

基于强化学习的 Web 服务众测任务分派方法

唐文君 张佳丽 陈 荣 郭世凯

大连海事大学信息科学技术学院 辽宁 大连 116026

(wjtang@dmlu.edu.cn)



摘 要 如何将众包测试任务分派给合适的众测工人,以较低的成本获得更好的测试结果,是一个重要问题。文中将 CWS 众测任务分派问题建模为一个基于马尔可夫决策过程的问题,且使用 Deep Q Network 进行学习和实时在线测试任务分派。该基于强化学习的方法被命名为 WTA-C。此外,文中根据众测工人执行任务的历史时间,通过统计条件概率计算测试工人在任务期限内完成任务的概率,将其作为工人信誉值来反映工人质量,并在每次分派完成后对工人信誉值进行更新。实验结果显示,WTA-C 在控制测试任务的“质量-成本”权衡和保证工人可靠度方面优于其他基于启发式策略的实时分派方法,并在分派效果上高于各启发式策略 18% 以上,从而证明了其可以更好地适应 CWS 的结构和众测环境的特点。

关键词: 众包测试;组合 Web 服务测试;Web 服务测试;强化学习

中图法分类号 TP311.5

Web Service Crowdtesting Task Assignment Approach Based on Reinforcement Learning

TANG Wen-jun, ZHANG Jia-li, CHEN Rong and GUO Shi-kai

College of Information Science and Technology, Dalian Maritime University, Dalian, Liaoning 116026, China

Abstract How to assign tasks to appropriate workers to get better testing results at a lower cost is an important problem. This paper modeled the CWS testing task assignment as a Markov decision process-based problem, and used Deep Q Network to learn and perform real-time online testing task assignment. The proposed approach based on reinforcement learning is named WTA-C. In addition, this paper calculated the probability of the testing worker completing the task within the duration through statistical conditional probability in accordance with the time of the worker's historical execution of tasks, and used it as the workers' reputation value to reflect their quality. The worker's reputation is updated after each assignment. The experimental results show that WTA-C is superior to other real-time assignment methods based on heuristic strategies in controlling the “quality-cost” trade-off of testing tasks and ensuring worker quality, and its assignment effect is more than 18% higher than that of each heuristic strategy, which demonstrates that WTA-C can better adapt to the structure of the CWS and the characteristics of Crowdsourcing environment.

Keywords Crowdtesting, Composite Web service testing, Web service testing, Reinforcement learning

1 引言

Web 服务是一种关于开发分布式的互操作的应用程序。随着服务功能逐渐复杂,单个 Web 服务常常已经无法完成工作。因此,目前采用将多个 Web 服务以 workflow 形式进行组合的模式来构成组合 Web 服务(Composite Web Service, CWS),以实现更加复杂的功能,提高 Web 服务的应用的程度。

随着 CWS 应用逐渐推广,为确保其满足预期的功能和质量要求,就必须对 CWS 和其中的每个 Web 服务进行测试。目前已存在一些协助测试人员创建和管理 Web 服务的测试

流程的测试工具^[1],测试人员使用测试工具对服务进行测试。然而,诸如此类的传统服务测试工具和人员有诸多限制。首先,Web 服务的运行易受网络环境的影响^[2],而测试工具往往无法覆盖到更多的测试区域,若将服务部署到不同网络环境进行测试,则会增加测试成本。其次,Web 服务测试还需要测试人员具有一定程度的相关知识和能力(如测试用例设计能力、相应的编程能力和测试报告撰写能力等),不能随意雇佣工人来执行测试任务。然而,雇佣大量专业、能力强的测试人员势必会造成成本的上升。因此,传统的测试方法和测试工具难以全面地满足测试需求。

投稿日期:2019-09-10 返修日期:2019-11-11 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61672122, 61902050, 61602077);中央高校基本科研业务费专项基金(3132019355);赛尔创新项目(NGH20190627)

This work was supported by the National Natural Science Foundation of China(61672122, 61902050, 61602077), Fundamental Research Funds for the Central Universities of Ministry of Education of China(3132019355) and CERNET Innovation Project (NGH20190627).

通信作者:陈荣(rchen@dmlu.edu.cn)

近年来出现了一种新的测试方式——众包测试(Crowd-testing),其已被应用于可用性测试^[3]、性能测试^[4]和 GUI 测试^[5]等方面。对于 Web 服务来说,众包测试可以通过互联网在全球范围内招募非特定终端的,具有不同硬件、地理位置、网络条件和防火墙配置的零散劳动力作为众测工人,并以任务的形式将测试工作交由众测工人处理。因此,如何将众测任务分派给更合适的测试人员,尽可能地以较低的测试成本来获得更好的测试结果是一个重要的研究问题。

现有的任务分派方法基于精确式算法或启发式算法,大都是离线方法。它们并不完全适合众包平台的诸多特点:1)众包工人的可靠程度、声誉值与其执行过的任务有关,是动态变化的,如果使用精确式算法,在每次分派时都需要大量的计算;2)工人的声誉值等信息是无法预知的,离线方法无法根据这些动态变化信息来动态调整测试任务的分派策略;3)每个 CWS 都以工作流的形式来组织 Web 服务,任务之间具有时序关系,对其中的服务进行测试是一个基于马尔可夫决策过程的问题,这类问题更适于使用强化学习相关算法来解决。

对于众包测试环境中的 Web 服务测试任务分派问题,我们更希望能够在线、实时地进行测试任务分派。因此,本文将该分派问题建模为一个基于马尔可夫决策过程的问题,提出了一种基于强化学习的 Web 服务众包测试任务分派模型,并构建了任务分派模型 WTA-C,实现实时地为 Web 服务众包测试任务选择合适的众包测试工人,并提升测试效果,提高工作效率。同时,为了保证测试更加有效,我们将每个测试任务分派给多个众测工人而非单个工人来处理,即将众测任务分派给一个工人组,来保证测试更加全面完善。在模型中,将每个任务作为其中的状态(State),将选择的众测工人作为动作(Action),将工人的测试效果作为回报(Reward),使用强化学习中的 Deep Q Network(DQN)算法对模型进行学习训练,将每个任务分派给一组合适的测试工人。在众测平台上,每个工人执行任务的时间符合幂律分布^[6],可根据该规律计算工人在任务期限内完成任务的概率,并将其作为工人的信誉值来反映其可靠程度。为使模型更符合众包的动态变化环境,每次分派后根据对工人执行任务情况的评估来更新工人的信誉值。

本文通过真实数据和基于真实数据生成的仿真数据来进行实验,并与现有的实时众包任务分派方法进行了对比,验证了提出的模型在控制测试任务的“质量-成本”权衡和保证工人可靠度方面都优于其他的实时分派方法。

2 相关工作

2.1 Web 服务测试

对 Web 服务进行测试,可以保证其质量,减小服务在应用时运行出错的概率,尽可能地提前处理掉服务运行中会出现的问题。但是,在服务测试中存在许多难以解决的问题,如冷启动问题、测试人员的不诚实行为和数据稀疏性等。Hussain 等^[1]介绍了 3 种目前常见的开源的 Web 服务测试工具,即 JMeter,soapUI 和 Storm,并从功能性、特点、服务项目以及质量因素(响应时间、吞吐量和可用性等)等方面比较了它们的测试性能。Tian 等^[7]提出了一种基于日志分析的 C/S

体系结构的 Web 服务可靠性测试方法。他们通过自动构建软件使用模型来描述用户通过马尔可夫链访问 Web 服务的实际情况。测试用例根据 Random Walk 方法生成,并应用于软件可靠性测试。总而言之,这些不使用人工的测试方法可能无法获得一些需要手动描述的测试结果。Du 等^[8]考虑到 CWS 的动态性和分布式特点,对 CWS 测试的定义、体系结构、测试方法和测试技术进行了总结和分析。

2.2 众包测试

众包测试是近年来兴起的一种测试方法。它将众包工人作为测试人员,利用互联网上的劳动力对产品^[9]进行更广泛、全面的测试。目前,众包测试已被应用于产品 QoE 测试^[10-11]、软件评测^[12-13]、程序调试^[14-16]、产品 GUI 测试^[5]等方面。然而,目前众包测试面临着许多挑战^[17],如众包测试人员的管理^[18]、测试任务的分派、测试技术的管理等。此外,还需对测试结果进行进一步的质量控制,以确保测试人员提交有价值的测试报告,诚信完成测试任务。对于 Web 服务测试,Liu 等^[2]提出了一个基于众包模型的测试系统,在一个目标 Web 服务器上自动执行分布式测试。系统提供了一个可靠的测试模型来模拟真实的用户网络浏览行为,并借助分散在世界各地的网络浏览器。此外,测试工人的可靠性和测试结果的可信度评估^[19]仍然是众包测试中一个亟待解决的问题,它在一定程度上阻碍了众包测试的广泛应用^[11]。Chen 等^[20]对众包测试质量的诸多影响因素进行了分析。众包测试正试图从理论研究转向实际应用。

2.3 众包任务分派

Feng 等^[21]对众包中存在的研究进行了总结梳理,指出众包任务分派问题是当前众包中一个非常重要的问题。目前,众包任务分派是一个以研究为导向的问题。

比较常见的方法是将这类众包测试任务分派问题规约为一个优化问题来处理,将该问题建模为一个约束求解问题,使用精确式算法或启发式算法对其进行优化求解。在文献^[22]中,研究者们提出使用基于贪心算法的启发式策略来实时快速地进行众测任务的分派,同时也在算法中考虑了众测工人和任务的各个相关属性。虽较基于精确式算法的分派方式而言,它的分派效果并不够优秀,但可以保证一定的实时性。Boutsis 等^[23-24]提出了两种实时环境下的众包任务分派方法,分别将任务分派给单个测试工人和一个测试工人组。在复杂众包任务常常无法由单个工人完成的情况下,Rahman 等^[4]考虑到众包工人这种人为因素之间的关联性,提出了一种将任务分派给一组可以相互协作的测试人员的方法,通过提高工人组内和组间的合作程度来提高任务完成的可靠度。

Han 等^[25]认为测试人员的情绪因素会影响其执行任务的效果和效率,他们采用对测试工人进行劳动管理的方式来提高工作效率。Roy 等^[26]提出了知识密集型众包中的任务分派方法。他们考虑测试人员的技能和成本,将任务分派问题规约为一个约束求解问题,并采用了精确式算法、贪心算法等对模型进行优化。Li 等^[27]考虑了工人的信誉因素来提高任务分派的精确性。

iTest 是近几年学者提出的一个 Web 服务众测任务分派模型^[28-29],它假设在网络类型、位置和调用时间上相近的众

测具有相似的运行环境,然后使用贪心算法从不同的相似测试工人集合中选择工人组来获取测试结果。

3 模型定义

3.1 数据模型

在众包测试平台上,有一系列的众包测试任务 $T = \{t_1, t_2, \dots, t_l\}$ 和众包测试工人 $W = \{\omega_1, \omega_2, \dots, \omega_n\}$, l 和 n 分别表示众测任务和工人的数量。在对 Web 服务进行测试时,需求者会将 Web 服务测试任务以众包测试任务的形式发布到众测平台上。在对 Web 服务进行测试时,不同的众测任务会对众测工人在各方面的能力有不同的要求。我们用一个集合 $D = \{d_1, d_2, \dots, d_m\}$ 来表示众测任务可能需要的 m 个技能领域,每个众测工人可能具有其中的某几个能力,而每个众测任务可能会在其中的某几个能力上有要求。例如,对一个 Web 服务进行众包测试会需要相应的众测工人具有以下能力:文字理解能力(d_1)、测试用例设计能力(d_2)、测试报告撰写能力(d_3)以及一定的编程能力(d_4)。组合 Web 服务、众测任务和众测工人的建模如下。

(1)组合 Web 服务(CWS)模型。假定 CWS 工作流中所有路径的集合为 P ,每条路径可用 $P_l = (\omega_{s_1}, \omega_{s_2}, \dots, \omega_{s_n})$ 表示。每个节点 ω_{s_j} 表示路径中的编号为 j 的 Web 服务。对 CWS 进行测试时,将对其中的每个 Web 进行测试作为一个众测任务。

(2)众测任务模型。每个 Web 服务众包测试任务 $t_j \in T$ 都包含以下几个属性: $\langle U_j, C_j, N_j, Exp_j \rangle$ 。 $U_j = \{U_{j1}, U_{j2}, \dots, U_{jm}\}$ 中的每一项 $U_{jk} \in [0, 1]$ 代表对完成任务 t_j 的测试工人在技能领域 k 内的能力值要求; C_j 表示任务 t_j 能接受的最高测试成本; N_j 是要为任务 t_j 分派的众测工人的数量; Exp_j 表示任务 t_j 的任务期限,任务需要在该规定时间段内完成。

(3)众测工人模型。每个众测工人都包含多个属性,因此每个工人 $\omega_i \in W$ 都用一个向量来表示,即 $\langle u_i, c_i, n_i, rep_i \rangle$ 。其中, $u_i = \{u_{i1}, u_{i2}, \dots, u_{im}\}$, 每一项 $u_{ik} \in [0, 1]$ 表示工人 ω_i 在技能领域 k 内的能力值, k 为技能领域的编号, $1 \leq k \leq m$, u_{ik} 的值越高代表工人在该技能上的能力越强,当 ω_i 不具备某个能力时,则该值置为 0; c_i 为众测工人要求的报酬(可以是金钱或其他形式的奖励); n_i 为工人 ω_i 可被分派的最大众测任务数; rep_i 表示工人 ω_i 的信誉值,它反映了工人的可靠程度,与工人过去是否能及时提交测试结果相关。对于新到来平台上的工人,我们一般将其初始信誉值设为 0.5。

众测工人的信誉值在其不断执行任务的过程中是动态变化的,考虑工人每次执行的任务对他们信誉值的影响,在他们完成一个任务后,对其信誉值进行更新。在每次进行任务分派时,将更新的信誉值作为奖励值判断的一部分,来使模型学习选择到信誉值较高的工人。

众测工人是否能在任务给定的期限内完成任务,是衡量工人可靠性的一个重要标准。因为众包环境中的工人一般并非专职工人,他们常会在已经接受了某个任务后,还会处理其他的工作或事务。在 Ipeirotis^[6] 的研究中发现,工人在众包系统中的任务执行时间都服从幂律分布^[30]。利用该规律,可以基于某个众测工人幂律分布的 CDF 及其先前完成任务的

执行时间,来估计工人在规定期限内完成任务的可能性^[24]。

定义 1 令一个变量 x 服从参数为 α 的幂律分布,则其概率密度函数为 $p(k)^{-\alpha}$,定义 $P(k)$ 为幂律分布的 CDF,则有:

$$P(k) = Pr(K \geq k) \quad (1)$$

其中, K 为观测值。进一步地,有:

$$P(k) = \int_k^{\infty} p(k') dk' = (k/k_{\min})^{-\alpha+1} \quad (2)$$

其中, k_{\min} 为幂律行为的最小值。 α 的计算方式如式(3)所示:

$$\alpha = 1 + n \left[\sum_{i=1}^n \ln(k_i / (k_{\min} - 1/2)) \right]^{-1} \quad (3)$$

在任务分配问题中, k_i 表示某个工人 ω_i 对某个任务 t_j 的执行时间 $time_{ij}$, k_{\min} 为工人 ω_i 对 t_j 的执行时间的最小值。概率 $P(time_{ij} \leq Exp_j)$ 即为任务能在 t_j 的任务期限内完成的概率,且可转换为:

$$P(time_{ij} \leq Exp_j) = 1 - P(Exp_j) \quad (4)$$

式(4)中的概率即为工人的信誉值,可反映工人的可靠程度,用于评判工人的好坏。

对于众测平台上工人的信誉值,在每次分派完成后根据其完成任务的概率来更新其信誉值,做到在训练学习过程中根据动态更新的工人信息实时调整分派策略,以更符合众包测试环境的实际需要。

3.2 问题模型

对 Web 服务众测任务分派的目标是使完成测试的效果最好,效果主要包含两个方面:1)针对任务分派的众测工人的能力尽可能高;2)完成任务的成本尽可能少。因此,将每个任务 t_j 的测试效果定义为 v_j ,分派的目标则是使所有任务的测试效果 $V = \sum_{t_j \in T} v_j$ 达到最大化。

对于 v_j ,在假设为任务 t_j 选择的一系列众测工人构成的集合为 $Z_j = \{\omega_1, \omega_2, \dots, \omega_{n_j}\}$ ($Z_j \subseteq W$) 的情况下, Z_j 在各技能领域 k 内能够达到的能力水平值之和为 $skill_{jk} = \sum_{\omega_i \in Z_j} u_{ik}$, Z_j 中的所有众测工人要求的报酬之和为 $cost_j = \sum_{\omega_i \in Z_j} c_i$,其中所有工人的信誉值之和为 $Rep_j = \sum_{\omega_i \in Z_j} rep_i$,则 v_j 可定义为:

$$v_j = B_1 \times \sum_{k=1}^m skill_{jk} + B_2 \times (1 - \frac{cost_j}{C_j}) + B_3 \times Rep_j \quad (5)$$

其中, B_1 , B_2 和 B_3 分别是能力值、成本报酬和信誉值这 3 个方面的权重值,且满足 $B_1 + B_2 + B_3 = 1$ 。通过将任务成本方面的工人报酬值之和 $cost_j$ 进行 $(1 - cost_j / C_j)$ 转换,来使工人能力和报酬这两个矛盾目标的诉求统一化,将对 Web 服务测试任务分派的多个优化目标统一规约为:

$$\text{maximize } V = \sum_{t_j \in T} v_j, Z_j \subseteq W \quad (6)$$

同时,在优化过程中每个任务 $t_j \in T$ 也要服从一系列的约束:

$$\text{约束 I: } skill_{jk} \geq U_{jk}, \forall k \in \{k | 1 \leq k \leq m\};$$

$$\text{约束 II: } cost_j \leq C_j;$$

$$\text{约束 III: } |Z_j| = N_j;$$

$$\text{约束 IV: } \omega Num_i \leq n_i。$$

约束 I 要求选择的众测工人要满足任务在各个技能领域上的能力要求,约束 II 要求所有工人要求索要的报酬不超过任务的最高测试成本。约束 III 规定了对每个任务进行测试的工人数量满足任务的要求。设置约束 IV 是为了避免某些优

秀工人“过劳”,工人 w_i 执行的任务数 $wNum_i$ 要小于或等于其可执行的最大任务数 n_i 。

此外,将每个工人执行任务的效果定义为 p_i ,其计算方式如式(7)所示:

$$p_i = B_1 \times \sum_{k=1}^m u_{jk} + B_2 / c_i + B_3 \times rep_i \quad (7)$$

4 基于强化学习的 Web 服务众测任务分派方法

在 CWS 中,对一个流中的某个 Web 服务进行测试常常依赖于对其前序 Web 服务的测试,在进行 Web 服务众测任务分派时,也要符合 workflow 中的时序要求。而对众测任务分派进行优化的问题,可以建模为一个基于马尔可夫决策过程(Markov Decision Process, MDP)的问题。这类问题适合使用 Q-learning 算法来解决。

在强化学习中,传统 Q-learning 算法具有一定的局限性。例如,Q-learning 使用一个 Q 表来存储状态和动作,在状态、动作较多时,Q 表会变得非常复杂且维度过高,查找和存储都需要消耗大量的时间和内存空间。同时,在众包测试环境中,通常需要将一个测试任务分派给多个众测工人来处理,以使用冗余的方法在一定程度上保证测试质量。在这种情况下,会产生大量的动作,传统 Q-learning 算法显然不适用。而 DQN(Deep Q Network)将 Q-learning 与神经网络结合,对 Q-learning 算法进行了改进,可以有效地解决传统 Q-learning 算法的此类缺陷^[31]。在众包测试环境中,状态和动作的数量较多,因此适宜使用 DQN 进行处理。

本节针对使用 DQN 模型来解决 Web 服务众包测试任务分派问题,提出了一种在线的任务分派方法。在模型中,任务分派智能体(Agent)通过对环境状态的观察来选择合适的众测工人;众测工人将任务的执行效果作为奖励值反馈给 Agent,Agent 根据这些选择和奖励值来学习对测试任务的分派,完成模型的训练过程。本文将提出的基于强化学习的 Web 服务众包测试任务分派模型命名为 WTA-C 模型。

4.1 基于众测的 Web 服务测试任务分派的 MDP 决策过程

马尔可夫决策过程(MDP)是一种典型的智能框架,是建模不确定性下的顺序决策问题。每个 CWS 中的 Web 服务众测任务的分派都可被描述为一个马尔可夫过程,用一个五元组 $\langle S, A, P, R, O \rangle$ 来表示。其中, S 为状态空间,是 Agent 观察到的一组特定的众包状态,为一系列的众包测试任务; $s \in S$, s 为状态空间 S 中的一个任务; A 为动作空间,可执行任务的工人集合,只有在一个工人 w_i 满足任务约束条件的情况下,才会出现在这个集合中,在每个状态 s 下,其可选择的每个工人定义为动作 a , $a \in A(s)$; P 是转移概率, $P(s' | s, a)$ 表示 Agent 在状态 s 下执行动作 $a \in A(s)$ 而转移下一个状态 s' 的概率,且有 $\sum_{s' \in S} P(s' | s, a) = 1, \forall s \in S, \forall a \in A$; R 是众包价值奖励函数, Agent 为状态 s 分派工人,且会得到回报, $r = R(s' | s, a)$; O 是观测空间,是 Agent 可完全观测到的众包测试环境。

MDP 的解是一系列的决策策略,每个策略为一个任务分派方案。本文将这些策略定义为 $\pi, \pi = S \rightarrow A$,其主要思想是在众测环境中,确定能够尽可能达成式(6)中目标的“任务分派策略”,为每个 Web 服务众包测试任务选择

能使其测试效果最好的工人。

4.2 WTA-C 模型

在使用 DQN 来寻找合适的分派策略时,需要先明确其中的各个参数应如何设置以适应 Web 服务众测任务分派的需要。

与马尔可夫决策过程类似,DQN 中一般用 s 表示环境中的某个状态(State),用 a 表示某个动作(Action)。在 WTA-C 方法的 DQN 任务分派环境中,状态 s 为各 Web 服务众测任务,状态空间为众测任务集合; a 为分派 Web 服务众测任务给工人,动作空间 A_i 为所有众测工人。在训练过程中,状态 $\{s_1, s_2, \dots, s_T\}$ 与测试任务 $\{t_1, t_2, \dots, t_T\}$ 中的每个元素一一对应,依次为每个任务选择众测工人。同时,每个工人可以执行的任务个数是有限的。当某个众测工人超过“工作负荷”时,该工人不可以再被分派测试任务。

对一个 CWS 进行测试,即可将对其中每个服务的测试看作一个众测任务 t_j ,这些任务的分派存在时序关系,按照 workflow 中流的顺序执行。

图 1 中,CWS 中的每个节点表示一个 Web 服务,其对应 5 个 Web 服务众测任务,即 $\{t_1, t_2, t_3, t_4, t_5\}$ 。对于该 workflow,因为前后序任务之间有依赖,可以拆分得到两条测试任务的执行路径 $\{t_1, t_2, t_3, t_5\}$ 和 $\{t_1, t_2, t_4, t_5\}$ 。为了保证后序任务都能在前序任务之前分派完成,我们采用广度优先遍历,得到任务分派顺序 $\{t_1, t_2, t_3, t_4, t_5\}$,状态 s 都由前一个任务转移到路径中的下一个任务状态 s' 。分派过程中的状态转换如图 2 所示。

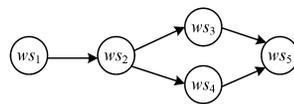


图 1 CWS 示例

Fig. 1 Example of CWS



图 2 任务分派的状态转换

Fig. 2 State transition of task assignment

因为一个任务会分派给多个众测工人,所以需要将同一个任务重复多次执行动作选择的学习,每次获得一个可选测试工人,最终获取一个满足测试要求的工人组。为了避免多次选择同一个工人,我们进行了限定,即在为一个任务选择测试效果最优的工人之后,下一次只能为其选择效果次之的工人,如此依次迭代。当为一个测试任务选择的工人组的各属性满足约束要求时,将不再为其分派工人,继而转入下一个任务。

在选择一组工人的过程中,在选择每个工人时,根据单个工人的能力值、报酬要求和信誉值,根据式(7)给定回报奖励值。在为任务 t_j 选定能达到约束要求的一系列测试工人后,能够获得式(5)中的测试效果值 v_{t_j} 。

每个时间步 t 的 Q 值的更新公式为:

$$Q(s_t, a_t) \leftarrow (1 - \alpha) Q(s_t, a_t) + \alpha [r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})] \quad (8)$$

其中, $Q(s_t, a_t)$ 表示 Agent 在状态 s_t 选择执行动作 a_t 的值函

数; α 为学习速率, $\alpha \in [0, 1]$, α 越大, 对之前训练的效果保留得就越多; r 为当前回报奖励值; s_{t+1} 和 a_{t+1} 为下一个时间步 $t+1$ 的状态和动作, $\max Q(s_{t+1}, a_{t+1})$ 为状态 s_{t+1} 可能产生的最大 Q 值; γ 为折扣因子, 是对未来奖励的衰减值, γ 越大就越重视以往的经验, 越小就越重视当前的回报。

为了弥补传统 Q-learning 的 Q 表的 Q 值记忆方式在存储空间和搜索时长上的缺陷, 在 DQN 中使用值函数近似的方法来得到 Q 值^[32], 不再用 Q 表来存储 Q 值, 而是通过神经网络来预测 Q 值, 并通过不断更新神经网络来进行最优分派策略的学习, 由神经网络学习到一个状态 s 和动作 a 之间的映射关系函数 $f(s, a) = Q(s, a)$, 通过这个函数来拟合 Q 值。DQN 中一般包含两个神经网络, 即当前值神经网络和目标值神经网络, 它们分别实现对当前 Q 值和目标 Q 值的更新。其中, 目标值神经网络是一个参数相对固定的网络, 当前值网络的参数会一直进行更新学习。回放记忆单元 (Replay Memory) 会存储一系列的分派结果, 神经网络以均匀随机采样的方式从回放记忆单元中抽取数据用于参数的更新和学习。

4.3 WTA-C 模型的学习过程

设 θ 表示当前值网络的参数; θ^- 表示目标值网络的参数; $Q(s, a; \theta)$ 表示当前值网络的输出, 用于评估当前“状态-动作对”的值函数; $Q(s_{t+1}, a_{t+1}; \theta^-)$ 表示目标值网络的输出。当前值神经网络和目标值神经网络有同样的结构。在每个时间步 t , 与分派环境交互所得的分配结果会作为一个分派样本存储到回放记忆单元中。基于 DQN 的 WTA-C 方法的学习过程如图 3 所示。

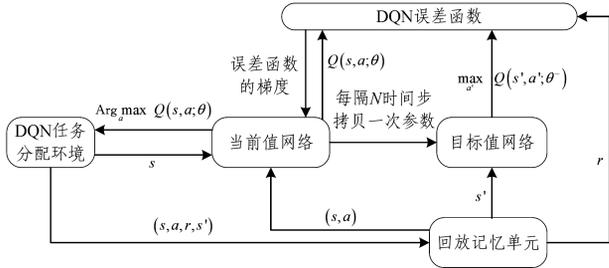


图3 基于 DQN 的 WTA-C 模型的学习过程

Fig.3 Learning process of WTA-C model based on DQN

图 3 中, 首先初始化回放记忆单元、当前值网络中的参数 θ 和目标值网络中的参数 θ^- 。在每个训练回合中, 先初始化状态集合 $\{s_1, s_2, \dots, s_t\}$ 。在每个时间步中, 对于其中的某个状态 s_t , 用 ϵ -greedy 策略来进行动作 a_t 的选择, 即按概率 ϵ 选择一个随机的众测工人, 或按概率 $1-\epsilon$ 从 A_t 中选择一个回报最大的动作 $a_t = \text{Argmax}_{a_t} Q(s_t, a_t; \theta)$, 如式 (9) 所示:

$$a_t = \begin{cases} \text{random}(A_t), & \epsilon \\ \text{Argmax}_{a_t} Q(s_t, a_t; \theta), & 1-\epsilon \end{cases} \quad (9)$$

然后执行动作 a_t , 获得当前回报 r , 将当前的分派结果与下一个状态 (s_t, a_t, r, s_{t+1}) 构成的组合存入回放记忆单元。之后, 从回放记忆单元中随机抽取一个之前的分派结果 (s_i, a_i, r_i, s_i) 进行神经网络的更新。两个神经网络产生的当前值和目标值将用于计算 DQN 的损失函数, 以更新当前值网络的

参数。式 (10) 为损失函数 $L(\theta)$ 的计算方法, 式 (11) 为损失函数梯度计算方法。

$$L(\theta) = E \left[(r + \gamma \max_{a_t} Q(s_{t+1}, a_{t+1}; \theta) - Q(s_t, a_t; \theta))^2 \right] \quad (10)$$

$$\frac{\partial L(\theta)}{\partial \theta} = E \left[(r + \gamma \max_{a_t} Q(s_{t+1}, a_{t+1}; \theta) - Q(s_t, a_t; \theta)) \frac{\partial Q(s_t, a_t; \theta)}{\partial \theta} \right] \quad (11)$$

DQN 的误差函数进行反向传递来学习和更新当前值网络的参数。每隔一定的时间步 (图 3 中标为 N 步), 将当前值网络参数复制给目标值网络。经过一定回合的训练后, 模型趋于收敛且稳定。

5 实验与结果分析

实验使用的仿真 CWS 的工作流来自于 Barabási-Albert 模型^[33], 其包含不同数量的节点。在目前 CWS 的现实应用中, 一般 5~20 个不等的 Web 服务就能够完成 CWS 的功能需求, 考虑到这种现实情况, 我们在实验中使用节点数量分别为 1 到 25 的 25 个工作流, 来获得更具有代表性的实验结果。工作流中的每个节点都对应一个 Web 服务, Web 服务的数据来自于 WS-DREAM¹⁾ 数据集^[34], 每个 Web 服务都有 ID 号进行唯一标识, 且数据集中还包含不同的用户执行特定 Web 的数据, 包括执行时间、是否执行成功等信息。将 WS-DREAM 中的 Web 服务与工作流中的节点随机配对, 构成仿真的 CWS 数据。

每个测试任务 $t_j \in T$ 对应一个 CWS 中的 Web 服务, 在每个技能领域 k 内的能力值要求 U_j 由均值为 15、标准差为 1 的正态分布生成, 然后再乘以一个均值为 50、标准差为 1 的正态分布随机数得到任务的成本要求 C_j ; 要为任务 t_j 分派的工人数量 N_j 为 1 到 100 间的随机数; Exp_j 为任务期限, 在实验中将其设定为任务 t_j 在历史执行中的平均执行时间。

每个众测工人 $w_i \in W$ 的 IP 及其对各个任务的执行信息数据来自于 WS-DREAM 数据集; 在每个技能领域 k 内的能力值 u_{ik} 都由标准正态分布生成; 报酬 c_i 也由标准正态分布生成; 工人 w_i 可被分派的最大众测任务数 n_i 在 1 到 100 之间随机生成。根据 WS-DREAM 中的每个用户执行 Web 服务的历史记录来计算工人信誉值更新的相关参数。

为了验证提出的 WTA-C 方法的有效性和可行性, 本文进行了一系列实验来进行验证和对比。主要评价标准为式 (6) 中的目标函数 $V = \sum_{t_j \in T} v_{t_j}$, 其值越高, 则实验效果越好。

实验 1 WTA-C 方法的训练效果验证实验

图 4 给出了使用 WTA-C 方法对数据训练 2000 次得到的实验结果。其中, 横坐标表示训练的回合次数, 纵坐标表示每回合的累计奖励值。可以看出, 在起初的训练回合内, 累计奖励较低。在 0 到 150 次区间内, 奖励值增长迅速, 处在一个学习过程中。在 200 次左右时, 累计奖励值达到了一个较高的水平, 且在之后的训练回合中都趋于稳定, WTA-C 模型达到了一个稳定收敛的训练效果。

¹⁾ <https://wsdream.github.io>

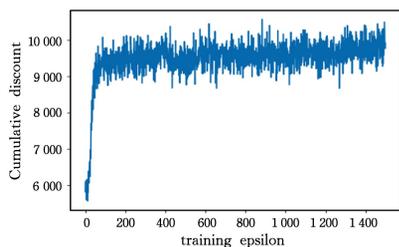


图4 WTA-C方法的训练效果

Fig. 4 Training effect of WTA-C approach

实验1的结果说明了基于深度强化学习的WTA-C方法能够达到一个稳定的训练效果。

实验2 WTA-C方法与其他在线分派方法的对比实验

实验中,基于精确式算法和启发式算法的方法常需要大量的离线计算,耗时过长,不适用于提出的实时任务分派方法进行对比。因此,参考Tung等^[22]提出的测试任务分派方法,本文设置了3个基于贪心算法的贪心策略进行任务分派效果的实验对比。

策略1(H1):仅关注众测工人的能力进行分派,即始终在可选范围内选择能力最高的工人。具体来说,对于测试任务需要的各个技能领域,优先选择这些领域内能力最高的可选测试工人。

策略2(H2):仅关注降低完成测试任务的成本进行分派。对于每个测试任务,始终在可选范围内选择要求报酬最低的众测工人,以降低完成成本。

策略3(H3):同时考虑众测工人的能力、成本和信誉值3方面因素进行分派。在式(7)中,将 B_1 、 B_2 和 B_3 的值都设定为 $\frac{1}{3}$,计算各个测试工人的 p_i 值,优先在可选范围内选择 p_i 值高的工人。

在以上3个策略中,当一个众测工人可执行测试任务数违反约束IV时,其不可再被选择。

对提出的方法WTA-C和提出的4个分派策略进行实验,将它们的目标函数上的实验结果进行对比,结果如图5所示。

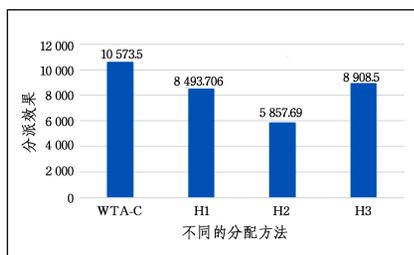


图5 不同任务分派方法的分派效果

Fig. 5 Performance of assignment of different task assignment approaches

图5中,横坐标表示WTA-C、H1、H2和H3这4种不同的任务分派方法,纵坐标表示分派效果(即式(5)中的 v_i),其反映了分派方法在工人质量、信誉值和任务成本这3个方面的分派效果。可以看出,WTA-C模型的分派效果比其他方法更好,比表现最差的H2高出80%,比H1和H3的效果分别高出24%和18%。WTA-C的分派效果比其他方法的表现都更好。

实验3 WTA-C方法与其他分派方法在工人可靠度控制方面的对比实验

众包环境是一个动态变化的环境,工人的信誉值与他们每次执行任务的情况密切相关,且会根据其每次任务执行情况进行更新。工人的信誉值反映了其可靠程度。为了适应环境的变化,我们希望为每个任务选择信誉值更高的工人。

图6为WTA-C方法与实验2中的其他分派方法在分派的工人可靠程度方面的对比结果。为了更直观地进行对比,实验结果采用为每个任务分派的工人的信誉值的均值。

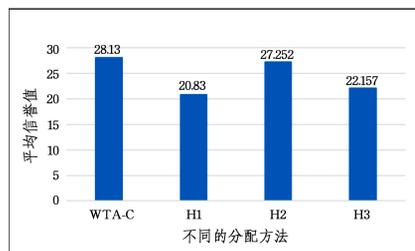


图6 不同任务分派方法在控制工人信誉值上的对比

Fig. 6 Comparison of different task assignment approaches in controlling workers' reputation

图6中,纵坐标是为每个任务分派的工人的信誉值的均值,表示各种方法平均能为各个任务分派何种信誉值的工人。可以看出,H1在控制工人可靠程度方面表现最差,WTA-C比其高出近35%,比H2和H3分别高出3%和27%。另外,考虑了工人信誉值的分配策略H3的表现差强人意,甚至低于仅考虑任务成本的策略H2,这可能与数据集中的数据分布有关。但WTA-C较其他分派方法而言,能为任务分派信誉值更高的工人,同时也可以保证任务被完成的可靠程度更高。

结束语 众包测试环境可利用互联网上零散的劳动力,以及能提供更加广泛的测试环境的特点,成为一种适合对组合Web服务进行测试的新方式。在众包测试中,如何在权衡质量和成本的情况下进行任务分派是一个重要问题。众包测试环境动态变化,且工人的可靠程度在初始情况下是无法预知的,而现有大部分分派方法都使用基于精确式或启发式的方法,计算代价太大且并不能完全适应环境的需要。对组合Web服务的测试是一个基于马尔可夫决策过程的问题。因此,本文提出了基于强化学习的测试任务分派方法,为存在时序关系的Web服务测试任务选择合适的众测工人。

实验结果证明了所提方法的可行性和有效性,但还存在一些局限性和仍未完成的任务:1)在考虑众包环境的动态性时,未将众包工人动态加入和离开的行为考虑在内;2)目前的模型将工人的能力作为已知量进行学习,实际上也应考虑工人的能力值是工人在平台上执行一定数量的任务后学习到的结果;3)我们将尝试在目前比较流行的几个众包平台上进行大规模的实验,来验证所提方法的真实有效性。

参考文献

- [1] HUSSAIN S, WANG Z, TOURE I, et al. Web Service Testing Tools: A Comparative Study [J]. CoRR, 2013, 10(1): 641-647.
- [2] LIU X, HSIEH Y, CHEN R, et al. Distributed testing system for web service based on crowdsourcing [J]. Complexity, 2018:

- 2170585;1-2170585;15.
- [3] LIU D, BIAS R, LEASE M, et al. Crowdsourcing for Usability Testing[C]// Association for Information Science and Technology. 2012;1-10.
- [4] RAHMAN H, ROY S, THIRUMURUGANATHAN S, et al. Task Assignment Optimization in Collaborative Crowdsourcing [C]// IEEE International Conference on Data Mining. 2015: 949-954.
- [5] KOMAROV S, REINECKE K, GAJOS K. Crowdsourcing performance evaluations of user interfaces[C]// Computer Human Interaction. 2013;207-216.
- [6] IPEIROTIS P. Analyzing the Amazon Mechanical Turk Marketplace[J]. CM Crossroads Student Magazine, 2010, 17(2): 16-21.
- [7] TIAN X, LI H, FENG L. Web Service Reliability Test Method Based on Log Analysis[C]// Software Quality, Reliability and Security, 2017;195-199.
- [8] DU Z, MIAO H. Research review on web service composition testing[C]// The International Workshop on Structured Object-oriented Formal Language and Method. 2018;39-51.
- [9] GUO S, LIU Y, CHEN R, et al. Improved SMOTE Algorithm to Deal with Imbalanced Activity Classes in Smart Homes[J]. Neural Processing Letters, 2019, 50(2): 1503-1526.
- [10] GARDLO B. Quality of Experience Evaluation Methodology via Crowdsourcing[D]. Žilina: University of Žilina, 2012.
- [11] GARDLO B, EGGER S, SEUFERT M, et al. Crowdsourcing 2.0: Enhancing execution speed and reliability of web-based QoE testing[C]// IEEE International Conference on Communications. 2014;1070-1075.
- [12] BLANCO R, HALPIN H, HERZIG D, et al. Repeatable and reliable search system evaluation using crowdsourcing [C]// International Conference on Research and Development in Information Retrieval. 2011;923-932.
- [13] SHERIEF N, JIANG N, HOSSEINI M, et al. Crowdsourcing software evaluation[C]// Evaluation & Assessment in Software Engineering. 2014;19:1-19;4.
- [14] CHEN F, KIM S. Crowd debugging, in Joint Meeting on Foundations of Software Engineering [C]// ESEC/SIGSOFT FSE. 2015;320-332.
- [15] PETRILLO F, SOH Z, KHOMH F, et al. Towards Understanding Interactive Debugging [C]// IEEE International Conference on Software Quality. 2016;152-163.
- [16] PETRILLO F, LACERDA G, et al. Visualizing interactive and shared debugging sessions [C]// IEEE Working Conference on Software Visualization. 2015;140-144.
- [17] GUAIANI F, MUCCINI H. Crowd and Laboratory Testing, Can They Co-exist? An Exploratory Study[C]// International Conference on Software Engineering. 2015;32-37.
- [18] LIU Y, ZHANG T, LI K, et al. Evaluation model of mobile application crowdsourcing testers [J]. Journal of Computer Applications, 2017, 37(12): 3569-3573.
- [19] ZHANG Z Q, PANG J S, XIE X Q, et al. Research on Crowdsourcing Quality Control Strategies and Evaluation Algorithm [J]. Chinese Journal of Computers, 2013, 36(8): 1636-1649.
- [20] CHENG J, GE L Q, ZHANG T, et al. Research on factors affecting quality of mobile application crowdsourced testing [J]. Journal of Computer Applications, 2018, 38(9): 2626-2630.
- [21] FENG J H, LI G L, GENG J H. A Survey on Crowdsourcing [J]. Chinese Journal of Computers, 2015, 38(9): 1713-1726.
- [22] TUNG Y, TSENGE S. A novel approach to collaborative testing in a crowdsourcing environment [J]. Journal of Systems and Software, 2013, 86(8): 2143-2153.
- [23] BOUTSIS I, KALOGERAKI V. On Task Assignment for Real-Time Reliable Crowdsourcing [C]// IEEE International Conference on Distributed Computing Systems. 2014;1-10.
- [24] BOUTSIS I, KALOGERAKI V. Crowdsourcing under Real-Time Constraints [C]// International Parallel (and Distributed) Processing Symposium. 2013;753-764.
- [25] HAN Y, SHEN Z, FAUVEL S, et al. Efficient scheduling in crowdsourcing based on workers' mood[C]// IEEE International Conference on Agents. 2017;121-126.
- [26] ROY S, LYKOURENTZOU I, THIRUMURUGANATHAN S, et al. Task assignment optimization in knowledge-intensive crowdsourcing [J]. The VLDB Journal, 2014, 24(4): 467-491.
- [27] LI Y J, GUO J F, GOU X M. Software Task Allocation Method in Crowdsourcing [J]. Computer Systems & Applications, 2015, 24(2): 1-6.
- [28] YAN M, SUN H, LIU X. iTest: testing software with mobile crowdsourcing[C]// CrowdSoft@SIGSOFT FSE. 2014;19-24.
- [29] YAN M, SUN H, LIU X. Efficient Testing of Web Services with Mobile Crowdsourcing[C]// Internetware. 2015;157-165.
- [30] CLAUSET A, SHALIZI C, NEWMAN M. Power-law distributions in empirical data[J]. SIAM Review, 2009, 51(4): 661-703.
- [31] RUMMERY G, NIRANJAN M. On-line Q-learning using connectionist systems [D]. University of Cambridge, 1994.
- [32] TIAN Y L, HUI L. Accelerated modify approach for initial state error iterative learning control in sense of Lebesgue-p norm[J]. Control and Decision, 2016, 31(3): 429-434.
- [33] BARABÁSI A, ALBERT R. Emergence of scaling in random networks[J]. Science, 1999, 286(5439): 509-512.
- [34] ZHENG Z, LYU M. Collaborative reliability prediction of service-oriented systems[C]// ACM/IEEE International Conference on Software Engineering. 2010;35-44.



TANG Wen-jun, born in 1994, Ph. D. Her research interests include crowdsourcing workflows, crowdsourcing task assignment and web service testing.



CHEN Rong, born in 1969, Ph. D, professor, is a member of the IEEE and a member of the ACM. His research interests include software diagnosis, collective intelligence, activity recognition, Internet and mobile computing.