Pandora's Box with Correlations: Learning and Approximation

Shuchi Chawla[1], *Evangelia Gergatsouli*[1], Yifeng Teng[1], Christos Tzamos[1], Ruimin Zhang[1]

[1]University of Wisconsin-Madison

CoReLab study group, September 2020

A Search Problem

Find the best out of n alternatives!

A Search Problem

Find the best out of n alternatives!

$\mathcal{D}_0$ $\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$ $\mathcal{D}_4$

- Stochastic information on value

A Search Problem

Find the best out of n alternatives!

$\mathcal{D}_0$ $\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$ $\mathcal{D}_4$

1 4 2 7 3

- Stochastic information on value
- Information is not free!

A Search Problem

Find the best out of n alternatives!

42	??	17	13	??
1	4	2	7	3

- Stochastic information on value
- Information is not free!

A Search Problem

Find the best out of n alternatives!

42	??	17	13	??
1	4	2	7	3

- Stochastic information on value
- Information is not free!
- Open boxes until decide to stop (*stopping rule*).
- Keep best value seen so far

Instatiation of values = *scenario*.

A Search Problem

Find the best out of n alternatives!

42	??	17	13	??
1	4	2	7	3

- Stochastic information on value
- Information is not free!

Maximization version: *max value - information cost*
Minimization version: *min value + information cost*

This paper: focus on minimization

A Search Problem - What do we know

Pandora's Box[1] greedy gives optimal!

- Assign an index to every box
- Search boxes in order of index until: current value better than index of next box

[1] [Weitzman *Econometrica* 1979]

A Search Problem - What do we know

Pandora's Box[1] greedy gives optimal!

- Assign an index to every box
- Search boxes in order of index until: current value better than index of next box

Crucial assumption: distributions are **independent**!

[1] [Weitzman *Econometrica* 1979]

A Search Problem - What do we know

Pandora's Box[1] greedy gives optimal!

- Assign an index to every box
- Search boxes in order of index until: current value better than index of next box

Crucial assumption: distributions are **independent**!

What about **correlation**?

[1] [Weitzman *Econometrica* 1979]

Approximating the Optimal

Hard Problem: encode location of best box in values of other boxes

Box i *Box j* *Box 42*



. . .


4 2 0

Example: values 4 and 2 means go to box 42 to find best value

Approximating the Optimal

Hard Problem: encode location of best box in values of other boxes

Box i | *Box j* | ... | *Box 42*



4 | 2 | | 0

Example: values 4 and 2 means go to box 42 to find best value

Cannot learn arbitrary mapping with finitely many queries!

Approximating the Optimal

Hard Problem: encode location of best box in values of other boxes

Box i	*Box j*		*Box 42*
		...	
4	2		0

Example: values 4 and 2 means go to box 42 to find best value

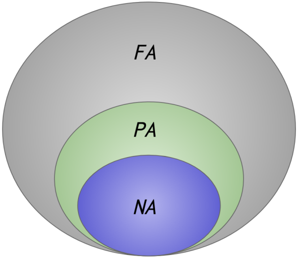
Cannot learn arbitrary mapping with finitely many queries!

Best Strategy: decide next box after seen values. Other strategies?

Strategies

Strategy: (1) What is next box? (2) When do I stop?

- *Fully Adaptive*: next box/stopping rule **both** adaptive
- *Partially Adaptive*: fixed order, adaptive stopping time (for independent $\mathcal{D}$ this gives optimal policy!)
- *Non-Adaptive*: fixed order **and** stopping time



Approximating Other Strategies

- Fully Adaptive: Learning/Approximation: Hard!
  Example: encoded location of best box
- Non-Adaptive:
  - Learning: Hard!: tiny probability scenario has value=∞ on all boxes but one$\rightarrow$either query all boxes **or** sample this scenario
  - Approximation: As hard as Set Cover! For $0/\infty$ values $\rightarrow$ find a 0 for every scenario $\rightarrow$ hitting set formulation of set cover
- Partially Adaptive: Can Learn & Efficiently approximate!

Main Theorem

Using polynomially in n samples we can efficiently find a Partially Adaptive strategy that is $O(1)$-competitive against the optimal Partially Adaptive strategy.

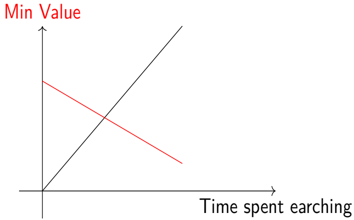
An Easier Partially Adaptive Family

Space of PA strategies can be large! $\rightarrow$ Scenario-aware PA
SPA: Fix order $\rightarrow$ scenario is revealed$\rightarrow$decide stopping time

Lemma

For any order, there is an adaptive stopping rule that 2-approximates the optimal Scenario-aware stopping rule.

Proof Sketch: Assume a SPA order$\rightarrow$ need to find a stopping rule for PA. Stop when best value seen so far is at most time spent until now[a].



[a] Argument is equivalent to Ski-Rental$\rightarrow$ can get 1.58 using ski rental algorithm.

An Easier Partially Adaptive Family

Space of PA strategies can be large! $\rightarrow$ Scenario-aware PA
SPA: Fix order $\rightarrow$ scenario is revealed$\rightarrow$decide stopping time

Lemma

For any order, there is an adaptive stopping rule that 2-approximates the optimal Scenario-aware stopping rule.

Focus on SPA then convert to PA losing a factor of 2.

An Easier Partially Adaptive Family

Lemma

Near-Optimal SPA Strategies can be efficiently learned from poly(n) number of samples.

Proof Sketch.

Possible permutations: $n!$

Each permutation has bounded cost$\rightarrow$can learn with few samples$\rightarrow$ union bound on all $n!$ permutations. $\square$

Enough to find good SPA strategies!

An Easier Partially Adaptive Family

Lemma

Near-Optimal SPA Strategies can be efficiently learned from poly(n) number of samples.

Proof Sketch.

Possible permutations: $n!$

Each permutation has bounded cost$\rightarrow$can learn with few samples$\rightarrow$ union bound on all $n!$ permutations. $\square$

Enough to find good SPA strategies!

This talk:

1. **SPA vs NA**
2. **SPA vs PA (main result)**

PA vs NA - LP Formulation

$$
\begin{array}{rrcll}
\text{minimize} & \sum_{i \in \mathcal{B}} x_i & + & \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{B}, s \in \mathcal{S}} c_{is} z_{is} & & \text{(LP-NA)} \\
\text{subject to} & \sum_{i \in \mathcal{B}} z_{is} & = & 1, & \forall s \in \mathcal{S} & (1) \\
& z_{is} & \leq & x_i, & \forall i \in \mathcal{B}, s \in \mathcal{S} & \\
& x_i, z_{is} & \in & [0, 1] & \forall i \in \mathcal{B}, s \in \mathcal{S} &
\end{array}
$$

x_i: indicates whether box i is opened

z_{is}: indicates whether box i is assigned to scenario s

c_{is}: value in box i for scenario s

PA vs NA - Algorithm

Given: Solution $\boldsymbol{x}, \boldsymbol{z}$ to LP, scenario s

1. Open box i wp $\frac{x_i}{\sum_{i\in\mathcal{B}} x_i}$
2. If box i is opened, select the box and stop wp $\frac{z_{is}}{x_i}$

Analysis: Bound probing cost + value

- Part 1: bound probing cost

$$\mathbf{Pr}\left[\text{stop at step } t\right] = \sum_{i\in\mathcal{B}} \frac{x_i}{\sum_{i\in\mathcal{B}} x_i} \frac{z_{is}}{x_i} = \frac{\sum_{i\in\mathcal{B}} z_{is}}{\sum_{i\in\mathcal{B}} x_i} = \frac{1}{\mathrm{OPT}_t},$$

Probing cost is optimal on expectation

PA vs NA - Analysis

- Part 2: bound the value
  For scenario s

$$
\begin{aligned}
\mathbf{E}\left[\mathrm{ALG}_{c,s}\right] &= \sum_{i\in\mathcal{B},t} \mathbf{Pr}\left[\text{select } i \text{ at } t \mid \text{stop at } t\right] \mathbf{Pr}\left[\text{stop at } t\right] c_{is} \\
&\le \sum_{i\in\mathcal{B},t} \frac{z_{is}}{\sum_{i\in\mathcal{B}} z_{is}} \mathbf{Pr}\left[\text{stop at } t\right] c_{is} \\
&= \sum_{i\in\mathcal{B}} z_{is} c_{is} \\
&= \mathrm{OPT}_{c,s}
\end{aligned}
$$

Take expectation over all scenarios $\mathbf{E}\left[\mathrm{ALG}_c\right] \le \mathrm{OPT}_c$

SPA Approximates NA $\rightarrow$ lose a 2-factor to convert to PA

PA vs PA - LP Formulation

LP-SPA

min $\frac{1}{|\mathcal{S}|}\sum_{i,s,t} t z_{ist} + \frac{1}{|\mathcal{S}|}\sum_{i,s,t} c_{is} z_{ist}$

LP-MSSC

min $\frac{1}{|\mathcal{S}|}\sum_{i,s,t} t z_{ist}$

Subject to: (1) Every time, pick 1 box/set

(2) Every box/set can be chosen at most once

(3) Only choose value/zero of already opened box/set

(4) Choose exactly/at least one box/set per scenario/element

PA vs PA - Reduction to MSSC

LP-SPA
$\mathcal{I} = (\boldsymbol{x}, \boldsymbol{z})$

PA vs PA - Reduction to MSSC

LP-SPA
$\mathcal{I} = (\boldsymbol{x}, \boldsymbol{z})$

$\longrightarrow$

LP-MSSC
$\mathcal{I}' = (\boldsymbol{x}', \boldsymbol{z}')$

Lose a factor $\left(\frac{\alpha}{\alpha-1}\right)^2$

Focus on "low cost" boxes

Use MSSC to find one fast

PA vs PA - Reduction to MSSC

$$\begin{array}{c}\textbf{LP-SPA}\\ \mathcal{I} = (\boldsymbol{x}, \boldsymbol{z})\end{array} \longrightarrow \begin{array}{c}\textbf{LP-MSSC}\\ \mathcal{I}' = (\boldsymbol{x}', \boldsymbol{z}')\end{array} \longrightarrow \begin{array}{c}\textbf{Integer MSSC}\\ \mathcal{I}'' = \text{pr.}\\ \text{order } \sigma\end{array}$$

Lose a factor $\left(\frac{\alpha}{\alpha-1}\right)^2$
Focus on "low cost" boxes
Use MSSC to find one fast

Lose a factor 4
Round fractional MSSC
Using greedy from
Feige et al. *APPROX* 2002

General Costs

So far: probing time/cost $p_i = 1$. **General costs?**

General Costs

So far: probing time/cost $p_i = 1$. **General costs?**

1. Round each cost to closest integer p_i
   Costs need to be polynomially bounded

General Costs

So far: probing time/cost $p_i = 1$. **General costs?**

1. Round each cost to closest integer p_i
   Costs need to be polynomially bounded
2. Modify LPs
   2.1 NA: Add cost to objective

$$\sum_{i \in \mathcal{B}} x_i \to \sum_{i \in \mathcal{B}} x_i p_i$$

   2.2 PA: Each opened box should be probed for p_i steps

$$\sum_{i \in \mathcal{B}} x_{it} = 1 \to \sum_{i \in \mathcal{B}} \sum_{t \leq t' \leq t+p_i-1} x_{it'} \leq 1$$

General Costs

So far: probing time/cost $p_i = 1$. **General costs?**

1. Round each cost to closest integer p_i
   Costs need to be polynomially bounded
2. Modify LPs
   2.1 NA: Add cost to objective

$$\sum_{i \in \mathcal{B}} x_i \to \sum_{i \in \mathcal{B}} x_i p_i$$

   2.2 PA: Each opened box should be probed for p_i steps

$$\sum_{i \in \mathcal{B}} x_{it} = 1 \to \sum_{i \in \mathcal{B}} \sum_{t \le t' \le t + p_i - 1} x_{it'} \le 1$$

All previous results still hold within constant.

Summary - Extensions

- Moving the optimal can help us find meaningful approximations
- PA for minimization of Pandora's box can be efficiently approximated

What about maximization?

Summary - Extensions

- Moving the optimal can help us find meaningful approximations
- PA for minimization of Pandora's box can be efficiently approximated

What about maximization?

- We cannot approximate the Non-Adaptive using a Fully Adaptive within any constant.

Proof Sketch.

Set Cover: sets + elements $=$ Search problem boxes + scenarios

- If element is covered by set $\Rightarrow$ box has high value for scenario
- Bad scenario with very low cost

Idea: NA is SC, cannot cover a significant portion with any FA $\square$

Summary - Extensions

What about complex feasibility constraints?

[2] Bansal et al. *SODA* 2010

Summary - Extensions

What about complex feasibility constraints?

- We can approximate the PA for selecting k items within $O(1)$.
- We can approximate the PA for selecting a matroid base of rank k within $\Theta(\log k)$.

[2] Bansal et al. *SODA* 2010

Summary - Extensions

What about complex feasibility constraints?

- We can approximate the PA for selecting k items within $O(1)$.
- We can approximate the PA for selecting a matroid base of rank k within $\Theta(\log k)$.

Both k items and matroids:

Generalized MSSC[2]: for each set select at least $k(S)$ elements

Use Generalized MSSC LP for SPA $\rightarrow$ similar to PA vs NA $\square$

[2] Bansal et al. *SODA* 2010

Future directions

Our work: tradeoff adaptivity vs computational complexity

Future Directions:

- What can we approximate by fully adaptive strategies?
  In this direction$\rightarrow$
  - Cannot approximate NA within constant.
- Can adaptive methods give efficient algos for hard problems?

Future directions

Our work: tradeoff adaptivity vs computational complexity

Future Directions:

- What can we approximate by fully adaptive strategies?
  In this direction$\rightarrow$
  - Cannot approximate NA within constant.
- Can adaptive methods give efficient algos for hard problems?

Thank you!