A Markov Decision Process Framework for Predictable Job Completion Times on Crowdsourcing Platforms*

Chandrashekar Lakshminarayanan, Ayush Dubey, Shalabh Bhatnagar
{chandrul,ayush.dubey,shalabh}@csa.iisc.ernet.in
Department of Computer Science and Automation,
Indian Institute of Science, Bangalore - 560 012, India.

Chithralekha Balamurugan
Chithralekha.Balamurugan@xerox.com
Xerox Research Centre India,
Bangalore -560 103, India.

Task starvation leads to huge variation in the completion times of the tasks posted on to the crowd. The price offered to a given task together with the dynamics of the crowd at the time of posting affect its completion time. Large organizations/requesters who frequent the crowd at regular intervals in order to get their tasks done desire predictability in completion times of the tasks. Thus, such requesters have to take into account the crowd dynamics at the time of posting the tasks and price them accordingly. In this work, we study an instance of the pricing problem and propose a solution based on the framework of Markov Decision Processes (MDPs). In the what follows we describe the problem and the proposed solution in brief.

Problem Setting

We consider the problem of posting CAPTCHA tasks on to the crowd at regular intervals over a period of 24 hours. The n^{th} CAPTCHA task is posted at time instant t_n and the successive tasks are separated by $\mathcal{T} = 12$ mins (i.e., $t_{n+1} - t_n = \mathcal{T} = 12$ mins). Each CAPTCHA task consists of 120 CAPTCHAs, and the requester has three pricing options namely, payment of 6 cents, 4 cents or 3 cents for the completion of 120 CAPTCHAs. We make use of the below notation:

- a_n is the price offered on completion of the n^{th} task.
- $u = \{a_1, \ldots, a_n, \ldots\}$ is the pricing policy or pricing scheme.
- c_n^1, c_n^2 and c_n^3 denote the completion times of tasks posted at t_n that offer 6 cents, 4 cents and 3 cents respectively for completion of the n^{th} CAPTCHA task.

The requester wants to reduce the variance in the completion times and at the same time needs to minimize the total amount paid.

Crowd Behavior MDP (CBMDP)

We formulate the above problem in the framework of Markov Decision Processes (MDPs), that we call the 'Crowd Behavior' MDP whose

- **State space** is denoted by $S = \{1, \ldots, k\}$, and $s_n \in S$ is the state that reflects the crowd behavior at t_n. It is important to note that the actual state of the crowd is a hidden variable and not amenable to direct measurement. Thus, we need to choose an observable quantity as the state. The completion times of the recently posted tasks is observable and contains information that indicates how much time it might take for the current task to get completed. In this work, we let the time taken to complete task that offered 4 cents for 120 CAPTCHAs be an indirect measurement of the state of the crowd[1]. Formally, we define the state as

$$s_n = \lceil \frac{k}{c_{\max}} \frac{1}{m} \sum_{i=0}^{m-1} c^2_{n-l-i} \rceil, \tag{1}$$

where $l > 0$, $m > 0$, $k > 0$ are positive integers and $c_{\max}$ is the maximum completion time. The tasks posted in the immediate past (i.e., tasks $n-1, \ldots, n-l+1$) might not have completed at t_n and as a result we cannot calculate their completion times. Hence we instead use the completion times of tasks $n-l-m+1$ to $n-l$. The interval $[0, c_{\max}]$ is partitioned into k bins, and the state at n^{th} instant is the bin corresponding the average completion times of tasks $n-l-m+1$ to $n-l$.
- **Action Space** is denoted by A, and $A = \{a(1) = 6, a(2) = 4, a(3) = 3\}$ cents.
- **Model Parameters** constitute the cost of action $a \in A$ in state $s \in S$ denoted by $g = (g_a(s), \forall s \in S, a \in A)$, and the probability of transitioning from state $s \in S$ to $s' \in S$ denoted by $P = (p_a(s, s'), \forall s, s \in S)$. Both g and P are learnt from data collected via experiments conducted on the crowd.
- **Cost Criterion** is the infinite horizon discounted cost which is defined for a (pricing) policy u to as below:

$$J_u(s) = \mathbf{E}[\sum_{n=0}^{\infty} \alpha^n g_{u_n}(s_n) | s_0 = s, u],$$

where $0 < \alpha < 1$ is a given discount factor.

Our aim is to compute the optimal pricing policy $u^* = \arg\max_{u \in U} J_u(s), \forall s \in S$, where U is the class of all pricing policies.

*This work was supported by Xerox Corporation USA.

[1] The task that pays 4 cents for 120 CAPTCHAs serves as the unit task.

s	Average completion of previous tasks in mins	$u^*(s)$ (Price offered in cents)
1	0-10	3
2	10-20	3
3	20-30	3
4	30-40	3
5	40-50	6
6	50-60	6
7	60-70	4
8	70-80	4
9	80-90	6

Table 1: Optimal policy u^* obtained by solving CBMDP.

Experiments and Model learning

Tasks were posted in CrowdFlower every $\mathcal{T} = 12$ mins for a period of 48Hrs. At time $t_n = n\mathcal{T}$, 3 copies of the n^{th} task with three different prices namely $a(1) = 6$, $a(2) = 4$ and $a(3) = 3$ cents were posted simultaneously and their respective completion times $c_n^i, i = 1, 2, 3$ were observed. We fit linear model for c_n^1 and c_n^3 in terms of c_n^2, and we learnt that $c_n^1 \approx \frac{33}{59}$, and $c_n^3 \approx \frac{82}{59}$.

We set $m = 5$, $l = 5$ and $k = 9$ while calculating s_n defined in (1). Since the actions only affect the current completion time and not future crowd behavior, the probability transition is the same for all the actions, and is computed from data as

$$P\{s_{n+1} = s' | s_n = s\} = \frac{\left|\{n | s_{n+1} = s', s_n = s\}\right|}{\mathcal{N}}, \quad (2)$$

where $\mathcal{N}$ is the total number of data points available (in our case we collected data for 48 Hrs, hence $\mathcal{N} = 48 \times 5 = 240$). In (2) $|B|$ denotes the cardinality of the set B. The cost $g_a(s)$ corresponding to action a in state s is $g_a(s) = w_1 \times a + w_2 \times T_a(s)$, for positive constants $w_1, w_2 > 0$ and

$$T_a(s) = \mu(\mathcal{I}_a) + 2 \times \sigma(\mathcal{I}_a), \text{where} \quad (3)$$

$$\mathcal{I}_a = \{c_i^a | s_i = s\}. \quad (4)$$

In (3), $\mu(\mathcal{I}_a)$ denotes the mean of the elements of $\mathcal{I}_a$, and $\sigma(\mathcal{I}_a)$ is the standard deviation of the same. Due to the presence of the σ term in (3), $T_a(s)$ (left plot of Figure 1) in a way captures the worst case completion time possible in state s as compared to the average scenario. Note that $g_a(s)$ is a linear combination of price offered and the worst case completion time.

Computing the optimal policy

We set $w_1 = 0.6$, $w_2 = 0.4/20$ (i.e., we scale down the completion times by a factor of 20 so that it is in scale with the price values that are in the range of 3 to 6 cents). We chose the discount factor α to be 0.99. We solve the CBMDP using policy iteration(D.P.Bertsekas 2007). Table 1 shows the optimal policy u^* learnt by solving the CBMDP. We compared the performance of the optimal policy u^* and a baseline policy u^2, which is a constant pricing policy $u^2 = \{a(2), \ldots, a(2)\}$, i.e., u^2 always pays 4 cents for 120 CAPTCHAs. We evaluated u^* and u^2 on four separate periods of 24 Hrs, each spread across the working days of the week. The result is presented in the right plot of Figure 1. We compare u^2 and u^* by deploying them simultaneously, i.e., two separate copies of the n^{th} task are posted at t_n, one priced at a constant value of $a(2) = 4$ cents, and the other is priced as $u^*(s_n)$. In the four experiments we thus conducted, the total price spent by u^* over 120 tasks was less than 460 cents (lower than 480 cents offered by u^2 over the same period). The mean and standard deviation of the completion times achieved by the baseline policy u^2 were 58 mins and 29 mins respectively, and the mean and standard deviation of the completions times achieved by the optimal policy u^* were 54 mins and 21 mins respectively. This indicates a clear improvement in performance when using the optimal pricing policy obtained upon solving the CBMDP.

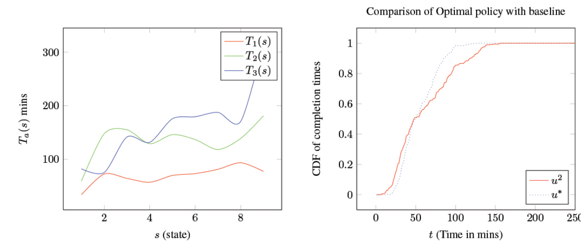

Figure 1: Left plot shows $T_a(s)$ which captures the worst case completion time in each state. The right plot shows the performance of the optimal policy u^* and the baseline policy u^2.

Other important works making use of MDP formulation in crowd setting are (Peng, Mausam, and Weld 2010; Kamar, Hacker, and Horvitz 2012). Optimal pricing problem is also addressed in (Faradani, Hartmann, and Ipeirotis 2011).

Conclusion

We suggested a novel methodology to address the problem of predictable completion time of tasks posted on to the crowd based on the framework of Markov Decision Processes. We presented a full cycle that included data collection, followed by learning of model parameters, computing the optimal policy and testing them in a real world scenario. It was found that the optimal policy achieves better predictability in completion times than the baseline policy while incurring lesser total expenditure.

References

D.P.Bertsekas. 2007. Dynamic Programming and Optimal Control, Vol. II. *Athena Scientific, Bellmont*.

Faradani, S.; Hartmann, B.; and Ipeirotis, P. G. 2011. What's the right price? pricing tasks for finishing on time. In *Human Computation*.

Kamar, E.; Hacker, S.; and Horvitz, E. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 467–474.

Peng, D.; Mausam, D.; and Weld, S. 2010. Decision-theoretic control of crowd-sourced workflows. In *In the 24th AAAI Conference on Artificial Intelligence (AAAI10*. Citeseer.