Sequential Resource Allocation Under Uncertainty: An Index Policy Approach

Weici Hu
Adviser: Peter Frazier

Cornell ORIE

July 6, 2017

Problem Setup

We consider an MDP $(\mathbb{S}^K, \mathbb{A}^K, \mathbb{P}^\cdot, R)$ that consists of K identical sub-processes $(\mathbb{S}, \mathbb{A}, P^\cdot, r)$, specifically,

- Time horizon $T < \infty$.
- State space $\mathbb{S}^K$ is the cross-product of K $\mathbb{S}$. $\mathbb{S}$ is assumed finite.
- Action space $\mathbb{A}^K$ is the cross-product of K $\mathbb{A}$. $\mathbb{A} = \{0, 1\}$.
- Reward $R_t(\mathbf{s}, \mathbf{a}) = \sum_{x=1}^K r_t(s_x, a_x)$, $1 \le t \le T$, is additive of the reward of individual sub-processes.
- Transition probability $\mathbb{P}^{\mathbf{a}}(\mathbf{s}', \mathbf{s}) = \prod_{x=1}^K P^{a_x}(s'_x, s_x)$.
- Starts at an initial state $\mathbf{s}_1$.

Problem Setup Con't

- A Markov policy $\boldsymbol{\pi} : \mathbb{S}^K \times \mathbb{A}^K \times \{1, ..., T\} \to [0,1]$, with $\boldsymbol{\pi}(\mathbf{s}, \mathbf{a}, t) = P(\mathbf{a}|\mathbf{S}_t = \mathbf{s})$ (Our decision).
  We require $\sum_{\mathbf{a}\in\mathbb{A}^K} \boldsymbol{\pi}(\mathbf{s}, \mathbf{a}, t) = 1.\ \forall \mathbf{s} \in \mathbb{S}^K, \forall 1 \le t \le T.$
- Objective

$$\begin{aligned} \underset{\boldsymbol{\pi}\in\boldsymbol{\Pi}}{\text{maximize}} \quad & \mathbb{E}^{\boldsymbol{\pi}}\left[\sum_{t=1}^{T} R_t(\mathbf{S}_t, \mathbf{A}_t)\right] \\ \text{subject to} \quad & P^{\boldsymbol{\pi}}(|\mathbf{A}_t| = m_t) = 1, \quad \forall 1 \le t \le T. \end{aligned} \tag{1}$$

Difficulty: Optimal solutions are computationally infeasible

Optimal solutions of (1) can be obtained with Bellman optimality equations.

But it requires $O(|\mathbb{S}|^K|\mathbb{A}|^K T)$ time complexity and $O(|\mathbb{S}|^K|\mathbb{A}|^K T)$ storage complexity.

The complexity grow exponentially with the number of sub-processes K, and becomes computationally infeasible for large K.

Past attempts

Relax the original problem (1) to

$$\begin{aligned} \underset{\boldsymbol{\pi}\in\boldsymbol{\Pi}}{\text{maximize}} \quad & \mathbb{E}^{\boldsymbol{\pi}}\left[\sum_{t=1}^{T} R_t(\mathbf{S}_t, \mathbf{A}_t)\right] \\ \text{subject to} \quad & \mathbb{E}^{\boldsymbol{\pi}}(|\mathbf{A}_t|) = m_t, \quad \forall 1 \le t \le T. \end{aligned} \tag{2}$$

The Lagrangian relaxation of (2)

$$P(\boldsymbol{\lambda}) = \max_{\boldsymbol{\pi}\in\boldsymbol{\Pi}} \mathbb{E}^{\boldsymbol{\pi}}\left[\sum_{t=1}^{T} R_t(\mathbf{S}_t, \mathbf{A}_t)\right] - \sum_{t=1}^{T} \lambda_t \left(\mathbb{E}^{\boldsymbol{\pi}}[|\mathbf{A}_t|] - m_t\right). \tag{3}$$

Pre-computations: 1. Optimal Lagrange Multiplier of (2)

Decomposition of the Lagrangian relaxation

$$P(\boldsymbol{\lambda}) = KQ(\boldsymbol{\lambda}) + \sum_t \lambda_t m_t, \tag{4}$$

where

$$Q(\boldsymbol{\lambda}) = \max_{\pi \in \Pi} \mathbb{E}^{\pi} \left[\sum_{t=1}^{T} r_t(S_t, A_t) - \lambda_t A_t \right], \tag{5}$$

is the objective function for sub-process $(\mathbb{S}, \mathbb{A}, P^{\cdot}, r)$. Definition of policy π is similar to $\boldsymbol{\pi}$, with $\pi(s, a, t) = P(a | S_t = s)$.

Pre-computations: 1. Optimal Lagrange Multiplier of (2)

Optimal Lagrange Multiplier $\boldsymbol{\lambda}^*$ is a solution to the Lagrange dual

$$\min_{\boldsymbol{\lambda}} P(\boldsymbol{\lambda}) = K \left(\min_{\boldsymbol{\lambda}} Q(\boldsymbol{\lambda}) + \sum_t \lambda_t \frac{m_t}{K} \right), \tag{6}$$

which can be solved by the following linear program (LP):

$$\begin{aligned}
\min_{\{V(s,t),\lambda_t : s\in\mathbb{S}, t\in\{1,\ldots,T\}\}} \quad & V(s_1,1) + \frac{1}{K}\sum_t \lambda_t m_t \\
\text{subject to} \quad & V(s,t) - \sum_{s'\in\mathbb{S}} P^a(s,s')V(s',t+1) \geq r_t(s,a) - \lambda_t a, \\
& \forall\, s\in\mathbb{S}, a\in\mathbb{A}, 1\leq t\leq T-1 \\
& V(s,T) \geq r_T(s,a) - \lambda_1 a \quad \forall\, s\in\mathbb{S}, a\in\mathbb{A}
\end{aligned} \tag{7}$$

where V^* corresponds to the value function of problem (5).

Remarks:

- Problem (7) has $O(|\mathbb{S}|T)$ variables and $O(|\mathbb{S}||\mathbb{A}|)T$ constraints, which is manageable.
- $P(\lambda)$ is an point-wise maxdimum of a group of affine functions of $\boldsymbol{\lambda}$, hence convex in $\boldsymbol{\lambda}$. (6) can also be solved by sub-gradient descent.

Pre-computations: 2. Occupation Measure ρ^* of an optimal policy of $Q(\lambda^*)$

Occupation measure of a policy π: the fraction of the time a process spent in each state-action pair at a time step under π.

To compute ρ^*, we solve the following linear program (LP):

$$\begin{aligned}
\max_{\rho} \quad & \sum_{t=1}^{T} \sum_{a \in \mathbb{A}} \sum_{s \in \mathbb{S}} \rho(s,a,t) r_t(s,a) \\
\text{subject to} \quad & \sum_{s \in \mathbb{S}} \rho(s,1,t) = \frac{m_t}{K}, \forall t = 1 \ldots, T \\
& \sum_{a \in \mathbb{A}} \rho(s,a,t) - \sum_{a \in \mathbb{A}} \sum_{s' \in \mathbb{S}} \rho(s',a,t-1) P^a(s',s) = 0, \forall s \in \mathbb{S}, 2 \le t \le T \\
& \sum_{a \in \mathbb{A}} \rho(s,a,1) = \mathbb{1}(s = s_1) \ \ \forall s \in \mathbb{S} \\
& \rho(s,a,t) \ge 0, \ \forall s \in \mathbb{S}, a \in \mathbb{A}, t = 1, \ldots, T.
\end{aligned} \tag{8}$$

Pre-computations: 2. Occupation Measure ρ^* of an optimal policy of $Q(\boldsymbol{\lambda}^*)$

Lemma

Let ρ^* be an optimal solution to (8), then ρ^* is the occupation measure of an optimal policy to $Q(\boldsymbol{\lambda}^*)$.

A quick justification of Lemma 1:
(8) is the dual of (7) attains $Q(\boldsymbol{\lambda}^*)$. By dynamic program theory ??????(Don't know what theory), solutions to (8) form the occupation measure of optimal solutions of $Q(\boldsymbol{\lambda}^*)$.

Pre-computations: 2. Occupation Measure ρ^* of an optimal policy of $Q(\boldsymbol{\lambda}^*)$

Remark:

- Any policy π^* constructed using ρ^* is an optimal solution to $Q(\boldsymbol{\lambda}^*)$ and satisfies

$$\mathbb{E}^{\pi}[|A_t|] = \frac{m_t}{K} \tag{9}$$

- Solving for ρ^* requires solving an LP with $|\mathbb{S}||\mathbb{A}|T$ variables and at most $T|\mathbb{S}|$ constraints.

Pre-computations: 3. Indices of states

We first describe a specific way of computing an optimal policy $\pi^{\boldsymbol{\lambda}}$ of $Q(\boldsymbol{\lambda})$:

Define value functions $V^{\boldsymbol{\lambda}} : \mathbb{S} \times \{1, ..., T\} \mapsto \mathbb{R}$ of $Q(\boldsymbol{\lambda})$ recursively by

$$V^{\boldsymbol{\lambda}}(s,t) = \begin{cases} \max_{a \in \mathbb{A}}\{r_T(s,a) - a\lambda_T\} & \text{if } t = T, \\ \max_{a \in \mathbb{A}}\{r_t(s,a) - a\lambda_t + \sum_{s' \in \mathbb{S}} P^a(s,s')V^{\boldsymbol{\lambda}}(s', t+1)\} & \text{o.w} \end{cases} \tag{10}$$

Construct $\pi^{\boldsymbol{\lambda}}$ by

$$\pi^{\boldsymbol{\lambda}}(s,1,t) = \begin{cases} 1 & \text{if one-step lookahead value for } a = 1 \text{ is} \\ & \text{greater than or equal to } a = 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Pre-computations: 3. Indices of states