Division of labour in communicating multi-armed bandits

Bhaskar Kumawat, Preetham Venkatesh

February 15, 2019

1 SETUP

$$\mathbb{N} = \{1, 2, 3, ..., N\} \rightarrow \text{Set of players}$$
$$\mathbb{K} = \{1, 2, 3, ..., K\} \rightarrow \text{Set of arms}$$

In all subsequent discussion, n and k will be used to refer to any member of sets $\mathbb{N}$ and $\mathbb{K}$ respectively.

$$M_n = \text{Set of neighbours of player n}$$

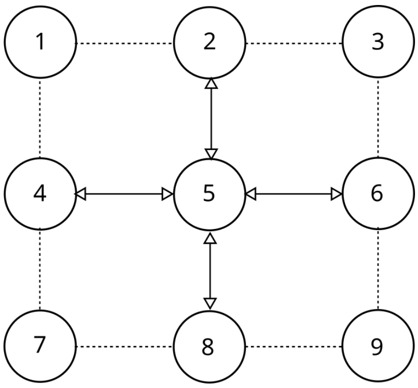

Figure 1: A network of multi armed bandits with information sharing

For example, in Figure 1,

$$M_5 = \{2, 4, 6, 8\}$$

Also,
$\eta =$ computational power available to each player

In this setup, the overall information with a player at the end of a timestep can be traded for utility in a non-linear fashion.

2 AGENTS

Each player has the following functions associated with it at a time-step t,

- $S_n(k)$ = Distribution of number of times arm k is pulled by player n at a time t

$$S_n : \mathbb{K} \rightarrow \mathbb{Z}$$

$$\sum_{k=1}^{K} S_n(k) \leq \eta$$

Number of arms that can be pulled at a time step is limited by computational power of the agent.

- $\psi_n(k)$ = Number of k-arm pulls from which information is released to neighbours.

$$\psi_n(k) \leq S_n(k), \forall k \in \mathbb{K}, \forall n \in \mathbb{N}$$

Example: If $S_n(1) = 4$, the player n has pulled arm one 4 times in the time step. Now he may choose to release any amount of information obtained from arm one being pulled between 0 to 4 times.

- $\rho_n(k)$ = Number of k-arm pulls about which information is recieved from neigbours.

$$\rho_n(k) = \frac{\sum_{j \in M_n} \psi_j(k)}{4}$$

The information about pulls gets divided equally among four neighbours of a player and hence the effective number of pulls is 1/4.

3 PROCESS

At each time step, the players do the following (in the same order).

1. Pull a number of arms to specify $S_n(k)$ for all k. Making sure that $\sum_k S_n(k) \leq \eta$.

2. Specify the send function ψ_n such that $\psi_n(k) \leq S_n(k)$ for all k.
3. Receive information about $k-$pulls from neighbours to specify ρ_n.
4. calculate total utility obtained over the time-step.

4 UTILITY

The utility function is a non-linear function of the information available to a player at the end of the time-step.

Define: Grand information function $I_n(k)$

$$I_n(k) = S_n(k) - \psi_n(k) + \rho_n(k)$$

The total information for an arm k for a player n is simply the sum of the information generated by pulling the arm (S_n) and the information received from neighbours (ρ_n), minus the information released to the neighbours (ψ_n).

A mechanism Ω is a series of utility functions that define the utility of associated with each arm k given it is played as a part of a grand information function I_n

$$\Omega =< U_1(I), U_2(I), U_3(I), ..., U_(K) >$$

Where utility of each player is given by,

$$\text{Utility of player n } = \sum_k U_k(I_n)$$

The goal is to design a mechanism that leads to division of labour among the players, with certain players specialising in pulling of a limited set of arms and sharing the information to obtain an overall benefit.

In retrospect, we will like to see the kinds of division of labour that can take place given the different ways the utility functions translate the information set into utility.