Multi-objective Bandits: Optimizing the Generalized Gini Index

Team Quark
Aagam Shah
Kushagra Chandak

Introduction: Multi-armed Bandits (MAB)

Plain vanilla version:

An agent repeatedly chooses among K options, metaphorically corresponding to pulling one of K arms of a bandit machine.

Agent receives reward/cost: depends on the selected arm.

Goal: Optimize an evaluation metric i.e. a form of reward or loss.

Then what's new?

Now, we have multiple evaluation metrics and thus multiple objectives:
Multi-Objective MAB.

Goal: Find a policy, which can optimize these objectives simultaneously in a fair way.

Formalized using Generalized Gini Index (GGI) aggregation function.

What is being done: An online gradient descent algorithm is proposed which exploits the convexity of the GGI aggregation function, and controls the exploration in a careful way.

Setting

A D-objective K-armed bandit problem is specified by K real valued multivariate random variables $\mathbf{X}_1, \ldots, \mathbf{X}_k$ over $[0,1]^D$.

Each time step: Agent selects one of the arms and obtains a cost vector from the corresponding distribution for various objectives.

Samples are assumed to be independent over time and across the arms, but not necessarily across the components of the cost vector.

At time step t, k(t) denotes the index of the arm played and $X_{k(t)}$ the resulting payoff. Associate $\boldsymbol{\mu}_k = \mathbb{E}[\mathbf{X}_k]$ with each arm as the expected vectorial cost of arm k.

Pareto Front

Notion of optimality: We define the Pareto dominance relation ≤ for vectors $\boldsymbol{v}, \boldsymbol{v}'$ as follows;

$$\boldsymbol{v} \leq \boldsymbol{v}' \Leftrightarrow \forall d = 1,\ldots,D, v_d \leq v_d'$$

Let $\mathcal{O} \subseteq \mathcal{R}^D$ be a set of D-dimensional vectors.The Pareto front of $\mathcal{O}$, denoted by $\mathcal{O}^*$, is a set of vectors such that;

$$\boldsymbol{v}^* \in \mathcal{O}^* \Leftrightarrow (\forall \boldsymbol{v} \in \mathcal{O}, \boldsymbol{v} \leq \boldsymbol{v}^* \Rightarrow \boldsymbol{v} = \boldsymbol{v}^*)$$

In multi objective optimization, one usually wants to compute the Pareto front or search for a particular element of the Pareto front.

Generalized Gini Index

$$G_w(x) = \sum_{d=1}^{D} w_d x_{\sigma(d)} = w^T x_\sigma$$

- The components in the cost and weight vector are sorted in non-increasing order. Given this assumption, G(x) is a piecewise linear convex function.

- GGI: aggregation or scalarizing function. (to compare return of arms)
  Non decreasing; allows every vector to receive a scalar value to be optimized.
  A solution to this problem yields a particular solution on the Pareto front.

- GGI formulation in terms of Lorenz vectors is

$$G_w(x) = \sum_{d=1}^{D} w'_d L_d(x)$$

Is GGI fair? Pigou-Dalton Transfer

- Increasing a lower valued objective by the same quantity s.t. the order between the two objectives is not reversed: the effect is to balance a cost vector.

$$\forall \mathbf{x} \text{ s.t. } \mathbf{x_i} < \mathbf{x_j}, \forall \epsilon \in (\mathbf{0}, \mathbf{x_j} - \mathbf{x_i}), \mathbf{G_w}(\mathbf{x} + \epsilon \mathbf{e_i} - \epsilon \mathbf{e_j}) \leq \mathbf{G_w}(\mathbf{x})$$

- As a consequence, among vectors of equal sum, the best cost vector (w.r.t. GGI) is the one with equal values in all objectives if feasible.

- GGI is decreasing with Pigou Dalton transfers and all the components of w’ are positive and in the range [0,1].

- Pareto dominance and Pigou-Dalton transfer are the two principles formulating natural requirements: balance GGI.

Optimal Policy

- Compute the GGI score of each arm k if its vectorial mean $\boldsymbol{\mu}_{\mathbf{k}}$ is known. Optimal arm k^* minimizes the GGI score as:

$$k^* \in \min_{k \in [K]} G_w(\mu_k)$$

- Mixed strategies: Each arm has a probability of being picked.
- These strategies may reach to lower GGI values than any fixed arm. A policy parameterized by α chooses arm k with probability α_k which can be obtained as follows:

$$\alpha^* \in \min_{\alpha \in A} G_w\left(\sum_{k=1}^{K} \alpha_k \mu_k\right)$$

Regret

- After playing T rounds, the average cost can be written as:

$$\bar{X}^{(T)} = \frac{1}{T}\sum_{t=1}^{T} X_{k_t}^{(t)}$$

- Goal: Minimize the GGI index of this term.
- Regret:

$$R^{(T)} = G_w(\bar{X}^{(T)}) - G_w(\mu\alpha^*)$$

- Pseudo regret:

$$\bar{R}^{(T)} = G_w(\mu\bar{\alpha}^{(T)}) - G_w(\mu\alpha^*)$$

Algorithm

- To minimize GGI: Use Online Gradient Descent algorithm.
  Multi-Objective Online Gradient Descent algorithm with Exploration (MO-OGDE).

- Outline: Pull each arm once as an initialization step.
  Then in each iteration, choose arm k with probability $\alpha_k(t)$ and compute the objective function based on empirical mean estimates.
  Next, the algorithm carries out the gradient step and computes the projection onto the nearest point of the convex set of α.

- Forced exploration is indispensable, since the objective function depends on the means of the arm distributions, which are not known.

Algorithm 1 MO-OGDE(δ)

Pull each arm once

Set $\boldsymbol{\alpha}^{(K+1)} = (1/K, \cdots, 1/K)$

for rounds $t = K+1, K+2, \ldots$ **do**

  Choose an arm k_t according to $\boldsymbol{\alpha}^{(t)}$

  Observe the sample $\mathbf{X}_{k_t}^{(t)}$ and compute $f^{(t)}$

  Set $\eta_t = \frac{\sqrt{2}}{(1-1/\sqrt{K})}\sqrt{\frac{\ln(2/\delta)}{t}}$

  $\boldsymbol{\alpha}^{(t+1)} = \mathrm{OGDEstep}(\boldsymbol{\alpha}^{(t)}, \eta_t, \nabla f^{(t)})$

return $\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\alpha}^{(t)}$

Experiment: Battery Control Task

Arms are the different cell control strategies; efficient balancing of cells needed for better battery performance.
Goal: State of charge, temperature and aging.

Strategy for learner:

- Chooses a control strategy for a short duration.
- Observes it's effects on the objectives (due to stochastic electric consumption).

Formulated as GGI optimization problem: evaluated MO-OGDE and MO-LP.
MO-OGDE is computationally more efficient.

Plan for final evaluation

- To update the code with the projection step.
- To use application dependent distributions for the cost vectors.
- Compare the performance MO-LP and MO-OGDE.
- Investigate the empirical performance of the algorithm by changing the parameters.