Q-learning for the Optimal Bernoulli Routing

Model Description

We consider a system with N parallel queues and a single dispatcher (or router). Jobs arrive to the system according to a Poisson process of rate λ. We assume that the service time of jobs in the queues is exponentially distributed and we denote by r_i the rate at which jobs at queue i are served. When a job arrives to the dispatcher it is immediately routed to Queue i with probability p_i. Hence, $\sum_{i=1}^{N} p_i = 1$. We aim to study the value of the routing probabilities such that the mean number of customers is minimized.

The authors in [2] study this system and characterize the optimal routing probability. Here, aim to show that Q-learning can be used to learn which is the optimal routing probability.

Markov Decision Process Formulation

We formulate the above problem as a Markov Decision Process in discrete time in which the discretization is carried out when a job arrives to the system. We consider the discounted cost problem, that is, we aim to find the probabilities $p_1, p_2, \dots, p_N$ such that the following expression is minimized:

$$\mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\delta^t Q_i(t))\right], \tag{1}$$

where $\delta \in (0,1)$ and $Q_i(t)$ is the number of jobs in Queue i at time slot t.

Let us define the following elements of the Markov Decision Process we consider:

- The state represents the number of jobs in each queue. Therefore, the set of states is a vector of size N such that each element is 0 or a natural number. That is, $\mathcal{S} = \mathbb{N}_0 \times \cdots \times \mathbb{N}_0$, where $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

- The action is a vector $(p_1, \dots, p_N)$ such that the i-th element is the probability that an incoming job is sent to Queue i. Therefore, the set of actions is a probability vector of size N. We assume that each element of the probability vector belongs to $\{0, \frac{1}{d}, \frac{2}{d}, \dots, \frac{d-1}{d}, 1, \}$ for a fixed d (this means that the probabilities will never be real values).

- The cost is the total number of customers in the system when a job is sent to one of the queues.

- The transition probabilities. Between two arrivals, one or more jobs can be served at Queue i. We denote by $q_{i,j}$ the probability that j jobs are served at Queue i in a interval of time of λ.

Calculation of the transition probabilities

The probabilities for a $G/M/1$ queue are calculated in [1]. In our case:

$$q_{i,j} = \begin{cases} \int_0^\infty \frac{(r_i t)^j}{j!} e^{-r_i t} \lambda e^{-\lambda t} dt & j < Q_i \\ 1 - \sum_{k=0}^{Q_i - 1} q_{i,k} & j = Q_i \end{cases}$$

Whre Q_i is the number of jobs at queue i. Thus,

$$q_{i,0} = \int_0^\infty \lambda e^{-t(r_i+\lambda)} dt = \frac{\lambda}{r_i + \lambda}$$

When $0 < j < Q_i$, integrating by parts we get:

$$\int_0^\infty \frac{(r_i t)^j}{j!} e^{-r_i t} \lambda e^{-\lambda t} dt = \frac{r_i^j}{j!} \lambda \int_0^\infty t^j e^{-t(r_i+\lambda)} dt \underset{\uparrow}{=} \frac{r_i^j}{j!} \lambda \left(\left. \frac{-t^j e^{-t(r_i+\lambda)}}{r_i + \lambda} \right|_0^\infty + \int_0^\infty j t^{j-1} \frac{e^{-t(r_i+\lambda)}}{r_i+\lambda} dt \right) =$$

$$\boxed{\begin{array}{cc} u = t^j & dv = e^{-t(r_i+\lambda)} \\ du = j t^{j-1} & v = \frac{-e^{-t(r_i+\lambda)}}{r_i+\lambda} dt \end{array}}$$

$$= \frac{r_i^j}{j!} \lambda \int_0^\infty j t^{j-1} \frac{e^{-t(r_i+\lambda)}}{r_i+\lambda} dt = \frac{r_i}{r_i+\lambda} \int_0^\infty \frac{(r_i t)^{j-1}}{(j-1)!} \lambda e^{-t(r_i+\lambda)} dt = \frac{r_i}{r_i+\lambda} q_{i,j-1} =$$

$$= \left(\frac{r_i}{r_i+\lambda} \right)^j q_{i,0} = \frac{r_i^j \lambda}{(r_i+\lambda)^{j+1}}$$

When $j = Q_i$:

$$q_{i,j} = 1 - \sum_{k=0}^{Q_i-1} q_{i,k} = 1 - \sum_{k=0}^{Q_i-1} \frac{r_i^k \lambda}{(r_i+\lambda)^{k+1}} = 1 - \frac{\lambda}{r_i+\lambda} \sum_{k=0}^{Q_i-1} \left(\frac{r_i}{r_i+\lambda} \right)^k = 1 - \frac{\lambda}{r_i+\lambda} \left(\frac{1 - \left(\frac{r_i}{r_i+\lambda}\right)^{Q_i}}{1 - \left(\frac{r_i}{r_i+\lambda}\right)} \right) =$$

$$= 1 - \frac{\lambda}{r_i+\lambda} \left(\frac{1 - \left(\frac{r_i}{r_i+\lambda}\right)^{Q_i}}{\frac{\lambda}{r_i+\lambda}} \right) = \left(\frac{r_i}{r_i+\lambda} \right)^{Q_i}$$

Therefore,

$$q_{i,j} = \begin{cases} \frac{r_i^j \lambda}{(r_i+\lambda)^{j+1}} & j < Q_i \\ \left(\frac{r_i}{r_i+\lambda} \right)^j & j = Q_i \end{cases}$$

Size of the action space

Let a_d be the size of the action space. Then, a_d is the number of ways N numbers from $\{0, \frac{1}{d}, \frac{2}{d}, \ldots, \frac{d-1}{d}, 1, \}$ can add to 1, i.e., the number of ways N numbers from $\mathbb{N}_0$ can add to d. By considering the generating function of the sequence $\{a_i\}_{i \in \mathbb{N}_0}$ we get that $a_d = \binom{N+d-1}{d}$.

$$\sum_{k=0}^\infty a_k x^k = (1 + x + x^2 + x^3 + \cdots)^N = (1-x)^{-N} = \sum_{k=0}^\infty \binom{N+k-1}{k} x^k$$

References

[1] I. Adan and J. Resing. Queueing theory, 2002.

[2] E. Altman, U. Ayesta, and B. Prabhu. Load Balancing in Processor Sharing Systems. *Telecommunication Systems*, 47(1-2):pp.35–48, May 2011.