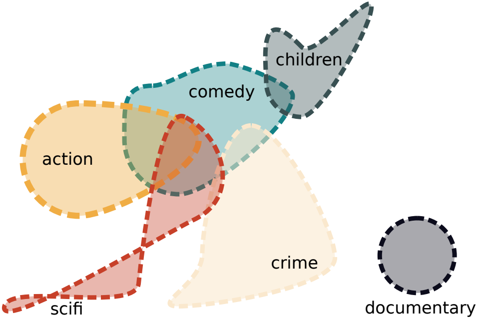
Adaptive Submodular Maximization in Bandit Settings

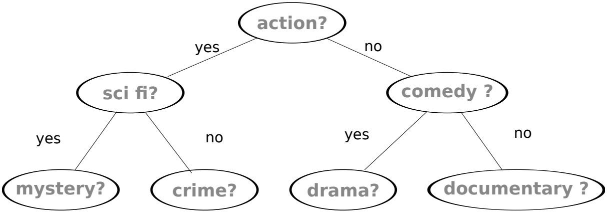
Joint work with Branislav Kveton, Zheng Wen,
Brian Eriksson, & S. Muthukrishnan

Submodular Maximization

Preference elicitation example

Try to identify as many as possible interesting movies for a user in **K** steps.



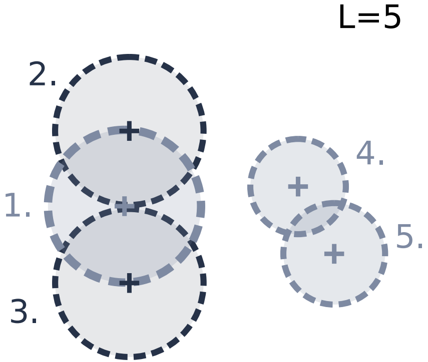


Sensor activation problem.

- 5 sensors ($L = 5$) whose placements is fixed (and known).

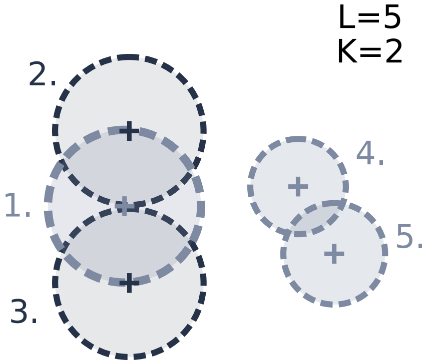
Sensor activation problem.

- 5 sensors ($L = 5$) whose placements is fixed (and known).
- Each area covered by a sensor is known



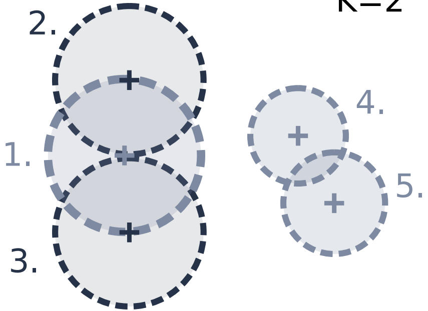
Sensor activation problem.

- 5 sensors ($L = 5$) whose placements is fixed (and known).
- Each area covered by a sensor is known
- **Goal:** Maximize the total covered area with 2 sensors ($K = 2$).



Sensor activation problem.

- 5 sensors ($L = 5$) whose placements is fixed (and known).
- Each area covered by a sensor is known
- **Goal:** Maximize the total covered area with 2 sensors ($K = 2$).

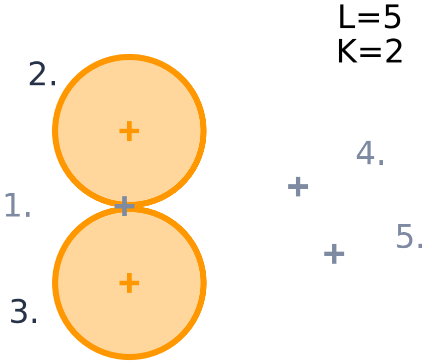


L=5
K=2

Good Solutions:

Sensor activation problem.

- 5 sensors ($L = 5$) whose placements is fixed (and known).
- Each area covered by a sensor is known
- **Goal:** Maximize the total covered area with 2 sensors ($K = 2$).

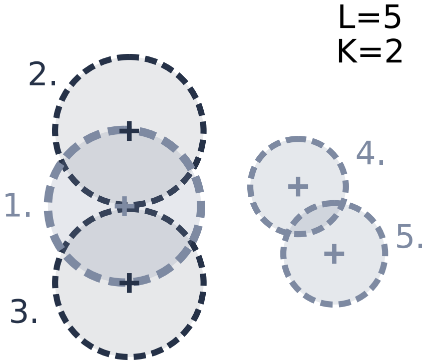


Good Solutions:

Optimal Solution: 2.& 3.

Sensor activation problem.

- 5 sensors ($L = 5$) whose placements is fixed (and known).
- Each area covered by a sensor is known
- **Goal:** Maximize the total covered area with 2 sensors ($K = 2$).



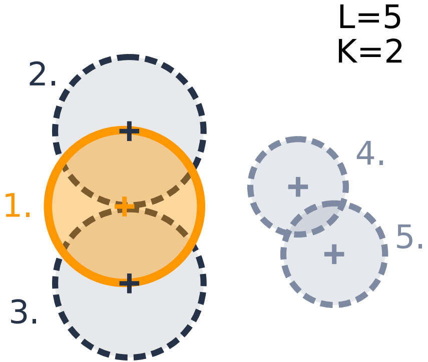
Good Solutions:

Optimal Solution: 2.& 3.

Greedy Solution:

Sensor activation problem.

- 5 sensors ($L = 5$) whose placements is fixed (and known).
- Each area covered by a sensor is known
- **Goal:** Maximize the total covered area with 2 sensors ($K = 2$).

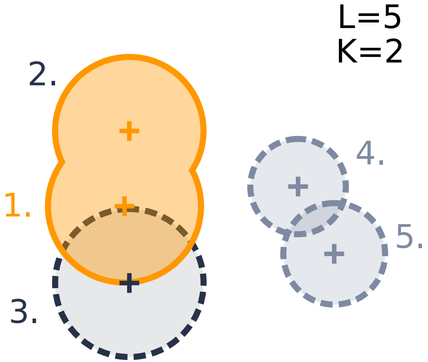


Good Solutions:

Optimal Solution: 2.& 3.
Greedy Solution: 1.

Sensor activation problem.

- 5 sensors ($L = 5$) whose placements is fixed (and known).
- Each area covered by a sensor is known
- **Goal:** Maximize the total covered area with 2 sensors ($K = 2$).

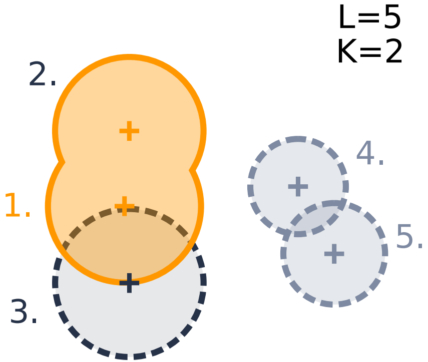


Good Solutions:

Optimal Solution: 2.& 3.
Greedy Solution: 1.& 2.

Sensor activation problem.

- 5 sensors ($L = 5$) whose placements is fixed (and known).
- Each area covered by a sensor is known
- **Goal:** Maximize the total covered area with 2 sensors ($K = 2$).



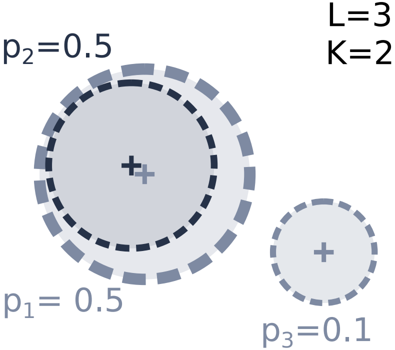
Good Solutions:

Optimal Solution: 2.& 3.
Greedy Solution: 1.& 2.

Because the problem is submodular, the greedy solution is optimal up to a constant multiplicative factor ($1 - 1/e \approx 0.63$)

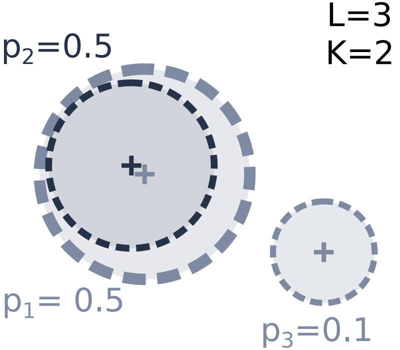
Stochastic sensor activation problem.

- Upon activation, each sensor i cover its area with probability p_i. There is 2 **states**: failure or success.
- **Goal:** Maximize the total **expected** covered area with 2 sensors, **chosen in advance**.



Stochastic sensor activation problem.

- Upon activation, each sensor i cover its area with probability p_i. There is 2 **states**: failure or success.
- **Goal:** Maximize the total **expected** covered area with 2 sensors, **chosen in advance**.



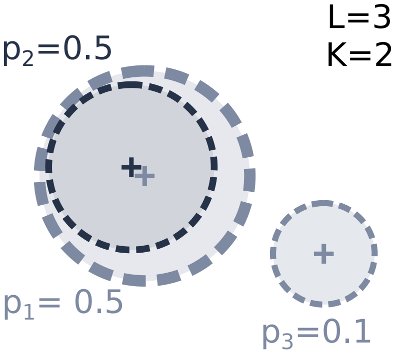
Good Solutions:

Optimal solution: 1.& 2.
Greedy solution: 1.& 2.

Again, the greedy solution is $(1 - 1/e)$−optimal

Adaptive sensor activation problem.

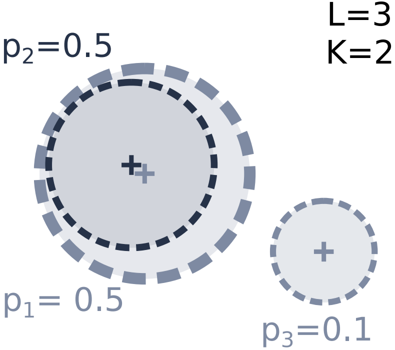
- Upon activation, each sensor i cover its area with probability p_i and the state of the sensor is **observed**.
- **Goal:** Maximize the total **expected** covered area with 2 sensors, **chosen adaptively**.



Adaptive sensor activation problem.

- Upon activation, each sensor i cover its area with probability p_i and the state of the sensor is **observed**.
- **Goal:** Maximize the total **expected** covered area with 2 sensors, **chosen adaptively**.

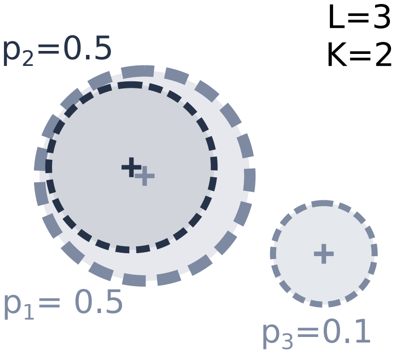
Good Policies:



Adaptive sensor activation problem.

- Upon activation, each sensor i cover its area with probability p_i and the state of the sensor is **observed**.
- **Goal:** Maximize the total **expected** covered area with 2 sensors, **chosen adaptively**.

Good Policies:

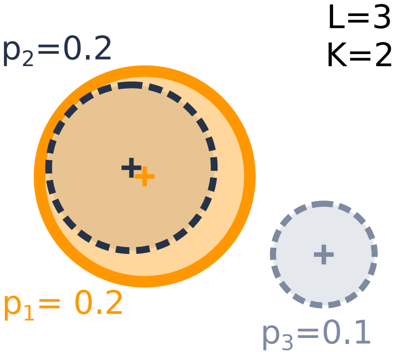


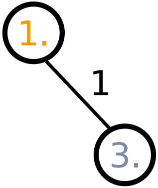
1.

Adaptive sensor activation problem.

- Upon activation, each sensor *i* cover its area with probability p_i and the state of the sensor is **observed**.
- **Goal:** Maximize the total **expected** covered area with 2 sensors, **chosen adaptively**.

Good Policies:



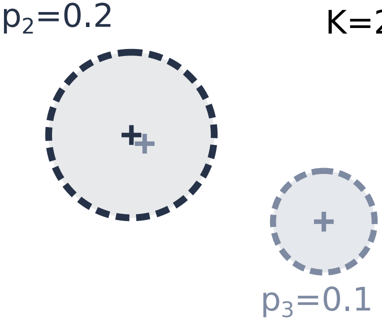


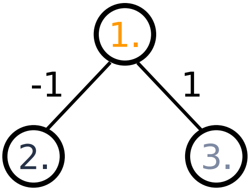
Adaptive sensor activation problem.

- Upon activation, each sensor i cover its area with probability p_i and the state of the sensor is **observed**.
- **Goal:** Maximize the total **expected** covered area with 2 sensors, **chosen adaptively**.

Good Policies:

L=3
K=2

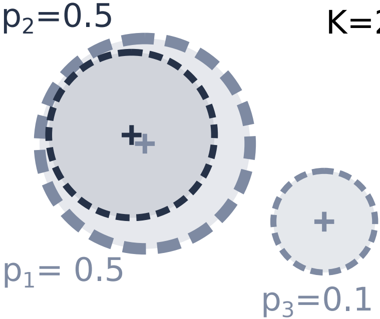




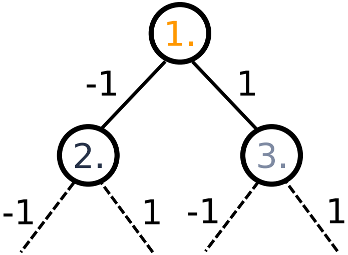
Adaptive **sensor activation problem.**

- Upon activation, each sensor i cover its area with probability p_i and the state of the sensor is **observed**.
- **Goal:** Maximize the total **expected** covered area with 2 sensors, **chosen adaptively**.

L=3
K=2

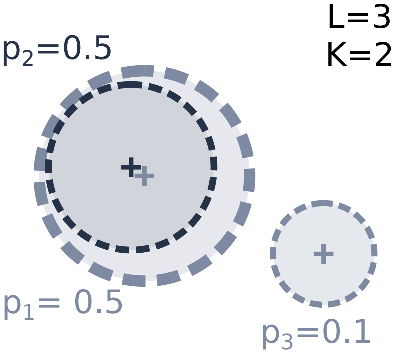


Good Policies:

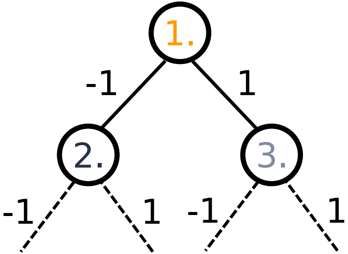


Adaptive sensor activation problem.

- Upon activation, each sensor i cover its area with probability p_i and the state of the sensor is **observed**.
- **Goal:** Maximize the total **expected** covered area with 2 sensors, **chosen adaptively**.



Good Policies:



Greedy is $(1-1/e)-$optimal

Adaptive submodular maximization

The objective is to maximize a **real** function of the form:

$$f(\underbrace{A}_{\text{Set of selected items}}, \underbrace{\phi}_{\text{State of the } L \text{ items}}) \to \mathbb{R}$$

- $\phi \in \{-1, 1\}^L$, where $\phi[i]$ is the state of item i.
- ϕ is drawn i.i.d. from $P(\Phi)$.

An **observation** is a vector $\mathbf{y} \in \{-1, 0, 1\}^L$.

$$\begin{array}{lllllll} \phi & = (& -1, & -1, & 1, & -1, & 1 \quad) \leftarrow \text{State} \\ \mathbf{y} & = (& 0, & -1, & 0, & -1, & 1 \quad) \leftarrow \text{observation} \\ \mathbf{y}' & = (& 0, & 0, & 0, & -1, & 0 \quad) \leftarrow \text{another observation} \end{array}$$

We denote $\mathbf{y} \succeq \mathbf{y}'$ and $\phi \sim \mathbf{y}$.
The selected items of $\mathbf{y}$ are $\mathrm{dom}(\mathbf{y}) = A = \{2, 4, 5\}$.

Adaptive submodular maximization

- a *policy* $\pi : \{-1, 0, 1\}^L \to \{1, \dots, L\}$
- $\pi_k(\phi)$ are the first k items chosen by policy π in state ϕ.

The optimal **K**-step policy satisfies:

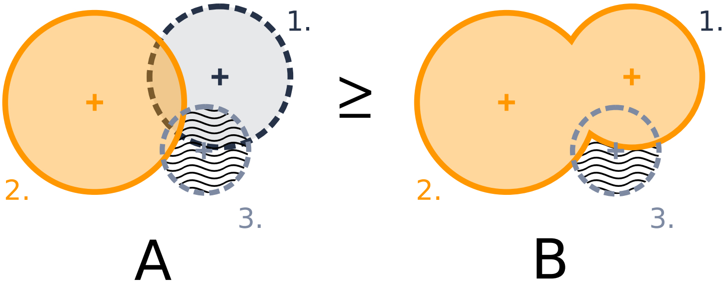
$$\pi^*_K = \arg\max_{\pi_K} \mathbb{E}_\phi[f(\pi_K(\phi), \phi)]\,.$$

Computing π^*_K is NP-hard. **BUT**, if the function is adaptive submodular and adaptive monotonic...

f is adaptive submodular if:

$$\mathbb{E}_\phi[\, f(A \cup \{i\}, \phi) - f(A, \phi) \,|\, \phi \sim \mathbf{y}_A \,] \geq \mathbb{E}_\phi[\, f(B \cup \{i\}, \phi) - f(B, \phi) \,|\, \phi \sim \mathbf{y}_B \,]$$

$i \in I \setminus B$ and $\mathbf{y}_B \succeq \mathbf{y}_A$, where $A = \mathrm{dom}(\mathbf{y}_A)$ and $B = \mathrm{dom}(\mathbf{y}_B)$.



Assumptions

f is adaptive submodular if:

$$\mathbb{E}_\phi[\, f(A \cup \{i\}, \phi) - f(A, \phi) \,|\, \phi \sim \mathbf{y}_A \,] \\ \geq \mathbb{E}_\phi[\, f(B \cup \{i\}, \phi) - f(B, \phi) \,|\, \phi \sim \mathbf{y}_B \,]$$

$i \in I \setminus B$ and $\mathbf{y}_B \succeq \mathbf{y}_A$, where $A = \mathrm{dom}(\mathbf{y}_A)$ and $B = \mathrm{dom}(\mathbf{y}_B)$.

f is adaptive monotonic if

$$\mathbb{E}_\phi[\, f(A \cup \{i\}, \phi) - f(A, \phi) \,|\, \phi \sim \mathbf{y}_A \,] \quad \geq \mathbf{0}$$

$i \in I \setminus A$ and $\mathbf{y}_A$, where $A = \mathrm{dom}(\mathbf{y}_A)$.

The greedy policy π^g

π^g always selects the item with the highest expected gain:

$$\pi^g(\mathbf{y}) = \arg\max_{i \in I \setminus \mathrm{dom}(\mathbf{y})} g_i(\mathbf{y}),$$

where:

$$g_{\mathbf{i}}(\mathbf{y}) = \mathbb{E}_\phi[\, f(\mathrm{dom}(\mathbf{y}) \cup \{\mathbf{i}\}, \phi) - f(\mathrm{dom}(\mathbf{y}), \phi) \,|\, \phi \sim \mathbf{y}\,]$$

is the *expected gain* of choosing item i after observing $\mathbf{y}$.

- π^g is **simple**.
- Then π^g is a $(1 - 1/e)$-approximation to π^*.

The greedy policy π^g

π^g always selects the item with the highest expected gain:

$$\pi^g(\mathbf{y}) = \arg\max_{i \in I \setminus \mathrm{dom}(\mathbf{y})} g_i(\mathbf{y}),$$

where:

$$g_{\mathbf{i}}(\mathbf{y}) = \mathbb{E}_\phi[\, f(\mathrm{dom}(\mathbf{y}) \cup \{\mathbf{i}\}, \phi) - f(\mathrm{dom}(\mathbf{y}), \phi) \mid \phi \sim \mathbf{y}\,]$$

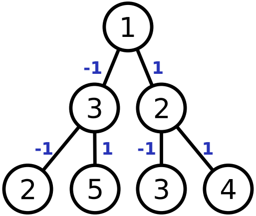
is the *expected gain* of choosing item i after observing $\mathbf{y}$.

- π^g is **simple**.
- Then π^g is a $(1 - 1/e)$-approximation to π^*.

Our approach

Our approach

Trying to **mimic** π^g while **learning** Φ.



What model for Φ?

Trade-off between

- Simple model for Φ $\rightarrow$ Easy to learn
- Complex model $\rightarrow$ More realistic.

Two models

- Assuming independence

- Modelling the dependencies.

Two models

- Assuming independence

- Modelling the dependencies.

Independence assumption

The states of one item is independent of the others.

$$\phi = \{Bernoulli(p_1), Bernoulli(p_2), \ldots, Bernoulli(p_L)\}$$

$\rightarrow$ Only L parameters to learn.

We define the expected gain as:

$$g_i(\mathbf{y}) = p_i \bar{g}_i(\mathbf{y}),$$

$\bar{g}_i(\mathbf{y})$ = gain associated with state 1. (area covered if sensor is active)

Independence assumption

The states of one item is independent of the others.

$$\phi = \{Bernoulli(p_1), Bernoulli(p_2), \ldots, Bernoulli(p_L)\}$$

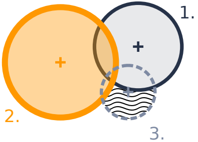
$\rightarrow$ Only L parameters to learn.

We define the expected gain as:

$$g_i(\mathbf{y}) = p_i \bar{g}_i(\mathbf{y}),$$

$\bar{g}_i(\mathbf{y})$ = gain associated with state 1. (area covered if sensor is active)

- We assume is easy to compute.
- Might not even be an expectation.



The framework & algorithm

We repetitively play the **K** step game.

Input: States $\phi_1, \ldots, \phi_n$

for $t = 1, 2, \ldots, \mathbf{n}$ **do** ◃ *n episodes*

1. **Play the K-step game with** π^t

2. **Update all statistics of the model**

end for

We try to design π^t in order to minimize the **cumulative regret**

$$R(n) = \mathbb{E}_{\phi_1,\ldots,\phi_{\mathbf{n}}}\left[\sum_{t=1}^{\mathbf{n}} f(\pi^g_{\mathbf{K}}(\phi_t), \phi_t) - f(\pi^t_{\mathbf{K}}(\phi_t), \phi_t)\right].$$

The framework & algorithm

We repetitively play the **K** step game.

Input: States $\phi_1, \ldots, \phi_n$

for $t = 1, 2, \ldots, \mathbf{n}$ **do** $\triangleleft$ *n episodes*

1. **Play the K-step game with** π^t

   where $\pi^t(\mathbf{y}) = \arg\max_i \widehat{g}_i(\mathbf{y})$ *(upper bound on $g_i(\mathbf{y})$)*

2. **Update all statistics of the model**

end for

We try to design π^t in order to minimize the **cumulative regret**

$$R(n) = \mathbb{E}_{\phi_1,\ldots,\phi_{\mathbf{n}}}\left[\sum_{t=1}^{\mathbf{n}} f(\pi^g_{\mathbf{K}}(\phi_t), \phi_t) - f(\pi^t_{\mathbf{K}}(\phi_t), \phi_t)\right].$$

The framework & algorithm

We repetitively play the **K** step game.

Input: States $\phi_1, \dots, \phi_n$

for $t = 1, 2, \dots, \mathbf{n}$ **do** $\triangleleft$ *n episodes*

1. **Play the K-step game with** π^t

   where $\pi^t(\mathbf{y}) = \arg\max_i \left(\widehat{p}_i(t) + \sqrt{\frac{1}{T_i(t)}}\right) \bar{g}_i(\mathbf{y})$

2. **Update all statistics of the model**

end for

We try to design π^t in order to minimize the **cumulative regret**

$$R(n) = \mathbb{E}_{\phi_1,\dots,\phi_{\mathbf{n}}}\left[\sum_{t=1}^{\mathbf{n}} f(\pi^g_{\mathbf{K}}(\phi_t), \phi_t) - f(\pi^t_{\mathbf{K}}(\phi_t), \phi_t)\right].$$

Bound on the regret

The gap: $\Delta_i(\mathbf{y}) = g_{i^*(\mathbf{y})}(\mathbf{y}) - g_i(\mathbf{y})$.

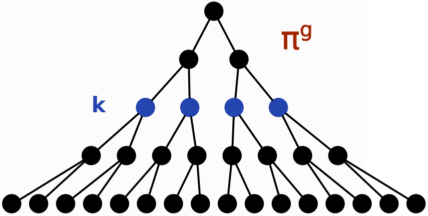
Theorem

$$R(n) \leq \underbrace{\sum_{i=1}^{L} \ell_i \sum_{k=1}^{K} G_k \alpha_{i,k}}_{O(\log n)} + O(1),$$

ℓ_i = Max # of wrong pulls of i

$\ell_i = \max_k \ell_{i,k}$ *and* $\sum_{k=1}^{K} \alpha_{i,k} = 1$

fdg



Bound on the regret

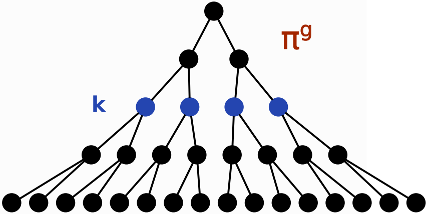
The gap: $\Delta_i(\mathbf{y}) = g_{i^*(\mathbf{y})}(\mathbf{y}) - g_i(\mathbf{y})$.

Theorem

$$R(n) \leq \underbrace{\sum_{i=1}^{L} \ell_i \sum_{k=1}^{K} G_k \alpha_{i,k}}_{O(\log n)} + O(1),$$

$\ell_{i,k} = 8 \max_{\mathbf{y} \in \mathcal{Y}_{k,i}} \frac{\bar{g}_i^2(\mathbf{y})}{\Delta_i^2(\mathbf{y})} \log n + 1,$

$\ell_i = \max_k \ell_{i,k}$ *and* $\sum_{k=1}^{K} \alpha_{i,k} = 1$



Bound on the regret

The gap: $\Delta_i(\mathbf{y}) = g_{i^*(\mathbf{y})}(\mathbf{y}) - g_i(\mathbf{y})$.

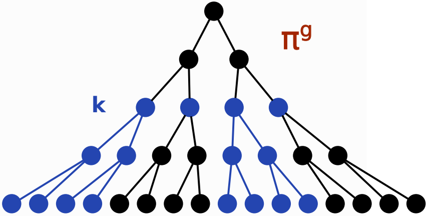
Theorem

$$R(n) \leq \underbrace{\sum_{i=1}^{L} \ell_i \sum_{k=1}^{K} G_k \alpha_{i,k}}_{O(\log n)} + O(1),$$

$\ell_{i,k} = 8 \max_{\mathbf{y} \in \mathcal{Y}_{k,i}} \frac{\bar{g}_i^2(\mathbf{y})}{\Delta_i^2(\mathbf{y})} \log n + 1,$

$\ell_i = \max_k \ell_{i,k}$ *and* $\sum_{k=1}^{K} \alpha_{i,k} = 1$

$G_k \geq$ *expected gain of* π^g *after level* k



Bound on the regret

The gap: $\Delta_i(\mathbf{y}) = g_{i^*(\mathbf{y})}(\mathbf{y}) - g_i(\mathbf{y})$.

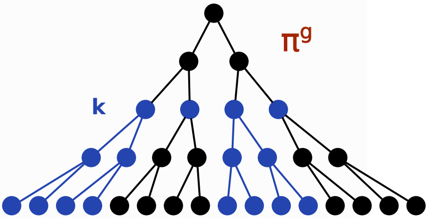
Theorem

$$R(n) \leq \underbrace{\sum_{i=1}^{L} \ell_i \sum_{k=1}^{K} G_k \alpha_{i,k}}_{O(\log n)} + O(1),$$

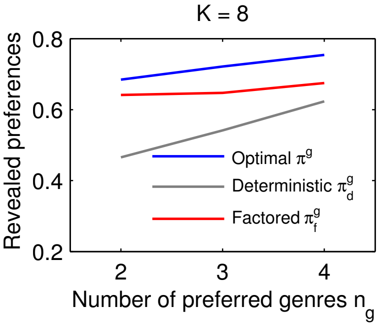
$\ell_{i,k} = 8 \max_{\mathbf{y} \in \mathcal{Y}_{k,i}} \frac{\bar{g}_i^2(\mathbf{y})}{\Delta_i^2(\mathbf{y})} \log n + 1,$

$\ell_i = \max_k \ell_{i,k}$ *and* $\sum_{k=1}^{K} \alpha_{i,k} = 1$

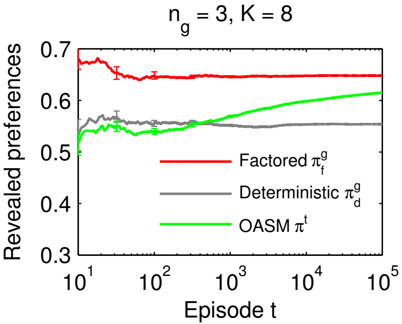
$G_k = (K - k + 1) \max_{\mathbf{y} \in \mathcal{Y}_k} \max_i g_i(\mathbf{y})$



Experiments on Movie-Lens



Offline



Online

Thank you!