Engineering Applications of Artificial Intelligence 74 (2018) 212–225

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Task assignment in microtask crowdsourcing platforms using learning automata

Alireza Moayedikia [a,*], Kok-Leong Ong [b], Yee Ling Boo [c], William G.S. Yeoh [a]

[a] Department of Information Systems and Business Analytics, Deakin University, Victoria 3125, Australia
[b] SAS Analytics Innovation Lab, ASSC, La Trobe University, Victoria 3086, Australia
[c] School of Business IT and Logistics, RMIT University, Victoria 3000, Australia

ARTICLE INFO

Keywords:
Crowdsourcing
Microtasking
Learning automata
Reinforcement learning
Task assignment

ABSTRACT

Conventional microtask crowdsourcing platforms rely on a random task distribution strategy and repeatedly assign tasks to workers. This strategy known as repeated labelling suffers from two shortcomings of high cost and low accuracy as a result of making random distributions. To overcome such shortcomings researchers have introduced task assignment as a substitute strategy. In this strategy, an algorithm selectively chooses suitable tasks for an online worker. Hence, task assignment has gained attentions from researchers to reduce the cost of microtasking whiling increasing its accuracy. However, the existing algorithms on task assignment suffer from four shortcomings as: (i) human intervention, (ii) reliance on a rough estimation of ground truth, (iii) reliance on workers' dynamic capabilities and (iv) lack of ability in dealing with sparsity. To overcome these shortcomings this paper proposes a new task assignment algorithm known as LEarning Automata based Task assignment (LEATask), that works based on the similarities of workers in performance. This algorithm has two stages of exploration and exploitation. In exploration stage, first a number of workers are hired to learn their reliability. Then, LEATask clusters the hired workers using a given clustering algorithm, and for each cluster generates learning automata. Later, the clusters of workers along with their attached learning automata will be used in exploitation stage. Exploitation stage initially assigns a number of tasks to a newly arrived worker to learn the worker's reliability. Then, LEATask identifies the cluster of worker. Based on the cluster that worker resides in and the attached learning automata, the next tasks will be assigned to the new worker. LEATask has been empirically evaluated using several real datasets and compared against the baseline and novel algorithms, in terms of root mean square error. The comparisons indicates LEATask consistently is showing better or comparable performance.

1. Introduction

Microtasking is the process of assigning tasks to random crowd of people. A typical strategy of microtasking is to assign the tasks repeatedly and on a random basis to several workers. Finally majority voting will aggregate the collected answers to determine the ground truth for each task (Sheng et al., 2008; Geiger and Schader, 2014; Chiu et al., 2014; Nevo and Kotlarsky, 2014). This strategy is known as repeated labelling, in which every assignment comes with an unconditional but certain amount of payment to incentivise workers in completing the tasks (Goncalves et al., 2015).

Repeated labelling has two major shortcomings (Amirkhani and Rahmati, 2014). The first issue is untargeted assignment. This means that all the tasks are assigned to workers without measuring their suitability for a given task. The second issue is untargeted aggregation due to applying majority voting answer aggregation technique. This means that, all the answers collected from workers are aggregated again without measuring their appropriateness. To overcome the issues of repeated labelling n microtask crowdsourcing platforms solutions are proposed that selectively match workers with proper tasks.

These algorithms are known as task assignment. Even though the existing task assignment algorithms have proven to be a proper substitute for repeated labelling, but they mainly suffer from four shortcomings, including:

- **Human intervention** indicates that the process of task assignment needs a human to monitor the process. As examples of such limitation are the algorithms proposed by Ho and Vaughan (2012) and Ho et al. (2013). These algorithms assume that a task

* Correspondence to: 70 Elgar Road, Burwood, VIC 3125, Australia.
E-mail address: amoayedi@deakin.edu.au (A. Moayedikia).

https://doi.org/10.1016/j.engappai.2018.06.008
Received 12 June 2017; Received in revised form 15 April 2018; Accepted 25 June 2018

master (i.e., someone who distributes the tasks) measures the benefit of workers while they are being assigned to tasks and the algorithms use the benefit amount to make further assignments. However, in reality workers and tasks are numerous and heterogeneous in the sense that they are not only large in amount but also very diverse in expected expertise and knowledge. Hence, relying on human intervention to guide the process of task assignment limits the utility of such task assignment algorithms.

- **Availability of ground truth** indicates that task assignment algorithm requires an estimation of ground truth to measure how good workers are solving the tasks. Algorithms that rely on grand truth availability such as Khattak and Salleb-Aouissi (2011) and Pfeiffer et al. (2012), the rough estimation dependants on workers answers. If answers are misleading and very different from the actual ground truth, then the rough estimation and therefore the entire process of task assignment will be misleading. The algorithm require expert-generated pairs of task-answer to check the performance of every worker against workers' answers.
- **Reliance on workers dynamic capabilities** means that the execution of an algorithm depends on intrinsic capabilities of workers such as, how fast a worker solves a task (i.e., speed) and how many tasks workers can solve (i.e., capacity). However, in reality the capacity and the speed a worker can solve a task vary. For instance, Boutsis and Kalogeraki (2014) proposed an algorithm that asks workers to solve a number of tasks within a timeframe. Workers who do not meet the timeframe condition are not reliable and should not be considered. In this algorithm the timeframe is specified regardless of the capability of workers. In another algorithm proposed by Ho et al. (2013) workers are asked to provide the number of task they can solve, while in reality this question depends on type of task and how difficult the tasks are.
- **No sparseness** is an assumption that most of the task assignments are based on. In this assumption all workers complete the assigned microtasks (Boutsis and Kalogeraki, 2014; Ho and Vaughan, 2012; Ho et al., 2013; Tran-Thanh et al., 2014). Non-sparsity is a common setting considered in the conventional task assignment algorithms. For example, the algorithms introduced by Karger et al. (2011) assume that workers answer all the tasks and based on that workers can be divided into two categories of hammers and spammers. However in reality workers are allowed to skip answering the assigned microtasks.

To overcome these shortcomings this paper introduces a novel task assignment algorithm called LEarning Automata based Task assignment (LEATask). The algorithm has two stages of exploration and exploitation. In exploration stage a number of workers are hired to learn their reliability. The learning process is accomplished by assigning a subset of tasks to the hired workers. Then, these workers will be clustered using a given clustering algorithm to group the ones who are similar in terms of performance (i.e., reliability). Then, LEATask associates each cluster with a Learning Automata (LA) and trains every automaton using the data resides in that cluster.

Exploitation stage uses the clusters and their associated learning automata to assign tasks to workers. This stage, initially assigns a number of tasks (known as sliding task) to a newly arrived worker, then estimates the reliability of workers on each task and consequently identifies the cluster of the new worker. Based on the cluster that worker resides in, and its associated learning automata the next task will be assigned. Specifically, LEATask makes the following contributions:

- **Unsupervised task assignment**: LEATask relaxes the requirement of human intervention by integrating reliability rate of workers with task assignments. Workers who seem to be unreliable will no longer be assigned to any further tasks.
- **No reliance on ground truth:** LEATask in two components of task assignment and worker reliability estimation, does not require a rough estimation of ground truth, instead, it formulates the process of task assignment as a two-stage algorithm where in the first stage it learns the reliability of workers and then in exploitation uses the reliability information to decide on task assignment.
- **No reliance on workers' dynamic capabilities:** LEATask relies on workers' reliability rather than dynamic characteristic of workers that might vary from worker to worker or might change over time (e.g., mathematical skills) a reliability estimation algorithm that measures suitability of a worker for a given task and then inputs the reliability to LEATask is much able to help LEATask with task assignment decision.
- **Robustness to sparseness:** LEATask is not affected by sparsity of worker-task matrix due to reliance on learning automata and clusters of workers. In cases that worker-task matrix is sparse and the process cannot recognise the next suitable task then LEATask relies on clusters of workers to predict the next proper assignment.

The rest of the paper is organised as follows. To have a better understanding about task assignment in crowdsourcing platforms Section 2 presents some of the related works on task assignment. The related works are reviewed based on their limitations they impose on task assignment. We introduce the problem formulation and the LEATask algorithm in Sections 3 and 4, respectively. Section 5 conducts some experiments and comparisons to highlight the strengths and identify weakness of LEATask relative to other novel and baseline algorithms and finally Section 6 concludes the paper.

2. Literature review

Matching workers with tasks can be done in two ways of either worker to tasks or task to workers, as showed by Moayedikia et al. (2017). The former known as worker selection in which a worker is assigned to tasks, while latter known as task assignment in which a task is assigned to a subset of workers. The focus of this paper is on task assignment and hence introduces a task assignment algorithm that resolve the shortcomings exist in some of the current task assignment algorithms.

Some of the recent algorithms are supervised approaches, which require human intervention. The algorithm proposed by Ho and Vaughan (2012) named Dual Task Assigner (DTA) uses a manual approach to collect information regarding workers. Similar to LEATask (proposed in this paper), DTA assumed that workers arrive sequentially and then they can be queried for their skills and competency level of their skills.

Then, based on the collected information and in a setting similar to AdWords, DTA assigns tasks to workers. In AdWords the problem is finding a proper advertisement for a user, while in task assignment the problem is to find proper task for worker. Algorithms proposed by Khattak and Salleb-Aouissi (2011) and Pfeiffer et al. (2012) also require human intervention to learn about workers' expertise. This is done through injecting some expert-labelled tasks to the pool of tasks. Based on the answers received from workers for those expert-generated tasks, workers' expertise will be estimated.

The other shortcoming that limits the utilities of the existing algorithms is reliance on dynamic humans' capabilities such the time it takes for someone to solve a task (Boutsis and Kalogeraki, 2014) or number of tasks a person can solve (Ho et al., 2013). The algorithm proposed by Boutsis and Kalogeraki (2014) expects workers to solve the tasks within a given timeframe.

The algorithm learns workers' expertise by assigning a task to them and then requiring them to solve the task within a given timeframe. If a worker solves a task within a given timeframe, then the answer collected from worker is not recognised as proper for aggregation. However, the