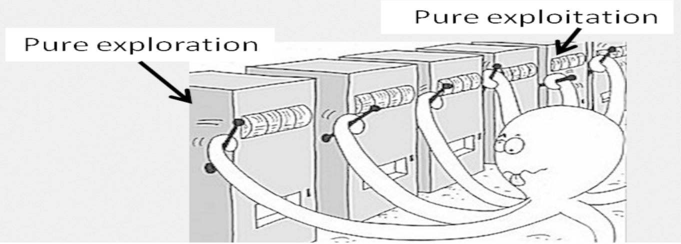


Multi-objective Bandits

Optimizing the Generalized Gini Index

Aagam Shah, Kushagra Chandak

Topics in Machine Learning

Table of contents

1. First Presentation Overview
2. Regret Analysis
3. Results
4. Conclusion

First Presentation Overview

Introduction

Plain vanilla MAB vs multi-objective MAB

The goal is to find a policy, which can *optimize* these objectives simultaneously in a *fair* way.

The problem is formalized using the Generalized Gini Index (GGI) aggregation function, and an online gradient descent algorithm is proposed to find the optimal policy.

Problem setting

A D-objective K-armed bandit problem is specified by K real valued multivariate random variables $X_1, ..., X_k$ over $[0, 1]^D$.

Agent selects one of the arms at each time step 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 $\mu_k = \mathbb{E}[X_k]$ with each arm as the expected vectorial cost of arm k.

Pareto Front

The Pareto dominance relation for vectors v and v' is

$$v \preceq v' \iff \forall d \in [D] \;\; v_d \leq v'_d$$

Let Θ ⊆ D be a set of D-dimensional vectors. The Pareto front of Θ , denoted by Θ^*, is a set of vectors such that;

$$v^* \in \Theta^* \iff (\forall v \in \Theta, v \preceq v^* \Rightarrow v = v^*)$$

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

Generalized Gini Index

GGI is an aggregation or scalarizing function to compare return of arms, given by:

$$\mathsf{G}_{\mathsf{w}}(\mathsf{x}) = \sum_{d=1}^{D} w_d x_{\sigma(d)} = \mathbf{w}^{\mathsf{T}} \mathbf{x}_{\sigma}$$

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.

It allows every vector to receive a scalar value to be optimized. A solution for minimizing this function yields a particular solution on the Pareto front.

Pigou Dalton transfers

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

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

GGI decreases with Pigou Dalton transfers and as a consequence, among vectors of equal sum, the best cost vector is the one with equal values in all objectives if feasible.

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

Optimal Policy

Pure strategies: Compute the GGI score of each arm k if its vectorial mean μ_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:

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

Algorithm

Minimize GGI: **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 α_{k_t} and compute the objective function based on empirical mean estimates.
- Gradient step and projection onto the nearest point of the convex set of α.
- Forced exploration: since the objective function depends on the means of the arm distributions, which are not known.

Algorithm

Algorithm 1 MOMAB algorithm

```
Pull each arm once
Set $\alpha^{K+1} = (1/K, ..., 1/K)$
for rounds $t = K+1, K+2, ...$ do
    Choose arm $k_t$ according to $\alpha^{(t)}$
    Observe the sample $X_{k_t}^{(t)}$ and compute $f^{(t)}$
    Set $\eta_t = \frac{\sqrt{2}}{1-1/\sqrt{K}} \sqrt{\frac{ln(2/\delta)}{t}}$
    $\alpha^{(t+1)} = \alpha^{(t)} - \eta_t \bigtriangledown f^{(t)}$
return 1/T $\sum_{t=1}^{T} \alpha^{(t)}$
```

Regret Analysis

Regret: Introduction and Notation

Learner's goal: Minimize GGI of average cost.
Average Cost:

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

Regret:

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

Pseudo Regret:

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

Regret analysis: Difference between $R^{(T)}$ and $\overline{R}^{(T)}$ is $O(T^{-1/2})$ with high probability, thus having a high probability regret bound $O(T^{-1/2})$ for one of them implies a similar order regret bound for the other one.

Regret Analysis

Main idea:

- Apply some online convex optimization algo on the current estimate $f^{(t)}(\alpha) = G_w(\hat{\mu}^{(t)}\alpha)$ of the objective function $f(\alpha) = G_w(\mu\alpha)$
- Use forced exploration of order $T^{1/2}$ and finally
- Show that the estimate of the objective function has error of $O(T^{-1/2})$ along the trajectory generated by the algo.

In particular,

$$f(\frac{1}{T}\sum_{t=1}^{T}\alpha^{(t)}) \leq \frac{1}{T}\sum_{t=1}^{T} f^{(t)}(\alpha^{(t)}) + O(T^{-1/2})$$

Mathematical Results

Theorem 1: With probability at least $1 - \delta$:

$$f(\frac{1}{T}\sum_{t=1}^{T}\alpha^{(t)}) - f(\alpha^*) \leq 2L\sqrt{\frac{6D\log^3(8DKT^2/\delta)}{T}}$$

for big enough T where L is the Lipshitz constant of $G_w(x)$.

Proposition 1: With probability at least $1 - 2(DT+1)K\delta$,

$$|G_w(\frac{1}{T}\sum_{t=1}^{T}\mu\alpha^{(t)}) - G_w(\frac{1}{T}\sum_{t=1}^{T}\hat{\mu}^{(t)}\alpha^{(t)})| \leq L\sqrt{\frac{6D(1+\log^2 T)\log(2/\delta)}{T}}$$

Proposition 1 along with convexity of GGI gives us:

Corollary 1: With probability at least $1 - 2(DT+1)K\delta$

$$f(\overline{\alpha}^{(t)}) \leq \frac{1}{T}\sum_{t=1}^{T}f^{(t)}(\alpha^{(t)} + L\sqrt{\frac{6D(1+ln^2T)ln(2/\delta)}{T}})$$

Regret vs Pseudo Regret

Claim 1: For any $t = 1, 2...$ and any $k = 1...K$, it holds that

$$P[|T_k(t) - \sum_{\tau=1}^{t} \alpha_k^{(\tau)} \geq \sqrt{2t \log(2/\delta)}] \leq \delta$$

Using Claim 1 and Propostion 2, we bound the difference between the regrets.

Corollary 2: With probability at least $1 - \delta$

$$|R^{(T)} - \overline{R}^{(T)}| \leq L \sqrt{\frac{12D \log\left(4(DT+1)/\delta\right)}{T}}$$

The difference $R^{(T)}$ and $\overline{R}^{(T)}$ is $O(T^{-1/2})$ with high probability, hence Theorem 1 implies a regret of $O(T^{-1/2})$ for MO-OGDE.

Results

Linear Programming Approach

minimize

$$\sum_{d=1}^{D} w_d{}' (dr_d + \sum_{j=1}^{D} b_{j,d})$$

subject to

$$r_d + b_{j,d} \geq \sum_{k=1}^{K} \alpha_k \hat{\mu}_{k,j}^{(t)} \quad \forall j, d \in [D]$$

$$\alpha^T 1 = 1$$

$$\alpha \geq \eta_t / K$$

$$b_{j,d} \geq 0$$

Synthetic Experiment

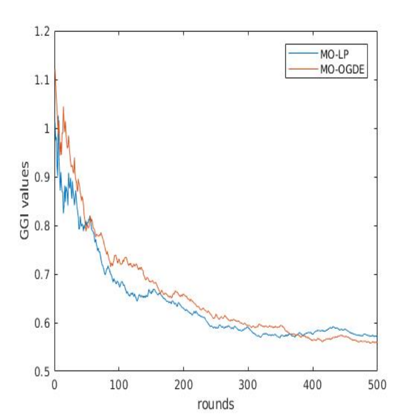


Figure 1: K=5, D=5

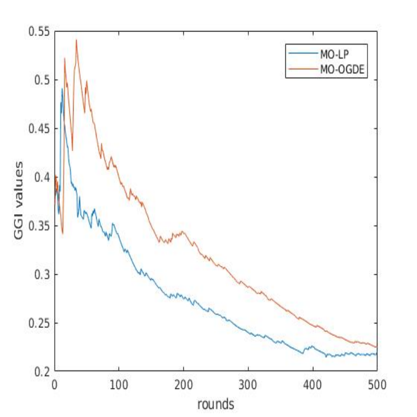


Figure 2: K=20, D=5

Placement season is here!

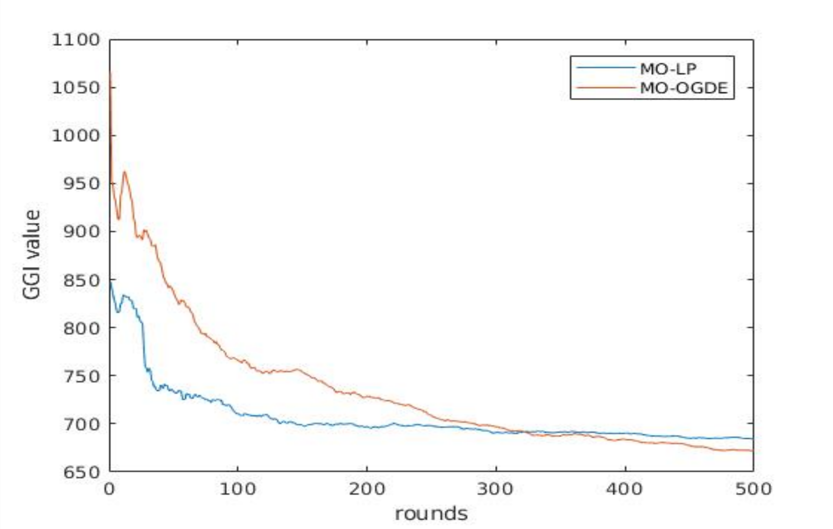
- Given companies as arms, the goal is to select best possible company to work in (according to personal preferences) while optimizing multiple criteria such as salary, location, working hours, paid leaves, reputation etc.
- We also see the effect of weights on different objectives.

Data

Table 1: Company data for freshers (source: Glassdoor)

Company	Salary	CTC	Location	WH	PL	Reputation
Directi	17	12	random	8	random	1
Google	14	10	random	8	random	1
Samsung	12	10	random	8	random	2
Uber	20	14	random	10	random	1
Amazon	15	10	random	7	random	1
TCS	4	4	random	8	random	5
Accenture	5	5	random	9	random	4
Deloitte	7	6	random	9	random	3
Infosys	6	5	random	8	random	3
InMobi	15	15	random	10	random	2

GGI for MO-LP vs MO-OGDE



And the winner is...

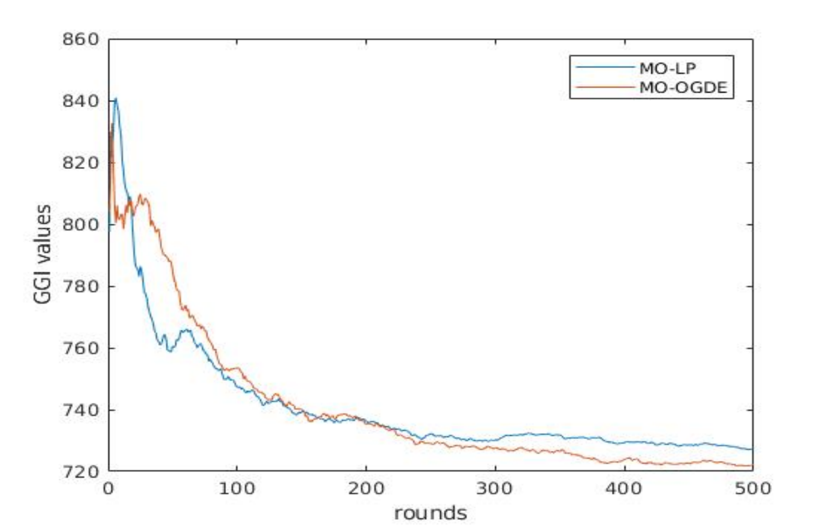
Table 2: Weights for various objectives

Objective	Salary	CTC	Location	WH	PL	Reputation
Weights	80	7	5	5	2	1

Table 3: Company finally selected: Uber

Companies	Directi TCS	Google Accenture	Samsung Deloitte	Uber Infosys	Amazon InMobi
Probabilities	0.0088 0.0088	0.0088 0.0088	0.0088 0.0088	0.9211 0.0088	0.0088 0.0088

What if we changed the weights?



And the new winner is...

Table 4: New weights for various objectives

Objective	Salary	CTC	Location	WH	PL	Reputation
Weights	30	20	20	20	5	5

Table 5: New company finally selected: Directi

Companies	Directi TCS	Google Accenture	Samsung Deloitte	Uber Infosys	Amazon InMobi
Probabilities	0.9211 0.0088	0.0088 0.0088	0.0088 0.0088	0.0088 0.0088	0.0088 0.0088

Conclusion

Summary and future work

- Incorporated fairness and optimality using Pigou-Dalton principle and Pareto front in the objective function.
- Optimized GGI using a gradient descent approach and compared it to a linear programming approach (baseline).
- MO-LP converges faster but is computationally expensive as compared to MO-OGDE.
- In future, check different aggregation functions (other than GGI).
- Extend the work to full reinforcement learning setting.

Thank You!