不仅有助于理解生态系统，而且还可以为用户提供支持，帮助他们发现合适的服务。

除了探索以 *ProgrammableWeb* 为代表的网络服务生态系统外，在最近的研究中还探讨了其他类型的服务生态系统的演变。例如，Sampaio 等人[24]的工作侧重于微服务生态系统，将不断发展的微服务系统的结构、部署和运行时信息汇总到一个模型中，提供了可操作的见解，以帮助开发者权衡服务升级、架构演变和不断变化的部署。Wang 等人[25]也关注了微服务生态系统，提出了一个基于知识的分布式演化模型（distributed knowledge-based evolution model, DKEM），该模型可以发现稳定的进化模式，并自动探索服务之间新的、更稳定的合作。Xue 等人[26][27]利用计算实验对工业服务生态系统进行了仿真。除了这些技术层面的服务生态系统，一些研究人员将兴趣转向了商业层面的服务生态系统的演化。最近的一项工作是 Liu 等人[28][29]提出的基于多层网络的服务生态系统模型（Multilayer Network-based Service Ecosystem Model, MSEM），它可以通过挖掘互联网上的大量文本数据集自动构建。通过引入服务事件的概念，MSEM 不仅可以探索演变模式，还可以确定演变的驱动因素。

现有工作从不同的角度对服务生态的演化进行了探究，但是大部分工作对于服务生态系统的演化分析依旧集中在简单网络拓扑性质的浅层次统

计分析上。现有工作进行演化分析时只关注服务生态系统中的单一侧面将服务生态演化分析的作用局限在了服务推荐、服务发现等传统服务计算任务。目前尚缺少一套完善的方法能够同时从多个层次对服务生态系统进行更全面的演化分析,以扩展服务生态演化的应用场景,支持商业和政务辅助决策等任务。

3 服务生态系统表示

表 1 常用符号说明

符号	含义
C', S'	时刻 t 服务生态系统中的 Mashup 和 API 集合
G^t	服务生态系统在时刻 t 快照
u, v	服务生态系统中的 API
r, s	服务社区
d_u	API 的被观测到的实际度
b_u	API u 所属的服务社区
b	服务生态系统快照的服务社区划分结果
B	服务社区的总数
\mathbf{A}	服务生态系统快照的邻接矩阵
\mathbf{x}	服务生态系统快照的权重矩阵
\mathbf{b}	服务生态系统快照的服务社区划分结果
\mathcal{O}	服务社区连接概率矩阵
\mathbf{k}	API 的度期望分布
$\mathbf{\gamma}$	服务社区间链接强度的矩阵

本节给出了服务生态系统的网络表示及基于随机块模型的服务生态系统生成表示,并给出了相关参数的求解过程。因本节设计到大量数学符号,表 1 中总结了常见符号及其含义以方便读者查阅。

3.1 服务生态系统的网络表示方法

ProgrammableWeb 服务生态系统可以被建模成以下两种动态图: 1) Mashup-API 图, 2) API-API 图。

定义 1. Mashup-API 图. Mashup-API 图是一个动态的二部图 $G_{CS} = \{C, S, E_{CS}\}$, 其中 C 表示服务生态系统中所有 Mashup 的集合, S 表示服务生态系统中所有 API 的集合, $E_{CS} = \{u \psi \text{ start } \text{end} \mid u \in C, v \in S\}$ 表示 Mashup 和 API 之间的调用关系, start 和 end 用来表示这个关系的持续时期。对于任意一个 API/Mashup 节点 $u \in C \cup S$ 可以被表示成 $u = \{\text{start}, \text{end}, b_u\}$, 其中 b_u 表示 u 所属的主要社区。 $G_{CS}^t = \{C^t, S^t, E_{CS}^t\}$ 用来表示 Mashup-API 图在时刻 t 对应的快照。

定义 2. API-API 图. API-API 图可以被表示为一个连续的网络快照集合 $G_{SS} = \{G_{SS}^1, G_{SS}^2, \dots, G_{SS}^T\}$ 。 $G_{SS}^t = \{S^t, E_{SS}^t\}$ 是 API-API 图在时刻 t 由对应的 Mashup-API 图 G_{CS}^t 生成的。 $E_{SS}^t = \{(u, v, w) \mid u, v \in S^t\}$ 表示 API 之间的共现关系, (u, v, w) 表示 API u 和

API v 在时刻 t 被可用的 w 个 Mashup 共同调用。

图 1 展示了 Mashup-API 图到 API-API 图的转换过程。图中的方形节点表示 Mashup, 圆形节点则代表着 API, 节点上的颜色意味着该 Mashup/API 所属的主要社区。图中的边表示调用关系或共现关系。例如, a_1 和 a_4 仅被 m_1 一个 mashup 共同调用, 所以在 API-API 图中 a_1 和 a_4 之间的权重为 1, 而 a_3 和 a_4 被 m_2, m_3 两个 mashup 共同调用, 所以 a_3 和 a_4 之间的权重为 2。

Mashup-API 图是 *ProgrammableWeb* 服务生态系统的天然表示形态, 然而由于该图是一个二部图使得 *ProgrammableWeb* 服务生态系统中大量的社交信息难以被有效的表达出来。因此, 通常在研究 *ProgrammableWeb* 服务生态系统演化时会将其转换成 API-API 图。本文也遵循这一惯例, 后续讨论中如无特别说明, 服务生态系统网络将指 API-API 图。

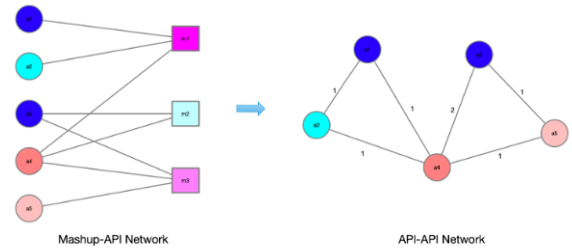


图 1 Mashup-API 图转换为 API-API 图的示意图

此外, 为了表示的简洁性, 后续讨论中将省略 SS 下标, 在和时刻信息无关且不引起歧义的情况下将省略上标 t 。进一步的, 为了便于下文中有随机块模型的表示, 本文使用 \mathbf{b} 表示服务生态系统快照 G 的社区划分结果, $b_u \in \{1, 2, \dots, B\}$ 用来表示 API 节点 u 所属的社区, 其中 B 表示的服务社区的总数目。为了进一步区分边的存在与边的权重, 使用符号 \mathbf{A} 和 \mathbf{x} 来分别表示服务生态系统快照 G 的邻接矩阵与权重矩阵。由于 G 是无向图, 所以 \mathbf{A} 和 \mathbf{x} 都是对称矩阵。

3.2 基于随机块模型的服务生态系统生成表示

在描述如何将随机块模型应用到服务生态系统网络之前, 本文先申明两个在 *ProgrammableWeb* 服务生态系统中的假设。随后, 介绍简单随机块模型 SBM (Stochastic Block Model) 及其在生成服务生态系统网络时的不足[28]。然后, 提出一种带权重的度修正随机块模型 WDCSBM (Weighted Degree Corrected Stochastic Block Model), 最后给出了 WDCSBM 的参数求解方法。

假设 1. 在任意时刻的服务生态系统快照 G 及对应的服务社区划分结果 \mathbf{b} 是可以直接观测到的;

假设 2. 在整个时间线上, 服务生态系统内包含的服务社区的总数量 B 是固定的。对于任意 API

节点 u , 其所属社区 b_u 是唯一的且不会随时间发生变化。

这两个在 *ProgrammableWeb* 服务生态系统中容易验证但却十分重要的假设。*ProgrammableWeb* 中的 API 服务只有一个主要标签, 该标签在 API 页面中可以被爬取到, 所以假设 1 在 *ProgrammableWeb* 上面是成立的; 并且这一标签在 API 提交时是固定的且后续是不允许更改的, 所以假设 2 在 *ProgrammableWeb* 上面是成立的。这两个假设的存在使得不用去关心服务社区划分、动态服务社区匹配等问题, 从而避免的对于本文研究内容不必要的误差积累。需要指出的是, 对于没有严格满足上述两个假设的服务生态系统, 本文提出的方法也是适用的。只需通过相应的服务社区划分算法获得每一生态系统快照中的服务社区划分结果, 并利用动态服务社区匹配算法获得前后快照中服务社区之间的对应关系即可。

在服务社区划分结果 \mathbf{b} 已知的条件下, SBM 假设在生成网络时, 两节点 u, v 之间存在链接 $A_{u,v}$ 的概率服从以社区连接概率 ω_{b_u, b_v} 为参数的伯努利分布, 即:

$$A_{u,v} | \mathbf{b} \sim \text{Bernoulli}(\omega_{b_u, b_v}) \quad (1)$$

其中 $\omega_{r,s} \in \boldsymbol{\omega}$ 表示社区 r 中的节点和社区 s 中的节点连接概率, 也称为社区连接概率, $\boldsymbol{\omega} \in \mathbf{R}^{B \times B}$ 则表示社区连接概率矩阵。

SBM 在对服务生态系统网络进行表示的时候存在的最大问题是它认为网络中所有成分都是等价的, 即无法区分任意取出两个节点或边。实际上性质差异在 *ProgrammableWeb* 服务生态系统中随处可见。例如, 首先对于 API 节点来说, SBM 简单的假设节点之间的链接存在仅和节点的所在社区相关且同一社区内所有节点都是等价的。然而, 现实情况并不是如此, *ProgrammableWeb* 中存在一些度明显高于其它结点的巨人节点, 例如 *google-maps*, *facebook*, *twitter*, *youtube*。这些巨人节点统治着服务生态系统中超过 80% 的链接, 所以他们在服务生态系统中具有更为重要的地位, 和其他节点产生链接概率时理应高于一般节点。其次, SBM 只考虑网络中的边是否存在并且认为所有存在的边都是等价的。然而在 *ProgrammableWeb* 服务生态系统中边表示的时 API 之间的共现关系, 这种共现关系有频次的不同, 这使得不同的边在服务生态系统网络中具有不同地位。所以, 简单随机块模型不能全面的刻画 *ProgrammableWeb* 服务生态系统网络。

为了应对 SBM 的上述不足, 本文提出了一种带权重的度纠正随机块模型 WDCSBM 来对 *ProgrammableWeb* 服务生态系统网络进行表示。WDCSBM 具有以下约束:

- 1) 节点之间的边存在与否不仅取决于节点所在的社区, 同时还会被节点自身的度属性所约束;
- 2) 节点之间的边具有权重这一属性;
- 3) 节点之间的是否存在边以及边的权重均是根据边上的节点采样获得的。

约束 1 控制了整个服务生态系统网络的拓扑结构, 使得随机块模型中可以出现不等价的节点。约束 2 则控制了服务生态系统网络中的边权重分布, 使得不等价边得以表达。约束 3 的存在, 使得约束 1 和约束 2 相独立, 即邻接矩阵 \mathbf{A} 和权重矩阵 \mathbf{x} 相独立。那么, WDCSBM 对于服务生态系统网络快照 G 的生成概率可以被写成下式:

$$P(\mathbf{A}, \mathbf{x} | \boldsymbol{\theta}, \boldsymbol{\gamma}, \mathbf{b}) = \underbrace{P(\mathbf{A} | \boldsymbol{\theta} = \{\boldsymbol{\omega}, \mathbf{k}\}, \mathbf{b})}_{\text{节点差异(度纠正)}} \underbrace{P(\mathbf{x} | \mathbf{A}, \boldsymbol{\gamma}, \mathbf{b})}_{\text{边差异(带权重)}} \quad (2)$$

其中 $\boldsymbol{\theta}$ 是用来支撑约束 1 的参数, 特别的, $\boldsymbol{\theta}$ 由 $\boldsymbol{\omega}, \mathbf{k}$ 组成, 其中 $k_u \in \mathbf{k}$ 用来表示节点 u 度的期望值。 $\boldsymbol{\gamma}$ 是表示社区间链接强度的矩阵被用来支撑约束 2 的参数。

为了能让 WDCSBM 准确的表示服务生态系统网络的生成规律, 需要寻找一组参数使得式(2)的对数似然函数取得最大值:

$$\ln P(\mathbf{A}, \mathbf{x} | \boldsymbol{\theta}, \boldsymbol{\gamma}, \mathbf{b}) = \ln P(\mathbf{A} | \boldsymbol{\theta} = \{\boldsymbol{\omega}, \mathbf{k}\}, \mathbf{b}) + \ln P(\mathbf{x} | \mathbf{A}, \boldsymbol{\gamma}, \mathbf{b}) \quad (3)$$

由于节点异质部分和边异质部分相互独立, 只需分别使两部分的对数似然函数取得最大值即可[29]。因此在下文中将依次介绍节点异质和边异质表达式的细节及相关超参数的后验求解。

在服务社区划分结果 \mathbf{b} 已知的条件下, WDCSBM 依旧认为两节点 u, v 之间产生边 $A_{u,v}$ 的概率服从伯努利分布, 但不同于简单随机块模型, WDCSBM 中伯努利分布参数不再仅是 $\boldsymbol{\omega}$ 而是由 $\boldsymbol{\omega}$ 吸收 \mathbf{k} 后形成, 即:

$$A_{u,v} | \mathbf{b} \sim \text{Bernoulli}(k_u k_v \omega_{b_u, b_v}) \quad (4)$$

且 $\boldsymbol{\omega}$ 和 \mathbf{k} 的归一化可以通过添加式(5)实现。

$$\sum_u k_u \omega_{b_u, r} = 1 \quad (5)$$

那么节点异质部分的生成概率则可以被表示为:

$$P(\mathbf{A} | \boldsymbol{\omega}, \mathbf{k}, \mathbf{b}) = \prod_{u < v} (k_u k_v \omega_{b_u, b_v})^{A_{u,v}} e^{-k_u k_v \omega_{b_u, b_v}} \\ = \prod_u k_u^{d_u} \prod_{r,s} \omega_{r,s}^{m_{r,s}} / 2 e^{-\frac{1}{2} D_r D_s \omega_{r,s}} \quad (6)$$

其中, d_u 表示节点 u 在快照 G 中观测到度。 D_r 则表示社区 r 里面所有节点的度的总和:

$$D_r = \sum_{u: b_u=r} d_u \quad (7)$$

n_r 表示社区 r 中包含的节点数量, $m_{r,s}$ 表示社区 r 和社区 s 之间边的数量:

$$m_{r,s} = \sum_{u,v} A_{u,v} \delta_{b_u,r} \delta_{b_v,s} \quad (8)$$

δ 是指示函数。

根据上述分析, 节点差异部分的似然函数可以进行如下表示:

$$\ln P(\mathbf{A} | \boldsymbol{\omega}, \mathbf{k}, \mathbf{b}) = \sum_u d_u \ln k_u + \frac{1}{2} (\sum_{r,s} m_{r,s} \ln \omega_{r,s} - \omega_{r,s} D_r D_s) \quad (9)$$

通过最大似然估算[17]可得后验参数估计:

$$\begin{aligned} \hat{k}_u &= d_u \\ \hat{\omega}_{r,s} &= \frac{m_{r,s}}{D_r D_s} \end{aligned} \quad (10)$$

对于边异质部分, *ProgrammableWeb* 服务生态系统中的边表示的是 API 之间的共现频率, 也就是说边的权值是离散正值, 因此可以认为边的权值分布符合泊松分布, 即:

$$P(x | \gamma) = \frac{\gamma^x e^{-\gamma}}{x!} \quad (11)$$

那么边异质部分的生成概率则可以被表示为:

$$\begin{aligned} P(\mathbf{x} | \mathbf{A}, \boldsymbol{\gamma}, \mathbf{b}) &= \prod_{u < v} P(x_{u,v} | \gamma_{u,v})^{A_{u,v}} \\ &= (\prod_{u < v} x_{u,v}!^{A_{u,v}})^{-1} \prod_{r,s} \gamma_{r,s}^{\mu_{r,s}} e^{-\gamma_{r,s} m_{r,s}} \end{aligned} \quad (12)$$

其中:

$$\mu_{r,s} = \sum_{u,v} \frac{A_{u,v} x_{u,v} \delta_{b_u,r} \delta_{b_v,s}}{1 + \delta_{r,s}} \quad (13)$$

根据上述分析边差异部分的似然函数可以进行如下表示:

$$\begin{aligned} \ln P(\mathbf{x} | \mathbf{A}, \boldsymbol{\gamma}, \mathbf{b}) &= -\sum_{u < v} A_{u,v} \ln(x_{u,v}!) + \sum_{r,s} \mu_{r,s} \ln \gamma_{r,s} - \gamma_{r,s} m_{r,s} \end{aligned} \quad (14)$$

利用最大似然估算[17]可以求得:

$$\hat{\gamma}_{r,s} = \frac{\mu_{r,s}}{m_{r,s}} \quad (15)$$

4 多层次服务生态系统演化探究

不同层次的分析会带来不同层面的见解。整体分析可以为决策者带来一些宏观的结论, 但是缺乏微观信息来对宏观结论进行解释; 而细粒度的局部和个体分析能够获取微观的信息进一步佐证整体分析获得的结论。因此本节将依照从整体到局部再到个体的分析顺序。依次分析服务生态系统全局演化、服务生态系统中的服务社区的演化和服务生态系统中的服务个体的演化。

4.1 服务生态系统全局演化探究

服务生态系统全局演化旨在检测服务生态系统的生成规律是否发生了变化以及发生变化的时间。为了实现这一目的, 使用长度固定为 w 的滑动窗口上的后验贝叶斯因子来检测相对于窗口内某一固定的服务生态系统网络的 WDCSBM 表示是否发生了变化, 以及变化发生的时间。

后验贝叶斯因子[32]是观测数据在零假设模型和选择假设模型下的似然函数比值。在零假设模型中, 服务生态系统在给定时间窗口内没有发生显著演化, 时间窗口内所有的服务生态快照 $\{G^{\tau-w+1}, G^{\tau-w+2}, \dots, G^{\tau}\}$ 可以由同一套超参集合 $\varphi^{(0)} = \{\boldsymbol{\omega}^{(0)}, \mathbf{k}^{(0)}, \boldsymbol{\gamma}^{(0)}\}$ 生成, 通常 $\varphi^{(0)}$ 取值可以为窗口内任意一个服务生态系统快照的超参取值。在选择假设模型中, 服务生态系统在时刻 t_c 发生了演化, 那么在该时间窗口内, t_c 之前的服务生态系统快照 $\{G^{\tau-w+1}, G^{\tau-w+2}, \dots, G^{t_c-1}\}$ 由超参集合 $\varphi^{(1)} = \{\boldsymbol{\omega}^{(1)}, \mathbf{k}^{(1)}, \boldsymbol{\gamma}^{(1)}\}$ 控制生成, 而 t_c 之后的服务生态

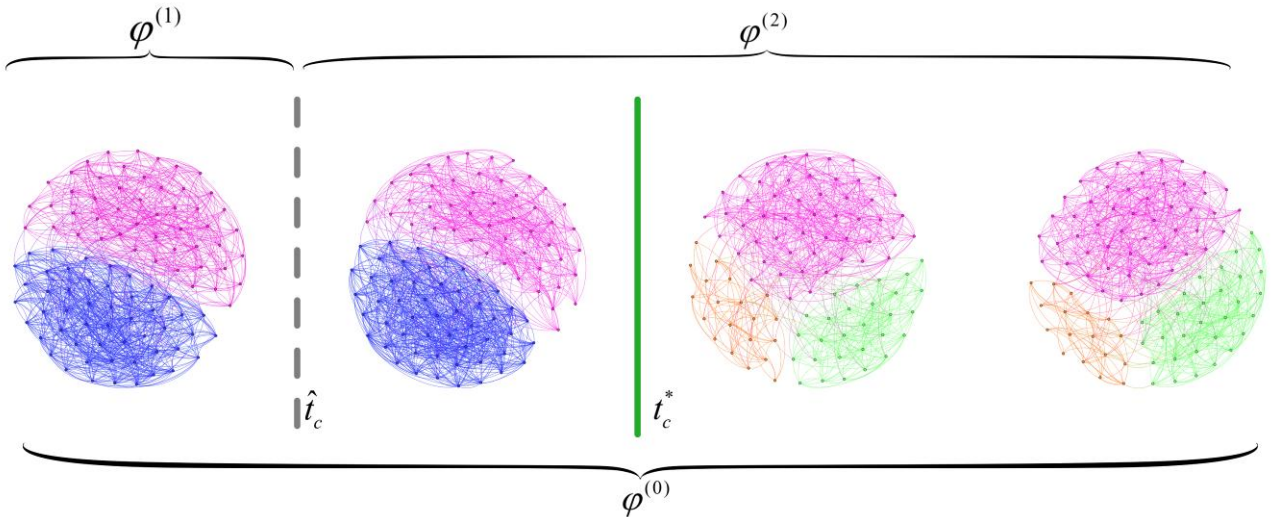


图 2 基于后验贝叶斯因子的给定滑动窗口内的全局演化点探究

系统快照 $\{G^t, G^{t+1}, \dots, G^\tau\}$ 由超参集合

$\varphi^{(2)} = \{\omega^{(2)}, \mathbf{k}^{(2)}, \gamma^{(2)}\}$ 控制生成。 $\varphi^{(1)}$ 和 $\varphi^{(2)}$ 中参数的取值方法如公式 (16):

$$\begin{aligned} \omega_{r,s}^{(1)} &= \sum_{t=\tau-w+1}^{t_c-1} \omega_{r,s}^t, k_u^{(1)} = \sum_{t=\tau-w+1}^{t_c-1} k_u^t, \gamma_{r,s}^{(1)} = \sum_{t=\tau-w+1}^{t_c-1} \gamma_{r,s}^t \\ \omega_{r,s}^{(2)} &= \sum_{t=t_c}^{\tau} \omega_{r,s}^t, k_u^{(2)} = \sum_{t=t_c}^{\tau} k_u^t, \gamma_{r,s}^{(2)} = \sum_{t=t_c}^{\tau} \gamma_{r,s}^t \end{aligned} \quad (16)$$

这样, 在给定以 WDCSBM 形式表示服务生态系统快照序列 $\{G^{\tau-w+1}, G^{\tau-w+2}, \dots, G^\tau\}$ 的后验贝叶斯因子表示如公式 (17):

$$\begin{aligned} \Lambda_{t_c} &= \ln \frac{\prod_{t=\tau-w+1}^{t_c-1} P(\mathbf{A}^t, \mathbf{x}^t | \mathbf{b}^t, \varphi^{(1)}) \prod_{t=t_c}^{\tau} P(\mathbf{A}^t, \mathbf{x}^t | \mathbf{b}^t, \varphi^{(2)})}{\prod_{t=\tau-w+1}^{\tau} P(\mathbf{A}^t, \mathbf{x}^t | \mathbf{b}^t, \varphi^{(0)})} \\ &= \sum_{t=\tau-w+1}^{t_c-1} \ln P(\mathbf{A}^t, \mathbf{x}^t | \mathbf{b}^t, \varphi^{(1)}) + \sum_{t=t_c}^{\tau} \ln P(\mathbf{A}^t, \mathbf{x}^t | \mathbf{b}^t, \varphi^{(2)}) \\ &\quad - \sum_{t=\tau-w+1}^{\tau} \ln P(\mathbf{A}^t, \mathbf{x}^t | \mathbf{b}^t, \varphi^{(0)}) \end{aligned} \quad (17)$$

那么, 对于给定的包含 w 个服务生态网络快照最后时间点为 τ 的滑动窗口 $\{G^{\tau-w+1}, G^{\tau-w+2}, \dots, G^\tau\}$ 中发生了显著演化的分数 $score_\tau$ 为:

$$score_\tau = \max_{\tau-w+1 < t_c \leq \tau} \Lambda_{t_c} \quad (18)$$

该窗口内潜在的演化点则是使得窗口内后验贝叶斯系数 Λ_{t_c} 取得最大值的时间点 t_c^* :

$$t_c^* = \arg \max_{\tau-w+1 < t_c \leq \tau} \Lambda_{t_c} \quad (19)$$

图 2 给出了上述过程的示意图, 其中节点的颜色表示节点所属的社区, 绿色实线表示窗口内的潜在演化点, 灰色虚线表示的是测试的时间点。

通过对窗口进行步长为 $w-1$ 的滑动, 使得能够测试完所有的可能 t_c 并且仅测试一次。当整个时间线中发生演化的次数 N 已知时, 可以认为演化得分前 N 的演化窗口发生了演化; 当演化发生次数未知时, 通常认为演化得分前 $h\%$ 的演化窗口发生了演

化。演化点则是窗口内对应的潜在演化点 t_c^* 。

4.2 服务生态系统中的服务社区演化探究

对于服务生态系统中服务社区层面的演化, 本文主要关注检测相邻时刻发生在服务社区自身及服务社区之间的社区演化事件。演化事件通常是指由 Palla 等人[33]定义的形成, 维持, 发展, 衰退, 分裂, 合并以及死亡。

利用 WDCSBM 进行表示的服务生态系统快照中参数 ω, γ 天然的被用来描述服务社区内部级服务社区之间的连接性质, 这些性质可以用来帮助快速检测可能发生的服务社区演化事件。服务社区 r 在时刻 t 形成并融入 *ProgrammableWeb* 服务生态系统意味着在 $t-1$ 时刻, 服务社区 r 自身内部没有合作关系且与其他服务社区产生交互; 在 t 时刻服务社区 r 中的 API 节点开始与其他 API 节点合作形成 *Mashup*。从而, 服务社区形成演化事件利用 ω, γ 进行表示可以表示如下:

$$\text{形成: } \begin{cases} \sum_{s \leq B} \omega_{r,s}^{t-1} = 0 \\ \sum_{s \leq B} \omega_{r,s}^t \neq 0 \end{cases} \quad \text{或} \quad \begin{cases} \sum_{s \leq B} \gamma_{r,s}^{t-1} = 0 \\ \sum_{s \leq B} \gamma_{r,s}^t \neq 0 \end{cases} \quad (20)$$

服务社区的死亡演化事件和服务社区的形成是相互对立的, 当一个服务社区 r 在时刻 t 失去了社区内部以及社区间的所有合作关系时, 则称服务社区 r 在 t 时刻死亡。服务社区的死亡演化事件可以表示为:

$$\text{死亡: } \begin{cases} \sum_{s \leq B} \omega_{r,s}^{t-1} \neq 0 \\ \sum_{s \leq B} \omega_{r,s}^t = 0 \end{cases} \quad \text{或} \quad \begin{cases} \sum_{s \leq B} \gamma_{r,s}^{t-1} \neq 0 \\ \sum_{s \leq B} \gamma_{r,s}^t = 0 \end{cases} \quad (21)$$

服务社区的合并是通常是指在某一时刻 t 之后, 服务社区 r 和服务社区 s 组合成一个新的服务社区。然而, 在 *ProgrammableWeb* 中给定的前提是服务社区是固定的, 所以服务社区合并 in *ProgrammableWeb* 中表现为两个服务社区 r, s 无法区分彼此, 即两社区内部及社区间的参数相似。利用 ω, γ 参数可以进行如下表示:

$$\text{合并: } \begin{cases} \text{var}(r, s, \omega^{t-1}) \square \text{var}(r, s, \omega^t) \\ \text{var}(r, s, \gamma^{t-1}) \square \text{var}(r, s, \gamma^t) \end{cases} \quad (22)$$

其中 $\text{var}(\bullet)$ 函数表示, 两社区内部及两社区之间的超参值的方差, 其表达式如下所示:

$$\text{var}(r, s, \omega) = \frac{1}{4} \sum_{i \in \{s, r\}} \sum_{j \in \{s, r\}} \sum_{p \in \{s, r\}} \sum_{q \in \{s, r\}} \frac{1}{2} (\omega_{i,j} - \omega_{p,q})^2 \quad (23)$$

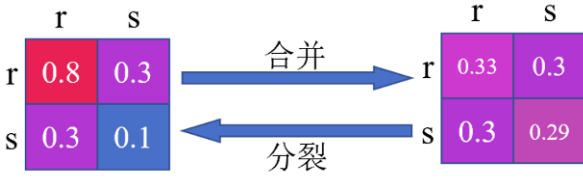


图 3 服务社区的合并与分裂演化事件示意图

服务社区的分裂演化事件则是和服务社区的合并演化事件相对立。服务社区的分裂在 *ProgrammableWeb* 中表现为两个服务社区在 t 时刻之前不能被区分, 在 t 时刻之后可以显著进行区分。利用 ω, γ 参数可以进行如下表示:

$$\text{分裂:} \begin{cases} \text{var}(r, s, \omega^{t-1}) \square \text{var}(r, s, \omega^t) \\ \text{var}(r, s, \gamma^{t-1}) \square \text{var}(r, s, \gamma^t) \end{cases} \quad (24)$$

图 3 给出了服务社区合并与分裂演化事件的示意图。需要指出的是, 基于 *ProgrammableWeb* 服务生态系统的特点, 本文仅考虑 2 个服务社区之间的合并与分裂。公式(22)-(24)可以容易的扩展到多个服务社区之间的合并与分裂。

服务社区的维持、发展以及衰退通常是指的服务社区中包含服务数量的的变化情况。在 *ProgrammableWeb* 服务生态系统中服务社区划分结果已知的情况下, 这些变化可以通过直接观察不同时刻服务生态系统的社区划分结果 \mathbf{b} 获得。因此, 本文中不再详细探讨这三种演化事件。

4.3 服务生态系统中的服务个体演化探究

对于服务个体的演化探究中, 本文所关注的是服务自身发展过程中如何划分出不同的演化阶段, 即发现服务个体在演化过程中的演化点。每一个演化阶段都是服务个体自身行为表现相对稳定的一段时间, 所以每个阶段的行为理论上可以看成是一个线性过程。

WDCSBM 进行表示的服务生态系统快照中参数 \mathbf{k} 可以用来表示服务个体的行为。所以上述问题可以被形式化的表示如下: 在给定服务个体 u 的行为序列 $y_u = \{k_u^1, k_u^2, \dots, k_u^T\}$, 该行为序列 y_u 是片状静止的, 也就是 y_u 的某些特征会在一些未知的时间点

$t_1^* < t_2^* < \dots < t_{K^*}^*$ 发生显著变化, 其中演化次数 K^* 是未知的, 任务是发现一个最合适的划分 $\mathbf{T} = \{t_1^*, t_2^*, \dots, t_{K^*}^*\}$ 使得给定的量化标准 $V(\mathbf{T}, y_u)$ 最小化:

$$V(\mathbf{T}, y_u) = \sum_{i=0}^{K^*} c(k_u^{t_i \dots t_{i+1}}) \quad (25)$$

其中, $c(\bullet)$ 是损失函数用来评估特定模型对于子行为序列 $k_u^{t_i \dots t_{i+1}} = \{k_u^t\}_{t_i}^{t_{i+1}}$ 的拟合优度 (goodness-of-fit)[34]。由于在寻找行为序列中的线性过程, 所以最小绝对偏差 (L1, Least absolute deviation) 被选作为损失函数, 即:

$$c(y^{a \dots b}) = \sum_{i=a}^b \|y_i - \bar{y}\| \quad (26)$$

其中, \bar{y} 是子序列 $y^{a \dots b}$ 的中位数。此外, 由于演化次数 K^* 未知, 为了避免产生过多不要的划分, 需要对划分施加一个惩罚 $\text{pen}(\mathbf{T})$:

$$\text{pen}(\mathbf{T}) = \beta |\mathbf{T}| \quad (27)$$

其中 $\beta > 0$ 是一个平滑参数用来平衡划分的复杂程度和拟合优度。 β 值越小越倾向于划分出更多的演化点, β 值越高越倾向于舍弃变化较小的演化点保留变化程度较大的演化点。因此, 最后需要优化的目标可以表示成:

$$\min_{\mathbf{T}} V(\mathbf{T}, y) + \text{pen}(\mathbf{T}) \quad (28)$$

K^* 未知使得在解空间中暴力搜寻准确的划分 \mathbf{T} 在时间上变得不可能。所以本文引入了修剪的精确线性时间 (Pelt, Pruned Exact Linear Time) 算法[35] 来寻找不确定演化点个数的划分 \mathbf{T} 。该方法依次考虑每个可能的时间点, 通过一个明确的剪枝规则来判定该时间点是否会是潜在的演化点。具体来说, 对于给定的两时间点 t_1 和 t_2 ($t_1 < t_2 < T$), 剪枝规则如下:

$$\left[\min_{\mathbf{T}} V(\mathbf{T}, y^{0 \dots t_2}) + \text{pen}(\mathbf{T}) \right] \leq \left[\min_{\mathbf{T}} V(\mathbf{T}, y^{0 \dots t_1}) + \text{pen}(\mathbf{T}) \right] + c(y^{t_1 \dots t_2}) \quad (29)$$

若式(29)成立, 则 t_1 不可能是 T 时刻之前的最后一个演化点。在惩罚 $\text{pen}(\bullet)$ 是线性时, Pelt 算法复杂

度是 $O(T)$ ，Pelt 算法的详细步骤见算法 1。

算法 1 Pelt 算法

输入： 行为序列 $\{y^t\}_{t=1}^T$ ，损失函数 $c(\bullet)$ ，平滑参数 β

1. 初始化一个长度为 $T+1$ 的数组 Z ; $Z[0] \leftarrow -\beta$
2. 初始化 $L[0] \leftarrow \emptyset$
3. 初始化可搜寻的索引集合 $\chi \leftarrow \{0\}$
4. **FOR** $t=1,2,\dots,T$ **DO**
5. $\hat{t} \leftarrow \arg \min_{s \in \chi} [Z[s] + c(y^{s:t}) + \beta]$
6. $Z[t] \leftarrow [Z[\hat{t}] + c(y^{\hat{t}:t}) + \beta]$
7. $L[t] \leftarrow L[\hat{t}] \cup \{t\}$
8. $\chi \leftarrow \{s \in \chi : Z[s] + c(y^{s:t}) \leq Z[t]\} \cup \{t\}$
9. **END FOR**

输出： 演化点索引集合 $L[T]$

在获取准确的划分 T 之后，其中包含 K^*+1 个演化阶段，每个演化阶段都是一个线性函数，易求得对于第 i 个演化阶段其斜率为 α_i ，行为子序列均值为

\bar{y}_i 。基于演化阶段斜率的值和均值可以将演化阶段

总结成如 **Error! Reference source not found.:**

表 2 演化阶段及判定条件

演化阶段	判定条件
退出期	$\alpha_i = 0$ 且 $\bar{y}_i = 0$
稳定期	$ \alpha_i < \delta$ 且 $\bar{y}_i \neq 0$
发展期	$\alpha_i > \delta$
衰退期	$\alpha_i < -\delta$

其中 δ 是一个接近 0 的正实数。

5 实验与分析

5.1 数据集及实验设置

本文选择校正后的 *ProgrammableWeb* 数据集¹ 作为实验数据集，相对于原始的 *ProgrammableWeb* 数据集，该数据集修正了服务的可访问状态及失效时间。该数据集中总共包含了 23,678 个 API 和 7,766 个 mashup。其中未与其他 API 共同组成 mashup 的

API 以及由少于 2 个 API 构成的 mashup 被从数据集中移除，因为这些 API 和 Mashup 是孤立于整个 *ProgrammableWeb* 服务生态系统的。基于过滤后的数据，从 2006 年 9 月 10 日到 2020 年 9 月 26 日每隔 30 天生成一个 API-API 图的快照，共有 172 个快照。API 的主类别被用来作为 API 所属的划分结果，共有 129 个不同类别，即 $B=129$ 。

基于在第 4 章中对参数物理含义以及参数影响的讨论，在全局演化实验中，滑动窗口的大小 w 被设置成 3，滑动步长则为 2。 h 的值被设置为 5，即只有 5% 的滑动窗口内发生了显著演化。在个体演化实验中，演化点划分时平滑参数 β 设置我们测试了 $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 10\}$ ，最终发现当 β 设置 60 时可以获取到充足的演化点同时避免过多微小变化的演化点。在演化阶段判定时， δ 被设置成 0.02 以容忍微小波动。

由于本文是首篇从多层次进行服务生态研究的文章，现有服务生态演化研究[19]-[25]侧重从语义概念层面对服务生态系统进行描述以及通过服务生态网络性质的统计来为下游任务提供输入或先验知识，然而这些方法无法被直接用来解决本文关注的服务生态系统的全局演化探究、服务社区演化探究以及服务个体演化探究。所以，本文通过数学推导、可视化实验结果及人工专家验证的方式来说明本文提出方法的有效性。数学推导部分在章节 3 和章节 4 中已经详细论述，本章的后续内容将展示本文提出方法的可视化实验结果和人工专家验证结果。

5.2 全局演化分析

图 4 展示了 *ProgrammableWeb* 服务生态系统全局演化的分析结果。图中蓝色的散点表示该时刻发生显著演化的得分 *score*，每相邻的 3 个散点处于同一个演化窗口内，图 4 中利用矩形标注出了其中 3 个具有代表性的演化窗口并给出了演化窗口前后 *ProgrammableWeb* 服务生态系统可视化的结果。其中两个绿色矩形框出的是发生了最为显著演化的演化窗口，而红色矩形框出的则是未发生显著演化最为稳定的演化窗口。

从图 4 可视化的服务生态系统网络来看，本文中提出的利用后验贝叶斯因子来寻找全局演化点的方法的确可以发现服务生态系统中在整体拓扑结构上面的剧烈演化。例如，图 4 中利用红色矩形标记出来的演化窗口总体得分最低。并且从可视化

¹ <https://github.com/HIT-ICES/Correted-ProgrammableWeb-dataset>

结果来看, 2008年11月28日的服务生态系统快照和2009年2月26日的服务生态系统快照的网络拓扑结构基本一致。所以可以断言在2008年11月28日到2009年2月26日期间, 整个 *ProgrammableWeb* 服务生态系统没有发生全局性剧烈演化。然而两个

由绿色矩形标记出来得分整体最高的两个演化窗口中, 图4给出了窗口内最高得分时间点前后的服务生态系统快照的可视化结果, 可以很明显的看出来拓扑结构发生了巨大的变化, 且得分总体越高的演化窗口内发生的整体演化越明显。

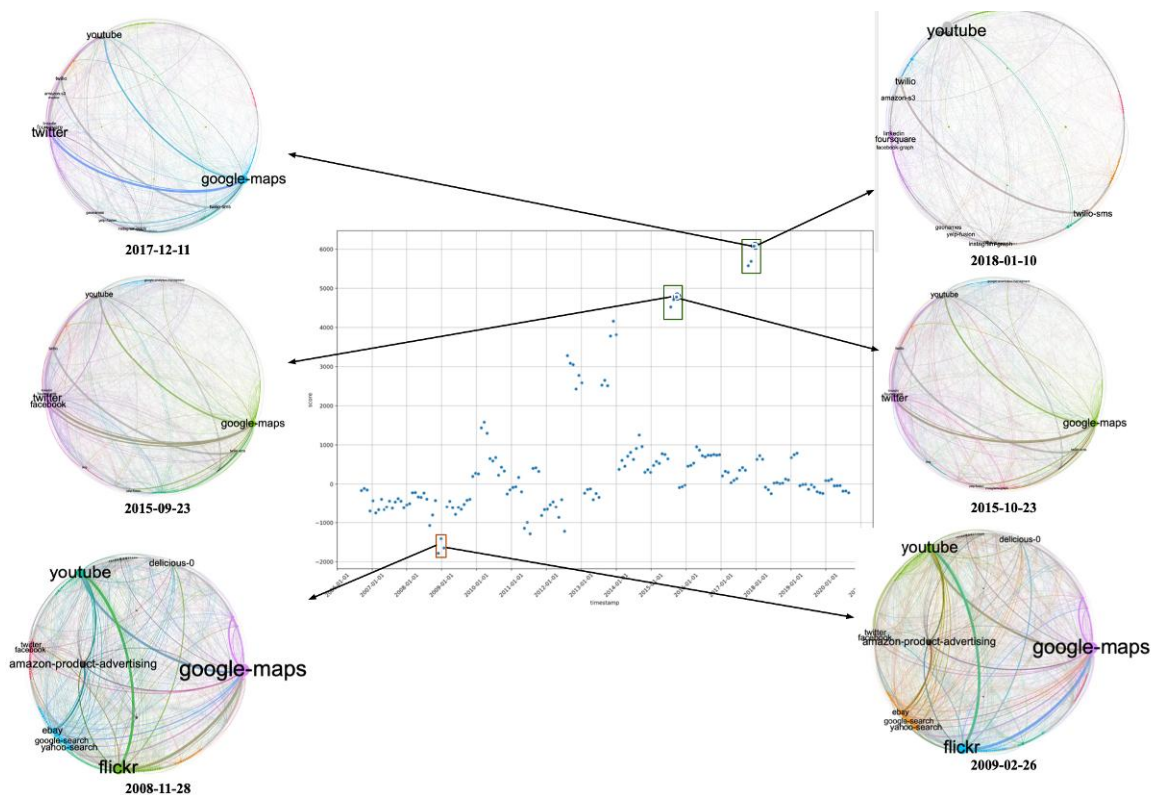


图4 *ProgrammableWeb* 服务生态系统全局演化结果。绿色矩形框表示发生了全局演化的窗口, 使用圆圈圈出的点表示全局演化点。

比如, 相对于2017年12月11日的服务生态系统快照, 2018年1月10日的服务生态系统快照从网络拓扑结构上来说密度明显降低, 一些巨人节点 (google-maps, twitter) 退出了服务生态。

进一步的对标记出来的两次全局演化前后服务生态系统的可视化结果进行分析与领域专家人工核验, 可以发现这两次全局演化都是由服务生态中的巨人节点的退出引发的。这说明了 *ProgrammableWeb* 服务生态系统是一个脆弱的系统。

5.3 服务社区演化分析

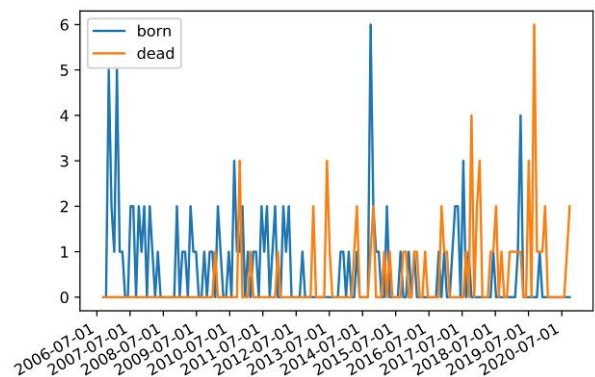


图5 *ProgrammableWeb* 服务生态系统中服务社区的形成与死亡随时间变化图

在 2006 年 9 月 10 日到 2020 年 9 月 26 日期间, *ProgrammableWeb* 服务生态系统中总共形成了 104 个新服务社区, 同时也有 66 个服务社区死亡, 且所有死亡的服务社区没有出现过复活现象。**Error! Reference source not found.** 展示了 *ProgrammableWeb* 服务生态系统中服务社区的形成与死亡随时间的变化情况。在 2010 年之前整个 *ProgrammableWeb* 服务生态系统的服务生态系统的多样性是在持续丰富的, 因为再次之前不断有新的服务社区产生但没有服务社区死亡。在 2010 年到 2015 年 7 月间, 新形成的服务社区和死亡的服务社区数量基本持平, 并且在 2014 年 9 月左右有一次新服务社区形成的高潮。自 2015 年 7 月之后, 新形成的服务社区数量变少, 但服务社区的死亡却变得频繁。造成这一现象的原因可能是, 自 2015 年之后移动服务的快速发展, 使得大量的服务提供者和开发者将 Web 服务转换成移动服务²。

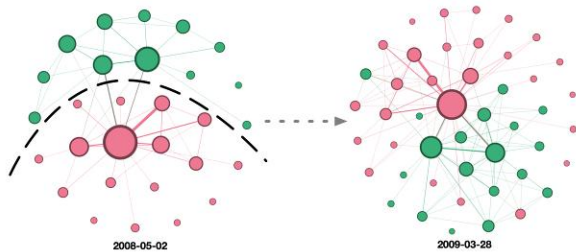


图 6 Social 服务社区和 Mapping 服务社区的合并

对于服务社区的合并和分裂演化事件, 在时间跨度为 30 天 (相邻的服务生态系统快照) 时没有发现任何显著的合并或分裂行为。这说明服务社区间的演化少有突发性的剧烈的演化。当扩大观察时间间隔, 可以发现一些服务社区的合并或分裂的演化事件。图 6 展示了 Social 服务社区和 Mapping 服务社区之间关系的演化, 其中粉色节点表示 Social。在 2008 年 5 月 2 日的服务生态系统快照中, Social 和 Mapping 两服务社区之间的超参 ω, γ 的方差 $\text{var}(\bullet)$ 值分别为 0.142 和 0.027, 而在 2009 年 3 月 28 日的服务生态系统快照中两超参的方差值变为了 0.019 和 0.011。依据式(22)可以认为 Social 服务社区和 Mapping 服务社区发生了合并演化事件。

表 2 Social 服务社区和 Mapping 服务社区超参方差随时间变化

	2008-05-02	2008-07-31	2008-10-29	2009-01-27	2009-03-28
$\text{var}(\omega)$	0.141	0.086	0.078	0.045	0.019
$\text{var}(\gamma)$	0.027	0.012	0.012	0.016	0.011

从图 6 的可视化结果来看, 在 2008 年 5 月 2 日的服务生态系统快照中, 两服务社区之间有明显的边界 (黑色虚线), 且两社区内部的拓扑结构也显著不同。但在 2009 年 3 月 28 日的服务生态系统快照中, 两服务社区之间的边界变得模糊。

5.4 服务个体演化分析

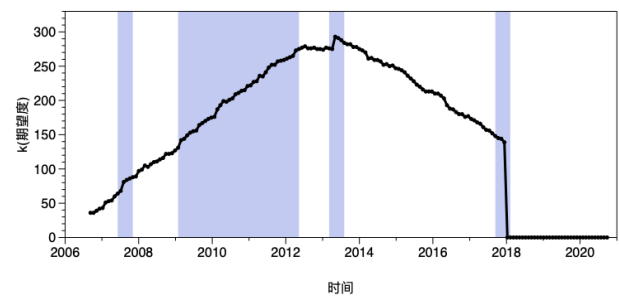


图 7 google-maps 演化阶段划分结果, k 表示 google-maps 的期望度 (每隔 30 天记录一次)

Error! Reference source not found. 以 *google-maps* 为例展示了服务个体演化阶段划分的效果。图中相邻两点之间的时间间隔为 30 天, 使用蓝色和白色来对不同的演化阶段进行划分, 蓝色和白色交界处则是服务个体行为的演化点。从整体演化趋势来看, *google-maps* 先整体上处于发展阶段, 期间发展速度经历了几次变化, 随后进入了一个维持时间较短的稳定期, 在之后发生了一次短时间的增长之后就进入了持续的衰退期, 直至因为由于分裂成其他 API 而退出服务生态。当将图 4 和图 7 的结果进行比较时, 可以发现 *google-maps* 退出服务生态的时间恰好是图 4 中服务生态系统发生全局演化的时间点。当发生全局演化时, 若可视化结果难以展示或可视化结果难以直接看出演化原因时, 可以通过服务个体的行为表现来推测服务生态发生全局演化的原因。此外, 对大量 API 服务个体的演化趋势进行分析后, 我们发现自 2014 年之后 API 服务个体整体演化趋势呈现下降状态。通过领

² <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-Google-play-store/>

域专家的人工核验，API 服务呈现下降的演化趋势是由于移动互联网的兴起引发的。

在对每一个 API 服务个体进行演化阶段划分后，本文总结了它们演化周期内频繁出现的由邻接演化阶段组成的演化模式。图 8 给出了这些服务个体演化模式的示意图。图 8.a 表示的是 API 服务个体的突变演化模式，意味着个体服务要么长时间没

在对每一个 API 服务个体进行演化阶段划分后，本文总结了它们演化周期内频繁出现的由邻接演化阶段组成的演化模式。图 8 给出了这些服务个体演化模式的示意图。图 8.a 表示的是 API 服务个体的突变演化模式，意味着个体服务要么长时间没有变化要么就是突然变到另一个状态，这种演化模式通常出现在不太被关注的 API 服务上。图 8.g-图 8.i 则是展示了个体服务的加速(减速)发展(衰退)的演化模式。从图 8.f 中可以看出，从发展期进入衰退期是一种常见的演化模式但是从衰退期进入发展期却很少见。同样的图 8.b, 图 8.k 表示了类似的情况。这些演化模式可以帮助对 API 服务个体的后续发展状况进行预测或用于支持其它下游任务。例如，每个演化模式共包含三个参数 S, DT ，和 α ，其中 S 用来表示该演化模式中包含的演化阶段的序列， DT 表示 S 中每个演化阶段的持续时间， α 则表示 S 中每个演化阶段的斜率。综上所述，每个演化模式都可以表示为 $EP = \langle S, DT, \alpha \rangle$ 。对于给定的 API 服务个体，它的演化轨迹可以通过一组演化模式以及不同的参数取值来表示。这样的参数取值表达可以作为不同应用场景下规范化的输入向量进行灵活使用。

有变化要么就是突然变到另一个状态，这种演化模式通常出现在不太被关注的 API 服务上。图 8.g-图 8.i 则是展示了个体服务的加速(减速)发展(衰退)的演化模式。从图 8.f 中可以看出，从发展期进入衰退期是一种常见的演化模式但是从衰退期进入发展期却很少见。同样的图 8.b, 图 8.k 表示了类似的情况。这些演化模式可以帮助对 API 服务个体的后续发展状况进行预测或用于支持其它下游任务。例如，每个演化模式共包含三个参数 S, DT ，和 α ，其中 S 用来表示该演化模式中包含的演化阶段的序列， DT 表示 S 中每个演化阶段的持续时间， α 则表示 S 中每个演化阶段的斜率。综上所述，每个演化模式都可以表示为 $EP = \langle S, DT, \alpha \rangle$ 。对于给定的 API 服务个体，它的演化轨迹可以通过一组演化模式以及不同的参数取值来表示。这样的参数取值表达可以作为不同应用场景下规范化的输入向量进行灵活使用。

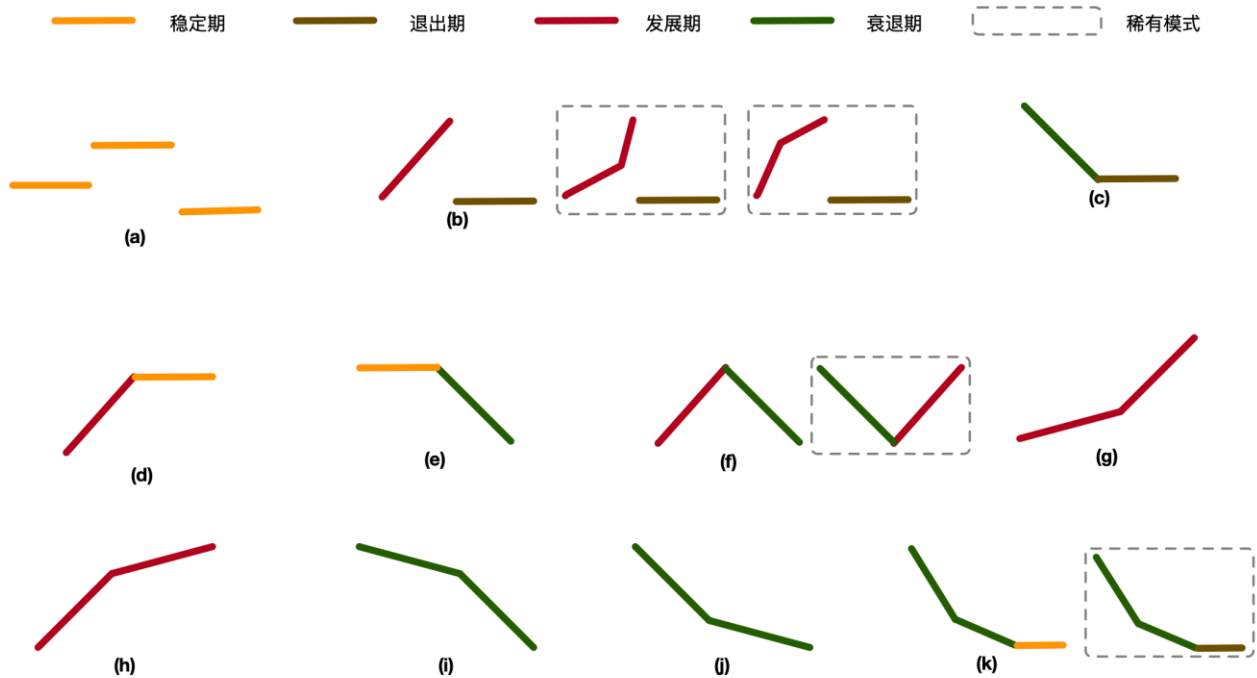


图 8 服务个体演化模式

6 结束语

本文利用一种带权重的度修正随机块模型对 ProgrammableWeb 服务生态系统进行了表示。基于

对带权重的度修正随机块模型中参数物理含义解释及求解，本文从服务生态整体、服务社区以及服务个体三个层面对 ProgrammableWeb 服务生态系统进行了演化发现。服务生态系统整体演化点的检测可以帮助更好的理解的服务生态系统当前所处的状态；服务社区演化事件的检测可以帮助后续进

行服务社区演化行为预测，从而进一步指导服务组合、服务推荐等下游任务。服务个体的演化阶段划分与演化模式的总结将帮助服务提供者预测 API 服务个体的后续发展状况，从而更好的调整他们的业务策略。

参考文献

- [1] Ngu A H H, Carlson M P, Sheng Q Z, et al. Semantic-based mashup of composite applications[J]. *IEEE Transactions on Services Computing*, 2010, 3(1): 2-15.
- [2] Brown A, Fishenden J, Thompson M. API economy, ecosystems and engagement models[M]//*Digitizing Government*. Palgrave Macmillan, London, 2014: 225-236.
- [3] Bouguettaya A, Singh M, Huhns M, et al. A service computing manifesto: the next 10 years[J]. *Communications of the ACM*, 2017, 60(4): 64-72.
- [4] Sheng Q Z, Qiao X, Vasilakos A V, et al. Web services composition: A decade's overview[J]. *Information Sciences*, 2014, 280: 218-238.
- [5] Cao Bu-Qing, Xiao Qiao-Xiang, Zhang Xiang-Ping, Liu Jian-Xun. An API Service Recommendation Method via Combining Self-Organization Map-Based Functionality Clustering and Deep Factorization Machine-Based Quality Prediction [J]. *Journal of Computers*, 2019,42(06):1367-1383. (in Chinese)
(曹步清,肖巧翔,张祥平,刘建勋.融合SOM功能聚类与DeepFM质量预测的API服务推荐方法[J].*计算机学报*,2019,42(06):1367-1383.)
- [6] Ma Y, Geng X, Wang J. A Deep Neural Network With Multiplex Interactions for Cold-Start Service Recommendation[J]. *IEEE Transactions on Engineering Management*, 2020, 68(1): 105-119.
- [7] Botangen K A, Yu J, Sheng Q Z, et al. Geographic-aware collaborative filtering for web service recommendation[J]. *Expert Systems with Applications*, 2020, 151: 113347.
- [8] Adeleye O, Yu J, Yongchareon S, et al. A fitness-based evolving network for web-apis discovery[C]//*Proceedings of the australasian computer science week multiconference*. 2019: 1-10.
- [9] Shi Min, Liu Jian-Xun, Zhou Dong, Cao Bu-Qing, Wen Yi-Ping. Multi-Relational Topic Model-Based Approach for Web Services Clustering [J]. *Journal of Computers*, 2019,42(04):820-836. (in Chinese)
(石敏,刘建勋,周栋,曹步清,文一凭.基于多重关系主题模型的Web服务聚类方法[J].*计算机学报*,2019,42(04):820-836.)
- [10] Tian Y, Kochhar P S, Lo D. An exploratory study of functionality and learning resources of web apis on programmableweb[C]//*Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*. 2017: 202-207.
- [11] Chen Z, Shen L, Li F. Exploiting Web service geographical neighborhood for collaborative QoS prediction[J]. *Future Generation Computer Systems*, 2017, 68: 248-259.
- [12] Chen Z, Sun Y, You D, et al. An accurate and efficient web service QoS prediction model with wide-range awareness[J]. *Future Generation Computer Systems*, 2020, 109: 275-292.
- [13] Weiss M, Gangadharan G R. Modeling the mashup ecosystem: structure and growth[J]. *R&D Management*, 2010, 40(1):40-49.
- [14] Adeleye O, Yu J, Yongchareon S, et al. Constructing and evaluating an evolving web-api network for service discovery[C]//*International Conference on Service-Oriented Computing*. Springer, Cham, 2018: 603-617.
- [15] Huang K, Fan Y, Tan W. Recommendation in an Evolving Service Ecosystem Based on Network Prediction[J]. *IEEE Transactions on Automation Science & Engineering*, 2014, 11(3):906-920.
- [16] Fokaefs M, Stroulia E. Wsdarwin: Studying the evolution of web service systems[M]//*Advanced Web Services*. Springer, New York, NY, 2014: 199-223.
- [17] Zhao Xue-Hua, Yang Bo, Chen He-Chang. Fast Learning Algorithm for Stochastic Blockmodel [J]. *Journal of Software*, 2016, 27(9):2248-2264.. (in Chinese)
(赵兴华, 杨博, 陈贺昌. 一种高效的随机块模型学习算法[J]. *软件学报*, 2016, 27(9):2248-2264.)
- [18] Yao L, Sheng Q Z, Ngu A, et al. Unified Collaborative and Content-Based Web Service Recommendation[J]. *IEEE Transactions on Services Computing*, 2015, 8(3):453-466.
- [19] Wang J, Chen H, Yu Z. Mining user behavior pattern in mashup community[C]// *IEEE International Conference on Information Reuse & Integration*. IEEE, 2009.
- [20] Yang Z, Fan Y, Huang K, et al. Time-Aware Service Recommendation for Mashup Creation in an Evolving Service Ecosystem[C]// *IEEE International Conference on Web Services*. IEEE Computer Society, 2014.
- [21] Bing B, Fan Y, Wei T, et al. SR-LDA: Mining Effective Representations for Generating Service Ecosystem Knowledge Maps[C]// *IEEE International Conference on Services Computing*. IEEE, 2017.
- [22] Gan Y, Xiang Y, Zou G, et al. Multi-label Recommendation of Web Services with the Combination of Deep Neural Networks[M]. 2019.
- [23] Lyu S, Liu J, Tang M, et al. Three-Level Views of the Web Service Network: An Empirical Study Based on ProgrammableWeb[C]// *IEEE International Congress on Big Data*. IEEE, 2014.
- [24] Jr A, Kadiyalax H, Bo H, et al. Supporting Microservice Evolution[C]// *IEEE International Conference on Software Maintenance & Evolution*. IEEE, 2017.
- [25] Wang X, Feng Z, Chen S, et al. Dkem: A distributed knowledge based evolution model for service ecosystem[C]//*2018 IEEE International Conference on Web Services (ICWS)*. IEEE, 2018: 1-8.

- [26] Xue X, Wang S, Zhang L, et al. Social learning evolution (SLE): computational experiment-based modeling framework of social manufacturing[J]. IEEE Transactions on Industrial Informatics, 2018, 15(6): 3343-3355.
- [27] Xue X, Guo Y, Chen S, et al. Analysis and controlling of manufacturing service ecosystem: a research framework based on the parallel system theory[J]. IEEE Transactions on Services Computing, 2019.
- [28] Liu M, Tu Z, Xu X, et al. A Data-driven Approach for Constructing Multilayer Network-based Service Ecosystem Models[J]. arXiv preprint arXiv:2004.10383, 2020.
- [29] Liu M, Tu Z, Wang J, et al. A Novel Multi-layer Network Model for Service Ecosystems[C]// 2020 International Conference on Service Science (ICSS). 2020.
- [30] Zhao, Yunpeng. A survey on theoretical advances of community detection in networks[J]. Wiley Interdisciplinary Reviews: Computational Statistics, 2017:e1403.
- [31] Chai Bian-Fang, Yu Jian, Jia Cai-Yan, Wang Jing-Hong. Fast Algorithm on Stochastic Block Model for Exploring General Communities[J]. Journal of Software, 2013, 000(011):2699-2709.. (in Chinese)
- (柴变芳, 于剑, 贾彩燕,等. 一种基于随机块模型快速广义社区发现算法[J]. 软件学报, 2013, 000(011):2699-2709.)
- [32] Wang Y, Chakrabarti A, Sivakoff D, et al. Hierarchical change point detection on dynamic networks[C]//Proceedings of the 2017 ACM on Web Science Conference. 2017: 171-179.
- [33] Palla G, Barabási A L, Vicsek T. Quantifying social group evolution[J]. Nature, 2007, 446(7136): 664-667.
- [34] Landaburu E, Pardo L. Matching moments for a closer approximation of the weighted (h, φ) -divergence test statistics in goodness-of-fit for finite samples[J]. Journal of the Franklin Institute, 2020, 342(1):115-129.
- [35] Wambui G D. The Power of the Pruned Exact Linear Time(PELT) Test in Multiple Changepoint Detection[J]. American Journal of Theoretical and Applied Statistics, 2015, 4(6):581.



Liu Ming-Yi, Ph.D Student. His main research interests focus on service computing and knowledge graph.

Tu Zhi-Ying, Ph.D. Associate Professor. His research interest focus on software engineering, service computing, knowledge engineering and enterprise business modeling.

Xu Xiao-Fei, Ph.D. Professor. His research interest focus on service computing, software service engineering, big service and Internet of Service, enterprise intelligent computing and enterprise business modeling.

Wang Zhong-Jie, Ph.D. Professor. His research interest focus on software engineering and service computing.

Background

With the widespread adoption of Service-Oriented Architecture (SOA), more and more services are using the Internet as their publishing and operation platform, exposing their functions in the form of APIs. APIs can be easily combined into larger granular applications (mashups), and these mashups can create and deliver unique new value to users that individual services cannot provide. This growing "service ecosystem" has gradually weakened the concept of traditional business domains, and barriers have opened up, making

cross-border cooperation and cross-border integration more common and having a profound impact on the traditional Internet ecosystem.

Service ecosystem has made it easier for service providers to integrate and innovate across borders, resulting in the creation of an increasing number of new applications. Moreover, as users gradually use these new applications, they may come up with new demands, which not only further accelerates the innovation process, but also intensifies market

competition. Service providers need to be sensitive to changes in user needs and preferences, constantly introducing new services or updating existing ones. When services change, the service ecosystem will also change. It is necessary to conduct research on the evolution of the service ecosystem and discover the laws of its changes from different perspectives.

Most of the existing researches stay on the shallow statistical analysis of the network topology of the ProgrammableWeb service ecosystem, and the evolutionary analysis often focuses on only one level of the service ecosystem as a whole, the service community or the service individuals, and there is a lack of a perfect method to conduct a more comprehensive evolutionary analysis of the service ecosystem from multiple levels at the same time.

This paper proposed an evolutionary analysis method of multi-level service ecosystem based on stochastic block model. The method first models the ProgrammableWeb service ecosystem as a dynamic network, and then describes the dynamic network mathematically and formally using a weighted degree-corrected stochastic block model to parameterize the behavior of the service ecosystem as a whole, the service community and the service individuals. Then, based on the parametric representation of the service ecosystem, the evolution problems of the service ecosystem as a whole, the service community and the service individual are solved separately, including: a global evolution point detection algorithm at the global level of the service ecosystem; an evolution event determination algorithm at the level of the service community; and an evolution stage classification algorithm at the level of the service individual. Finally, experiments are conducted on a real ProgrammableWeb dataset to confirm the feasibility of this paper.

The research in this paper is partially supported by the National Key Research and Development Program of China (No 2018YFB1402500) and the National Natural Science Foundation of China (61772155, 61832004, 61802089, 61832014)