Dispersion for Data-Driven Algorithm Design, Online Learning, and Private Optimization

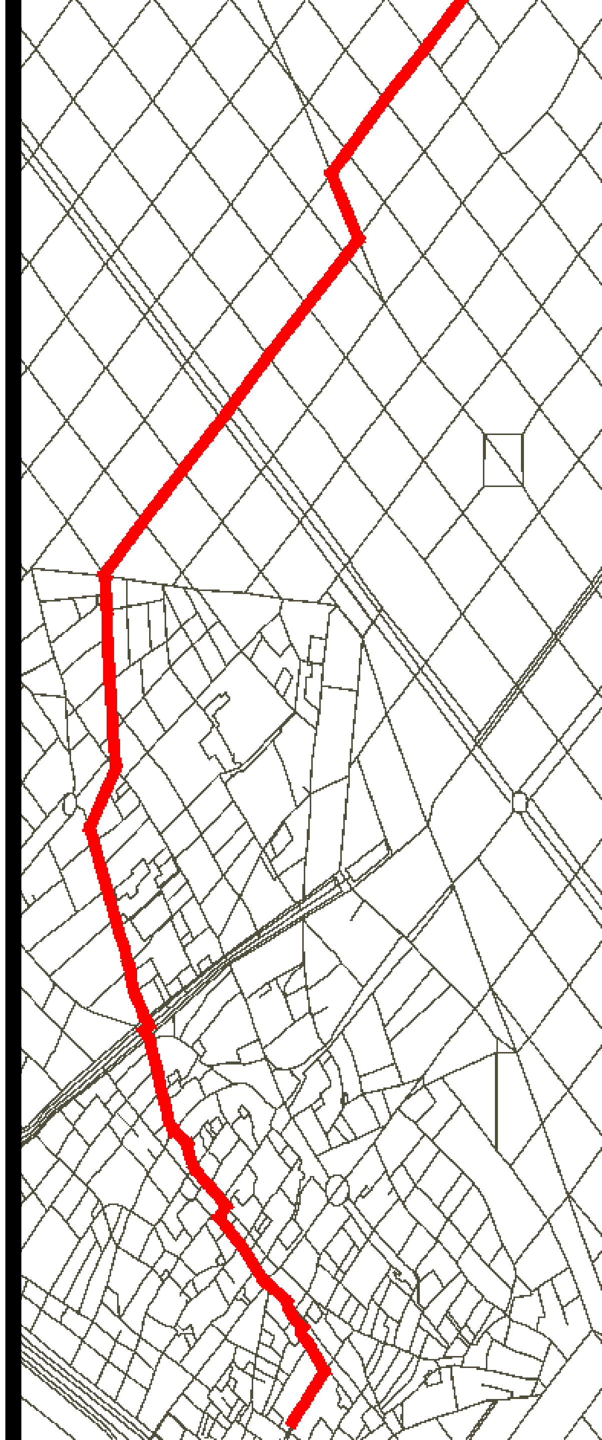
Ellen Vitercik

Northwestern Quarterly Theory Workshop

Joint work with Nina Balcan and Travis Dick

Many problems have fast, optimal algorithms

- E.g., sorting, shortest paths

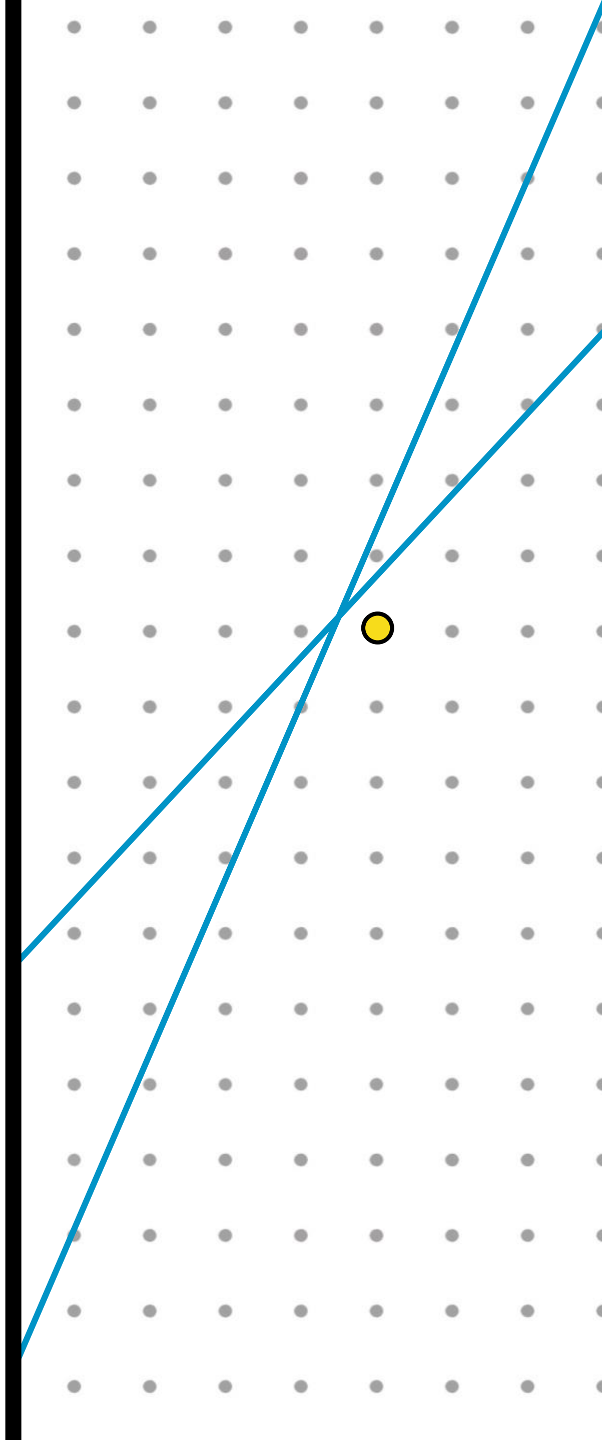


Many problems have fast, optimal algorithms

- E.g., sorting, shortest paths

Many problems don't

- E.g., integer programming, subset selection
- Many approximation and heuristic techniques
- Best method depends on the application
  - Which to use?

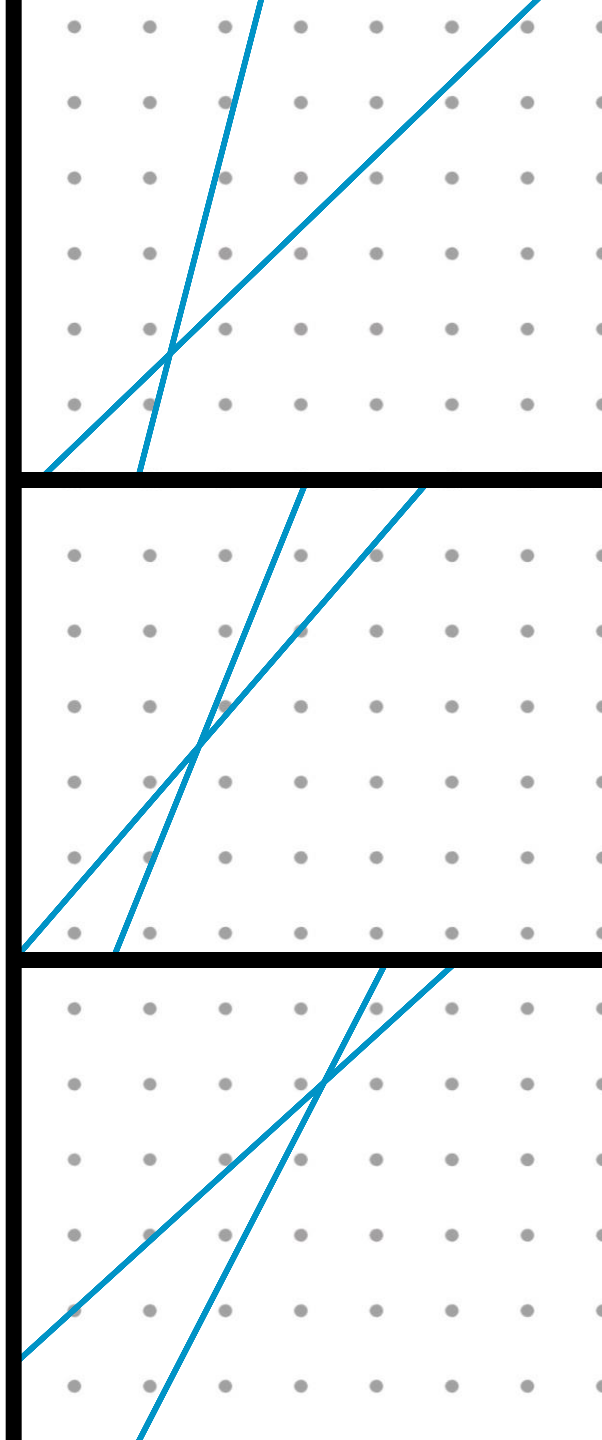


Practitioners repeatedly solve problems

- Maintain same structure
- Differ on underlying data

Should be algo that’s good across all instances

Use ML to automate algorithm design



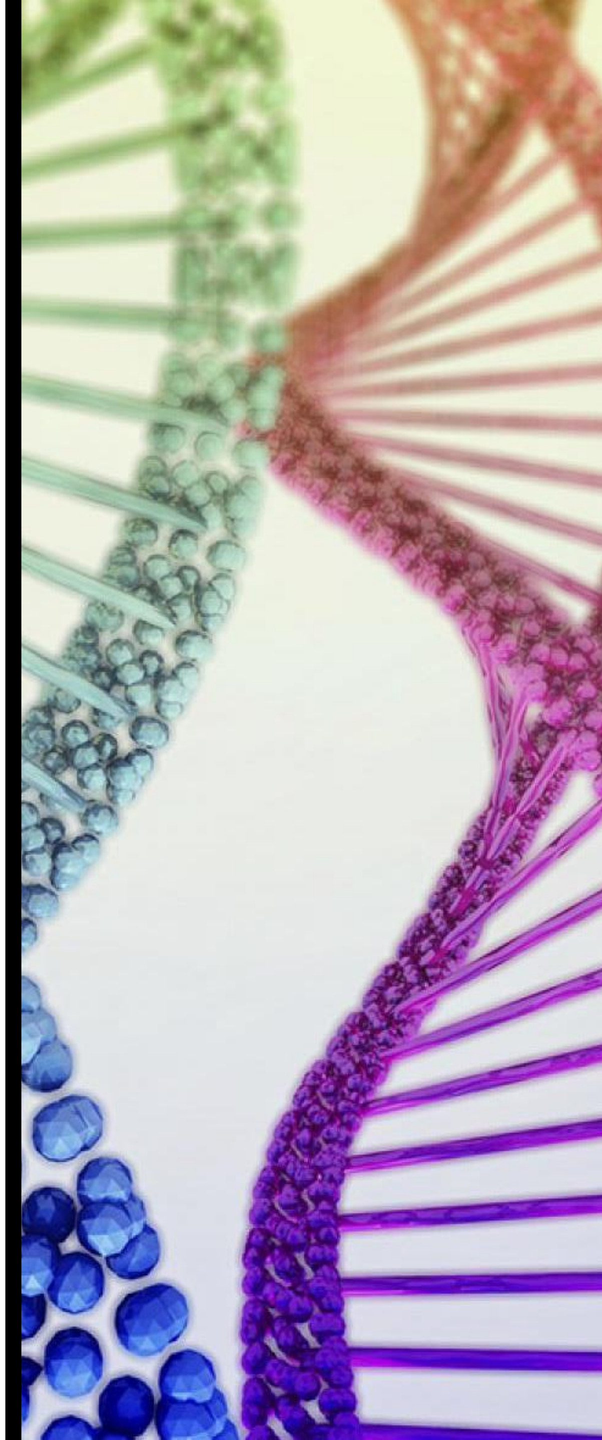
Automated algorithm design

Use ML to automate algorithm design

Large body of empirical work:

- Comp bio [DeBlasio and Kececioglu, ‘18]
- AI [Xu, Hutter, Hoos, and Leyton-Brown, ’08]

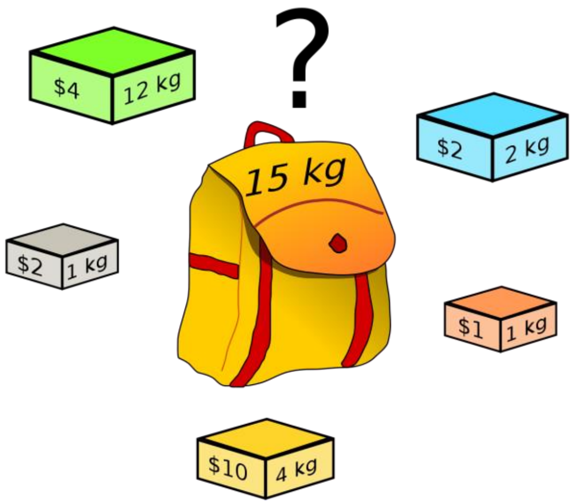
This work: formal guarantees for this approach



Simple example: knapsack

Problem instance:

- n items; Item i has value v_i and size s_i
- Knapsack with capacity K



Goal: find most valuable items that fit

Algorithm (parameterized by $\rho \geq 0$)**:**

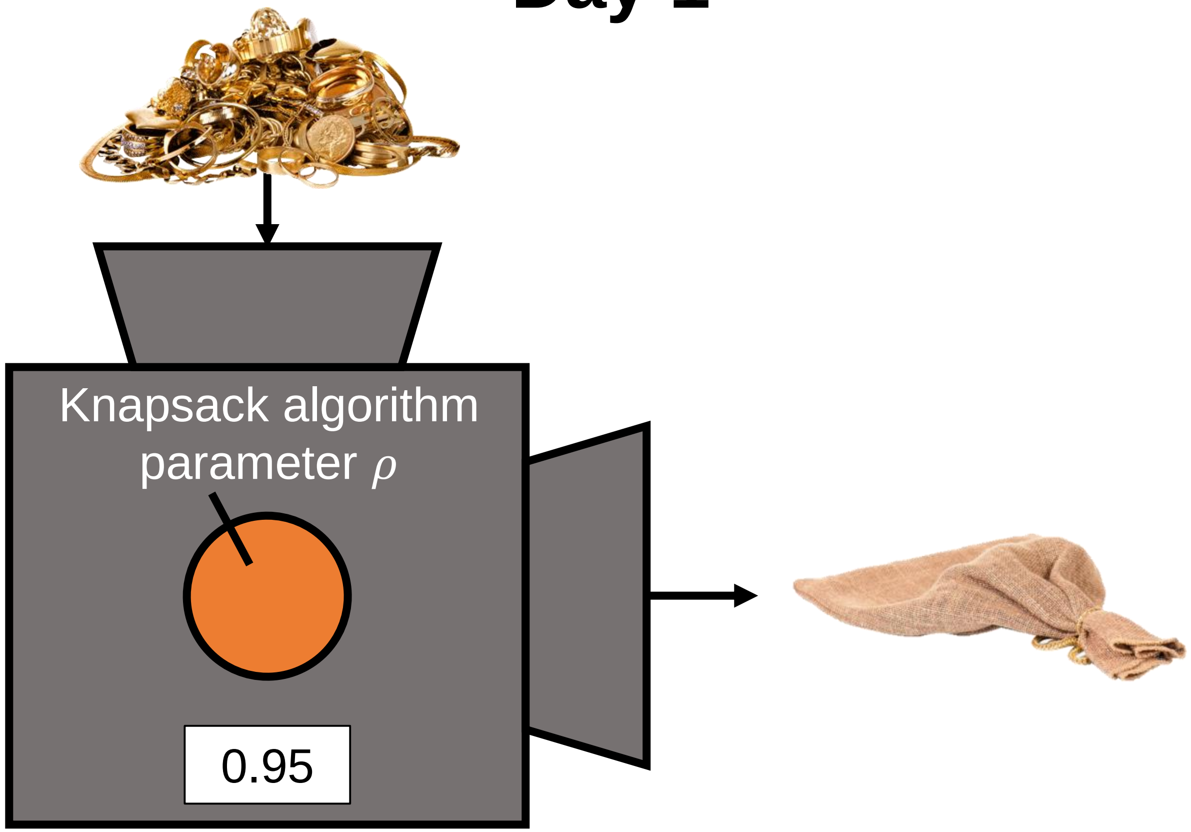
Add items in decreasing order of $\frac{v_i}{s_i^\rho}$ ← How to set?

[Gupta and Roughgarden, ‘17]

Application domain: stealing jewelry

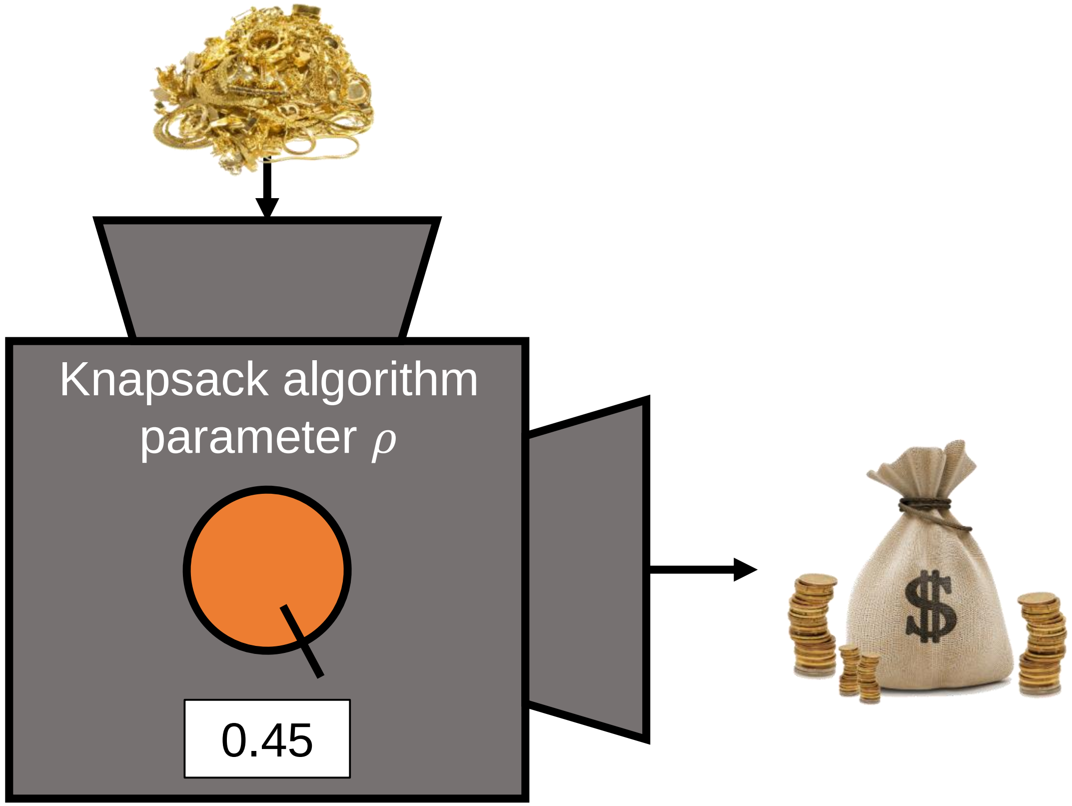
Online algorithm configuration

Day 1



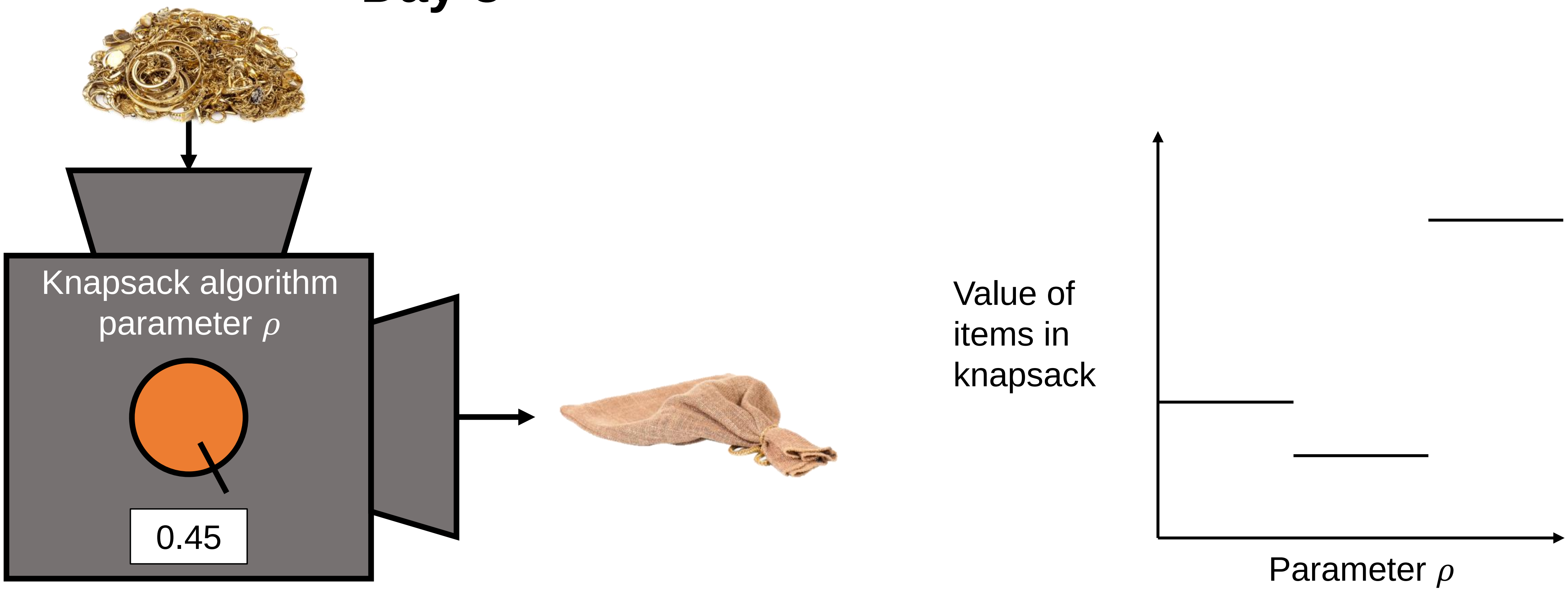
Online algorithm configuration

Day 2



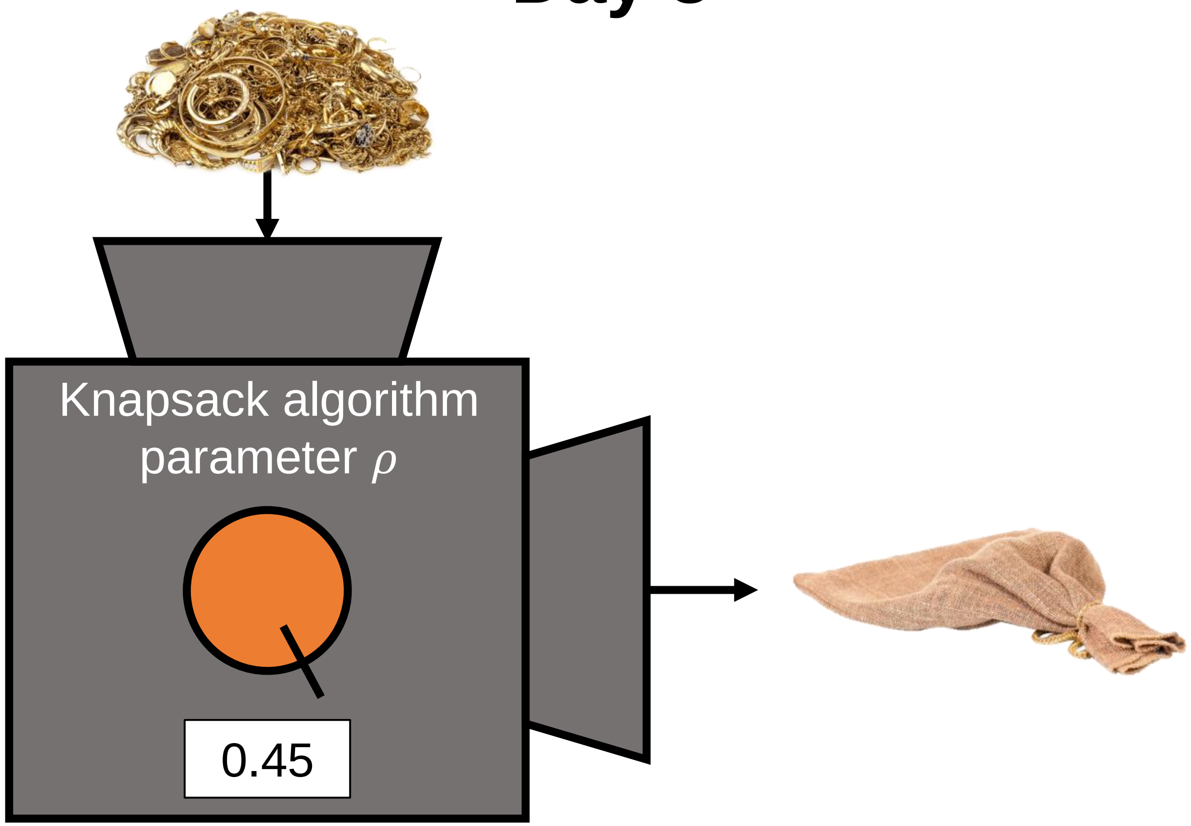
Online algorithm configuration

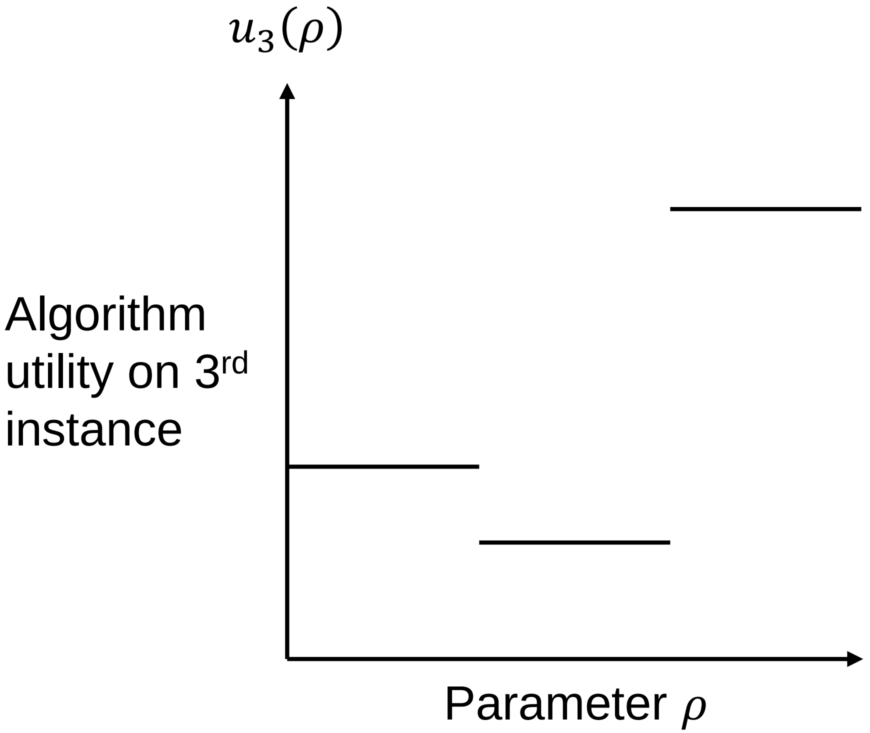
Day 3



Online algorithm configuration

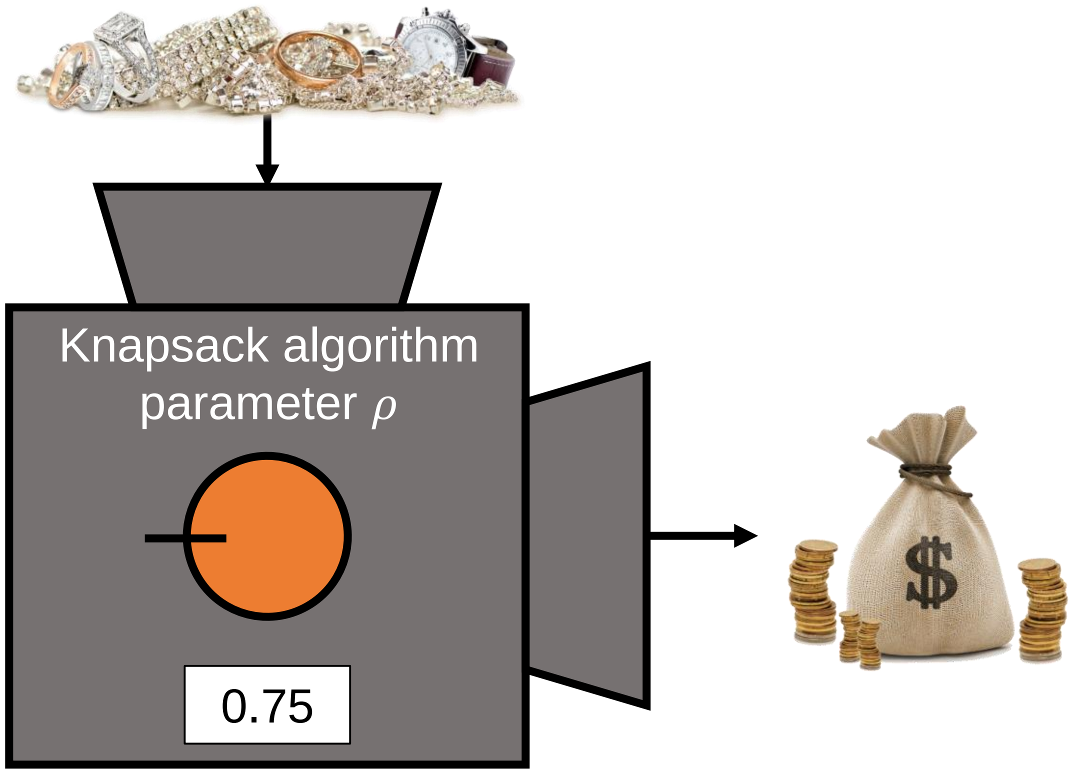
Day 3

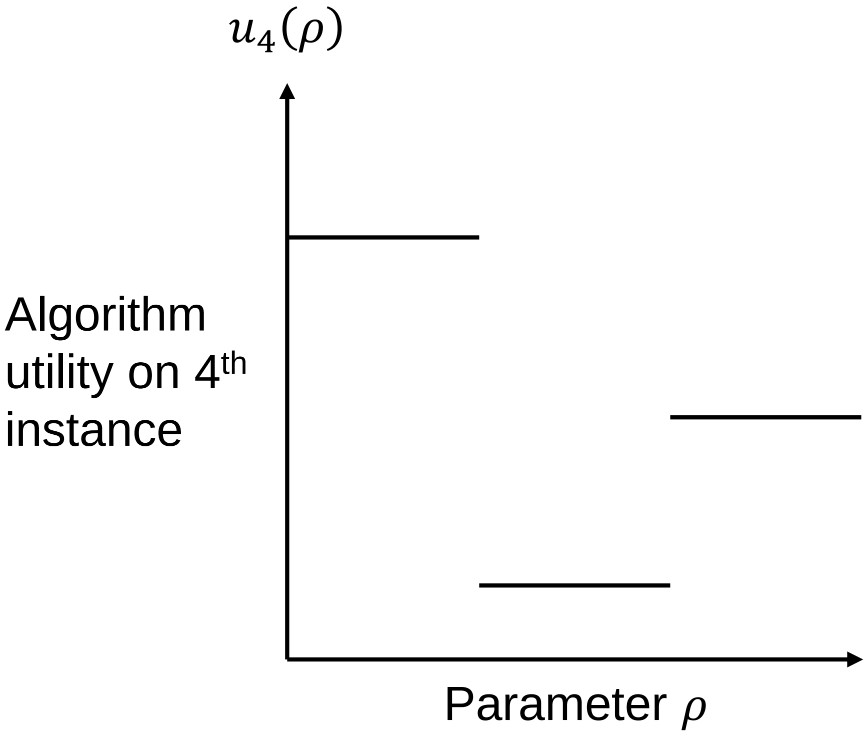




Online algorithm configuration

Day 4





Online algorithm configuration

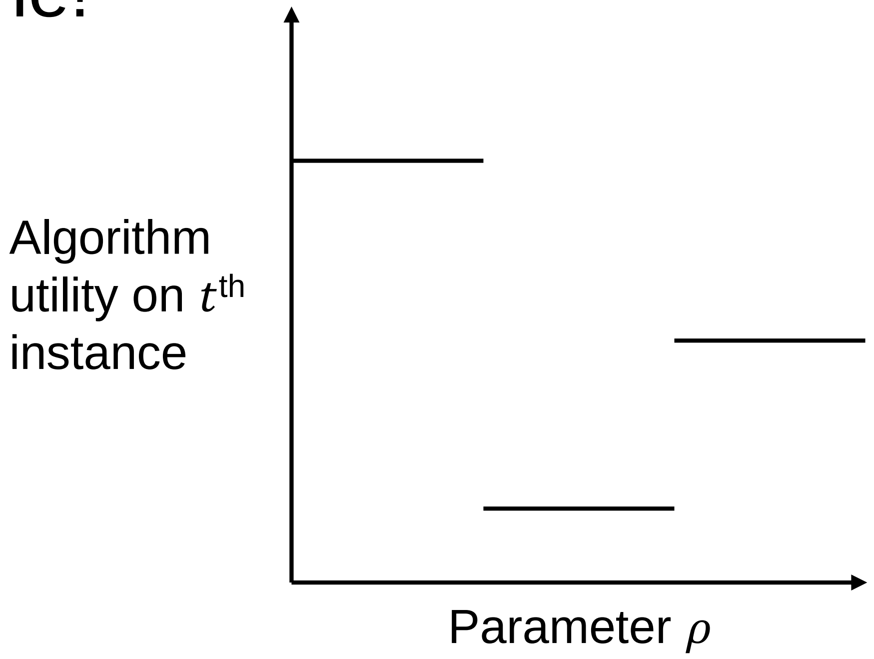
Goal: Compete with best fixed parameters in hindsight.

*Minimize **regret**.*

Optimizing piecewise Lipschitz functions

Configuration ⇔ optimizing sums of piecewise Lipschitz functions

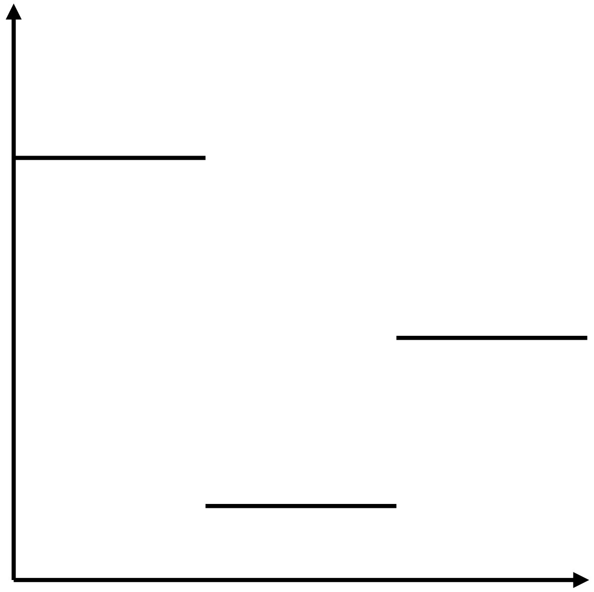
Worst-case **impossible** to optimize online!



Our contributions

Structural property ***dispersion*** implies strong guarantees for:

- Online optimization of PWL functions
- Uniform convergence in statistical settings
- Differentially private optimization



Dispersion satisfied in real problems
under very mild assumptions

Outline

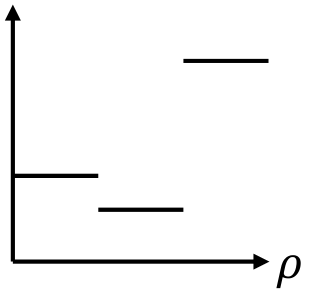
1. **Online learning setup**
2. Dispersion
3. Regret bounds
4. Examples of dispersion
5. Other applications of dispersion
6. Conclusion

Online piecewise Lipschitz optimization

For each round $t \in \{1, \ldots, T\}$:

1. Learner chooses $\boldsymbol{\rho}_t \in \mathbb{R}^d$
2. Adversary chooses piecewise L-Lipschitz function $u_t: \mathbb{R}^d \to \mathbb{R}$
3. Learner gets reward $u_t(\boldsymbol{\rho}_t)$
4. **Full information:** Learner observes function u_t

$u_t(\rho) =$ Algorithm utility on t^{th} instance



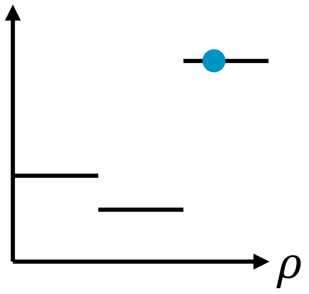
Online piecewise Lipschitz optimization

For each round $t \in \{1, \dots, T\}$:

1. Learner chooses $\boldsymbol{\rho}_t \in \mathbb{R}^d$
2. Adversary chooses piecewise L-Lipschitz function $u_t: \mathbb{R}^d \to \mathbb{R}$
3. Learner gets reward $u_t(\boldsymbol{\rho}_t)$
4. **Full information:** Learner observes function u_t

   Bandit feedback: Learner only observes $u_t(\boldsymbol{\rho}_t)$

$u_t(\rho) =$ Algorithm utility on t^{th} instance



Online piecewise Lipschitz optimization

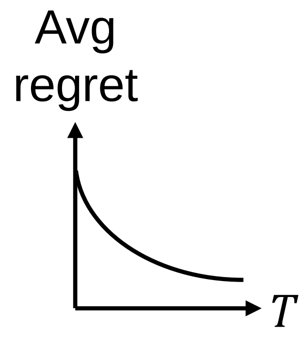
For each round $t \in \{1, \dots, T\}$:

1. Learner chooses $\boldsymbol{\rho}_t \in \mathbb{R}^d$
2. Adversary chooses piecewise L-Lipschitz function $u_t: \mathbb{R}^d \to \mathbb{R}$
3. Learner gets reward $u_t(\boldsymbol{\rho}_t)$
4. **Full information:** Learner observes function u_t

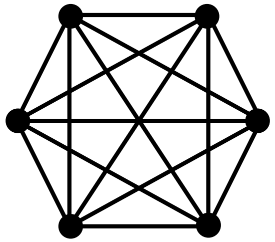
   Bandit feedback: Learner only observes $u_t(\boldsymbol{\rho}_t)$

Goal: Minimize regret $= \max_{\boldsymbol{\rho} \in \mathbb{R}^d} \sum_{t=1}^{T} u_t(\boldsymbol{\rho}) - \sum_{t=1}^{T} u_t(\boldsymbol{\rho}_t)$

Want regret sublinear in T

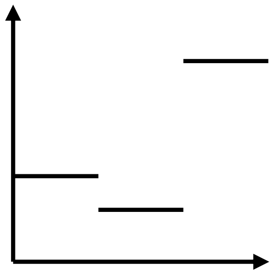


Prior work on PWL online optimization



Gupta and Roughgarden ['17]:

Max-Weight Independent Set algo configuration



Cohen-Addad and Kanade ['17]:

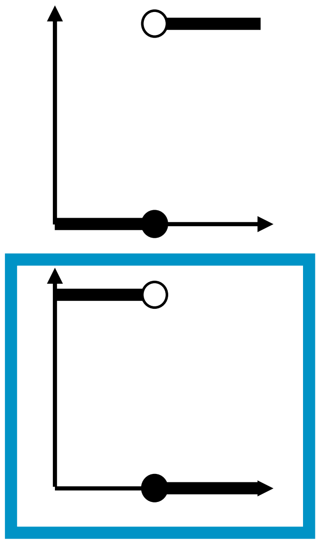
1D piecewise constant functions

Mean adversary

Exists adversary choosing piecewise constant functions s.t.:

Every full information online algorithm has **linear regret**.

Round 1:



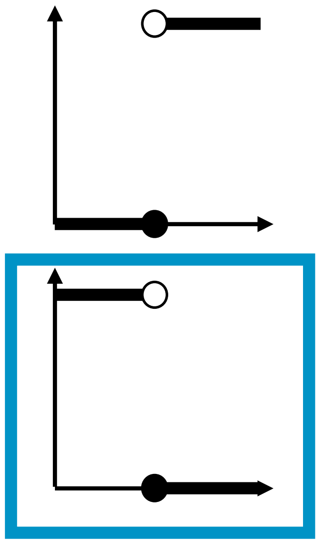
Adversary chooses one or the other with equal prob.

Mean adversary

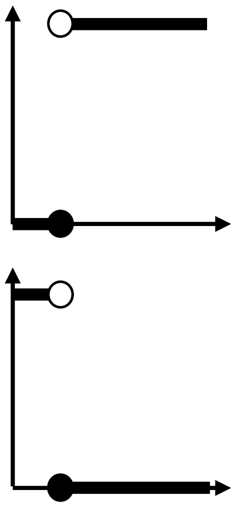
Exists adversary choosing piecewise constant functions s.t.:

Every full information online algorithm has **linear regret**.

Round 1:



Round 2:

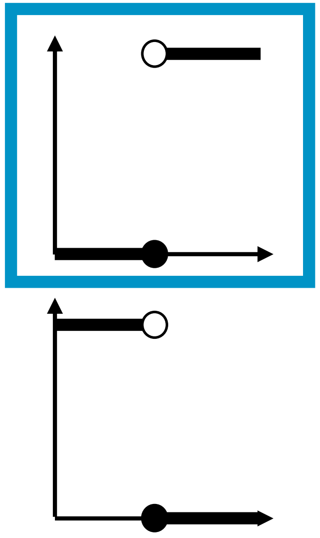


Mean adversary

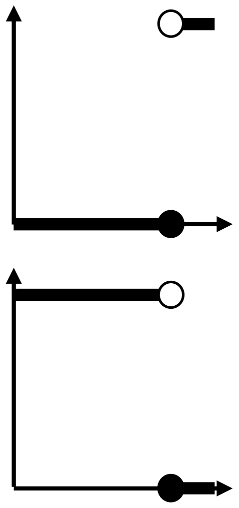
Exists adversary choosing piecewise constant functions s.t.:

Every full information online algorithm has **linear regret**.

Round 1:



Round 2:



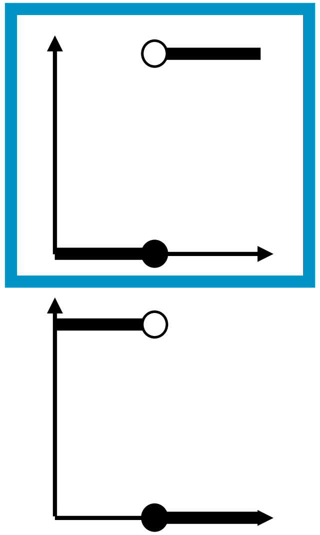
Repeatedly halves optimal region

Mean adversary

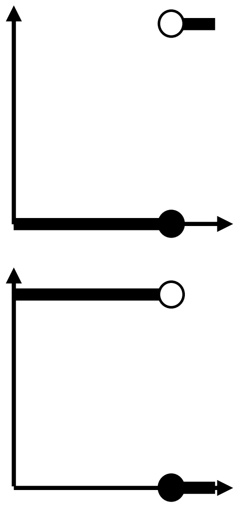
Exists adversary choosing piecewise constant functions s.t.:

Every full information online algorithm has **linear regret**.

Round 1:



Round 2:



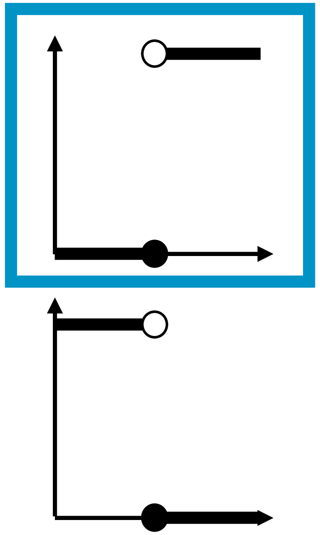
Repeatedly halves optimal region

Mean adversary

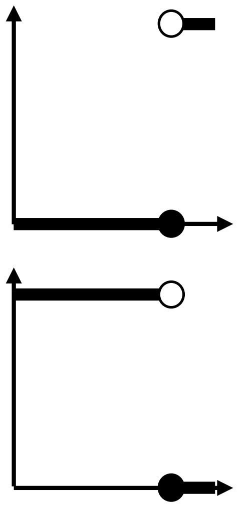
Exists adversary choosing piecewise constant functions s.t.:

Every full information online algorithm has **linear regret**.

Round 1:



Round 2:



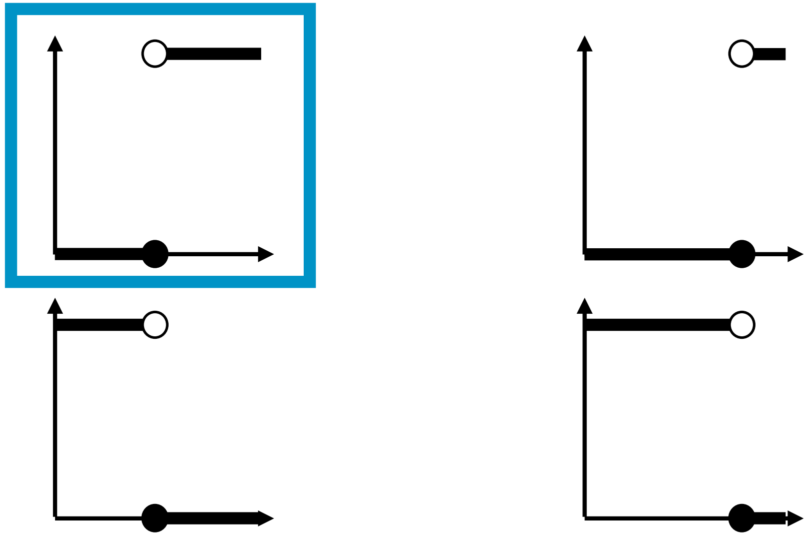
Repeatedly halves optimal region

Mean adversary

Exists adversary choosing piecewise constant functions s.t.:

Every full information online algorithm has **linear regret**.

Round 1: Round 2:



Repeatedly halves optimal region

Learner's expected reward: $\frac{T}{2}$

Reward of best point in hindsight: T

Expected regret = $\frac{T}{2}$

Outline

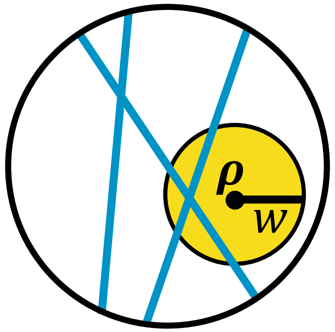
1. Online learning setup
2. **Dispersion**
3. Regret bounds
4. Examples of dispersion
5. Other applications of dispersion
6. Conclusion

Dispersion

Mean adversary concentrates discontinuities near maximizer ρ^*
Even points very close to ρ^* have low utility!

$u_1, \ldots, u_T$ are ***($\boldsymbol{w}, \boldsymbol{k}$)-dispersed at point $\boldsymbol{\rho}$*** if:

ℓ_2-ball $B(\boldsymbol{\rho}, w)$ contains discontinuities for $\leq k$ of $u_1, \ldots, u_T$

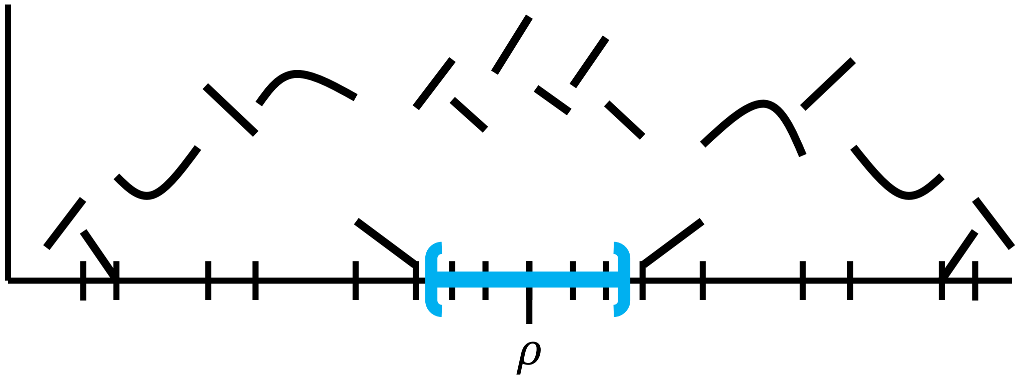


Ball of radius w about $\boldsymbol{\rho}$ contains 2 discontinuities.
→ $(w, 2)$-dispersed at $\boldsymbol{\rho}$.

Sums of piecewise dispersed functions

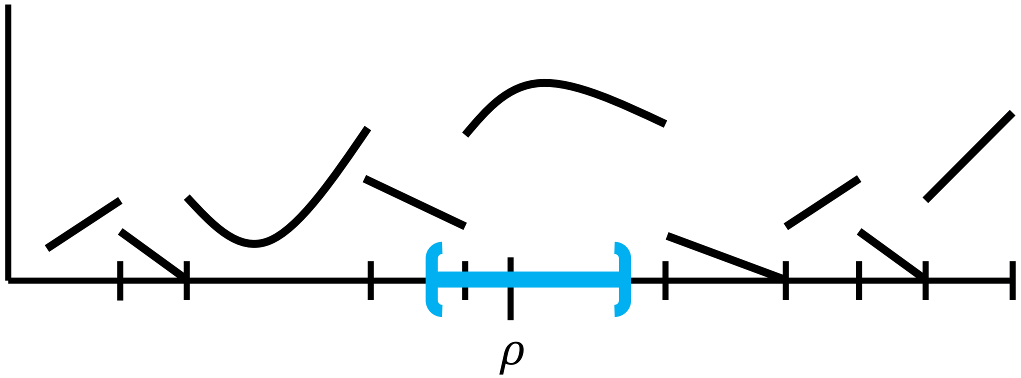
Given $u_1, \dots, u_T$, plot of sum $\sum_{t=1}^{T} u_t$:

Not dispersed



Many discontinuities in interval

Dispersed



Few discontinuities in interval

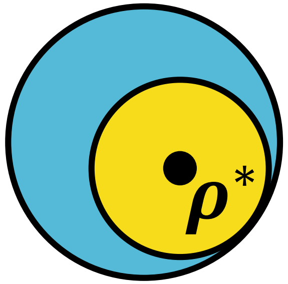
Key property of dispersed functions

If $u_1, \ldots, u_T: \mathbb{R}^d \to [0,1]$ are

1. Piecewise L-Lipschitz
2. (w, k)-dispersed at maximizer $\boldsymbol{\rho}^*$,

For every $\boldsymbol{\rho} \in B(\boldsymbol{\rho}^*, w)$: $\sum_{t=1}^{T} u_t(\boldsymbol{\rho}) \geq \sum_{t=1}^{T} u_t(\boldsymbol{\rho}^*) - TLw - k$.

Proof idea : $u_1, \ldots, u_T$



Is u_t L-Lipschitz on $B(\boldsymbol{\rho}^*, w)$?

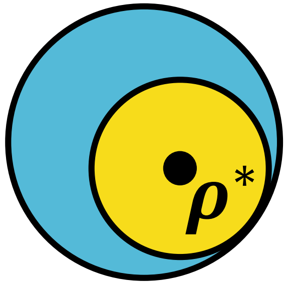
Key property of dispersed functions

If $u_1, \ldots, u_T : \mathbb{R}^d \to [0,1]$ are

1. Piecewise L-Lipschitz
2. (w, k)-dispersed at maximizer $\boldsymbol{\rho}^*$,

For every $\boldsymbol{\rho} \in B(\boldsymbol{\rho}^*, w)$: $\sum_{t=1}^{T} u_t(\boldsymbol{\rho}) \geq \sum_{t=1}^{T} u_t(\boldsymbol{\rho}^*) - TLw - \boldsymbol{k}$.

Proof idea : $u_1, \ldots, u_T$



Is u_t L-Lipschitz on $B(\boldsymbol{\rho}^*, w)$?

No $(\leq k$ functions) → $|u_t(\boldsymbol{\rho}) - u_t(\boldsymbol{\rho}^*)| \leq 1$

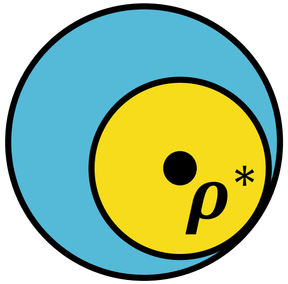
Key property of dispersed functions

If $u_1, \ldots, u_T : \mathbb{R}^d \to [0,1]$ are

1. Piecewise L-Lipschitz
2. (w, k)-dispersed at maximizer $\boldsymbol{\rho}^*$,

For every $\boldsymbol{\rho} \in B(\boldsymbol{\rho}^*, w)$: $\sum_{t=1}^{T} u_t(\boldsymbol{\rho}) \geq \sum_{t=1}^{T} u_t(\boldsymbol{\rho}^*) - \boldsymbol{TLw} - k$.

Proof idea : $u_1, \ldots, u_T$



Is u_t L-Lipschitz on $B(\boldsymbol{\rho}^*, w)$?

$|u_t(\boldsymbol{\rho}) - u_t(\boldsymbol{\rho}^*)| \leq 1$

No (≤ k functions)

Yes (≤ T functions)

$|u_t(\boldsymbol{\rho}) - u_t(\boldsymbol{\rho}^*)| \leq Lw$

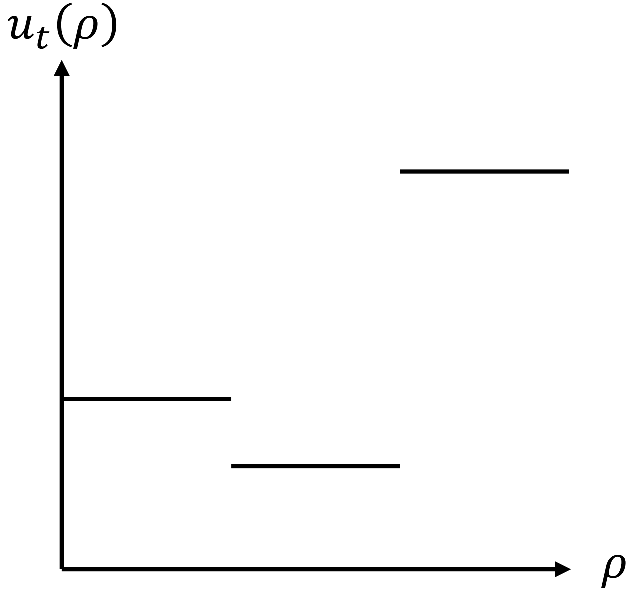
Outline

1. Online learning setup
2. Dispersion
3. Regret bounds
   1. **Full information**
   2. Bandit feedback
4. Examples of dispersion
5. Other applications of dispersion
6. Conclusion

Full information online learning

Exponentially Weighted Forecaster [Cesa-Bianchi & Lugosi ’06]:

At round t, sample from dist. w/ PDF $f_t(\boldsymbol{\rho}) \propto \exp(\lambda \sum_{s=1}^{t-1} u_s(\boldsymbol{\rho}))$.



Full information online learning

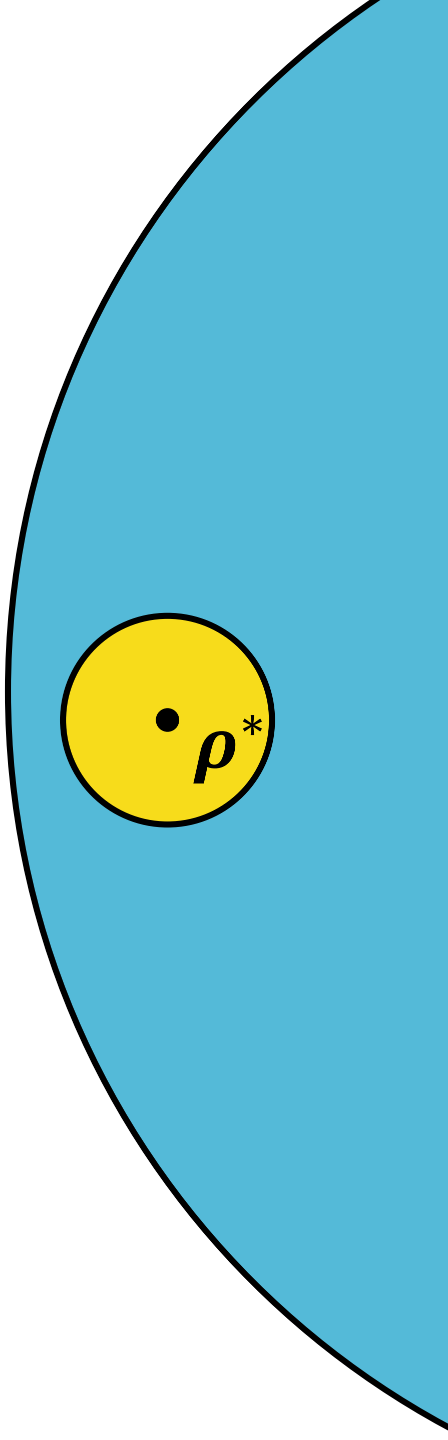
Theorem: If $u_1, \ldots, u_T : B_d(\mathbf{0}, 1) \to [0,1]$ are:

1. Piecewise L-Lipschitz
2. (w, k)-dispersed at $\boldsymbol{\rho}^*$,

EWF has regret $O\left(\sqrt{\boldsymbol{Td}\,\mathbf{log}\frac{\mathbf{1}}{\boldsymbol{w}}} + \boldsymbol{TLw} + \boldsymbol{k}\right)$.

When is this a good bound?

For $w = \frac{1}{L\sqrt{T}}$ and $k = \tilde{O}(\sqrt{T})$, regret is $\tilde{O}(\sqrt{Td})$



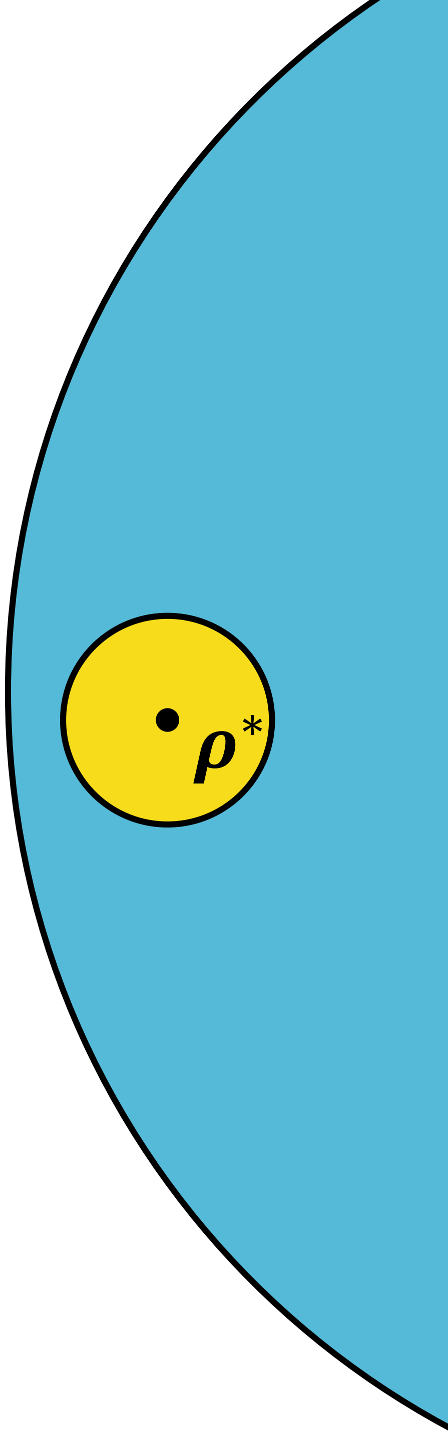
Full information online learning

Theorem: If $u_1, \ldots, u_T : B_d(\mathbf{0}, 1) \to [0,1]$ are:

1. Piecewise L-Lipschitz
2. (w, k)-dispersed at $\boldsymbol{\rho}^*$,

EWF has regret $O\left(\sqrt{\boldsymbol{Td}\,\mathbf{log}\,\frac{\mathbf{1}}{\boldsymbol{w}}} + \boldsymbol{TLw} + \boldsymbol{k}\right)$.

Intuition: Every $\boldsymbol{\rho} \in B(\boldsymbol{\rho}^*, w)$ has utility $\geq OPT - \boldsymbol{TLw} - \boldsymbol{k}$.



Full information online learning

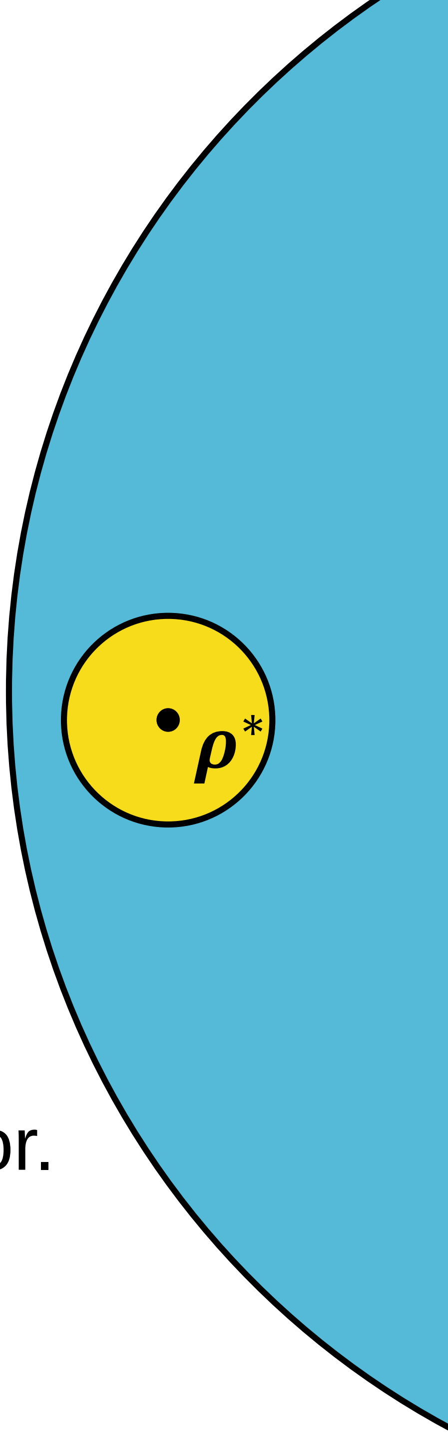
Theorem: If $u_1, \ldots, u_T: B_d(\mathbf{0}, 1) \to [0,1]$ are:

1. Piecewise L-Lipschitz
2. (w, k)-dispersed at $\boldsymbol{\rho}^*$,

EWF has regret $O\left(\sqrt{\boldsymbol{Td}\log\frac{\mathbf{1}}{\boldsymbol{w}}} + \boldsymbol{TLw} + \boldsymbol{k}\right)$.

Intuition: Every $\boldsymbol{\rho} \in B(\boldsymbol{\rho}^*, w)$ has utility $\geq OPT - \boldsymbol{TLw} - \boldsymbol{k}$.

EWF can compete with $B(\boldsymbol{\rho}^*, w)$ up to $O\left(\sqrt{\boldsymbol{Td}\log\frac{\mathbf{1}}{\boldsymbol{w}}}\right)$ factor.



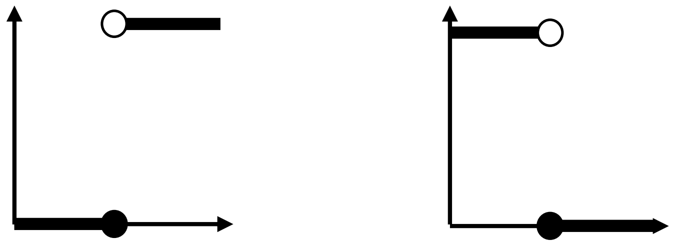
Matching lower bound

Theorem: For any algorithm, exist PW constant $u_1, \ldots, u_T$ s.t.:

Algorithm's regret is $\Omega\left(\inf_{(w,k)} \sqrt{Td \log \frac{1}{w}} + k\right)$.

Inf over all (w, k)-dispersion parameters $u_1, \ldots, u_T$ satisfy at $\boldsymbol{\rho}^*$.

Upper bound = $O\left(\inf_{(w,k)} \sqrt{Td \log \frac{1}{w}} + k\right)$.



Outline

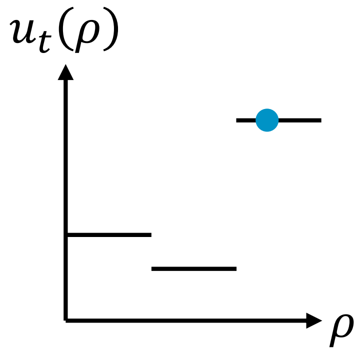
1. Online learning setup
2. Dispersion
3. Regret bounds
   1. Full information
   2. **Bandit feedback**
4. Examples of dispersion
5. Other applications of dispersion
6. Conclusion

Bandit feedback

Theorem: If $u_1, \ldots, u_T : B_d(\mathbf{0}, 1) \to [0,1]$ are:

1. Piecewise L-Lipschitz
2. (w, k)-dispersed at $\boldsymbol{\rho}^*$,

There is a bandit algorithm with regret $\tilde{O}\left(\sqrt{Td\left(\frac{1}{w}\right)^d} + TLw + k\right)$.

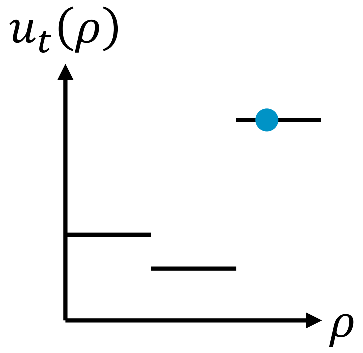


Bandit feedback

Theorem: Exists algorithm with regret $\tilde{O}\left(\sqrt{Td\left(\frac{1}{w}\right)^d} + TLw + k\right)$.

When is this a good bound?

If $d = 1$, $w = \frac{1}{\sqrt[3]{T}}$, and $k = \tilde{O}\left(T^{2/3}\right)$, regret is $\tilde{O}\left(LT^{2/3}\right)$.

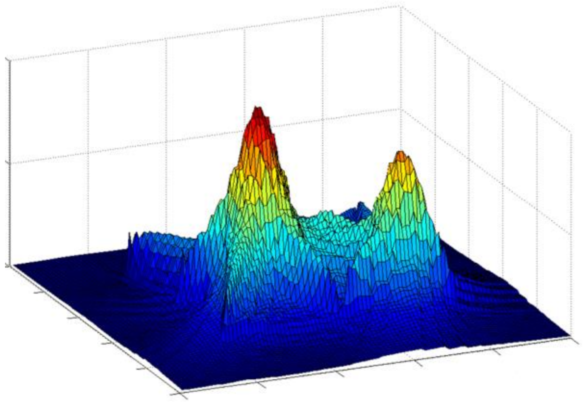


Bandit feedback

Theorem: Exists algorithm with regret $\tilde{O}\left(\sqrt{Td\left(\frac{1}{w}\right)^d} + TLw + k\right)$.

When is this a good bound?

If $w = T^{\frac{d+1}{d+2}-1}$, $k = \tilde{O}\left(T^{\frac{d+1}{d+2}}\right)$, then regret is $\tilde{O}\left(T^{\frac{d+1}{d+2}}\left(\sqrt{d3^d} + L\right)\right)$.

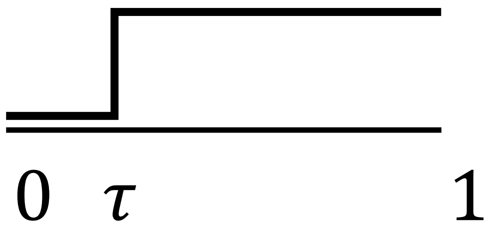


Outline

1. Online learning setup
2. Dispersion
3. Regret bounds
4. **Examples of dispersion**
5. Other applications of dispersion
6. Conclusion

Smooth adversaries and dispersion

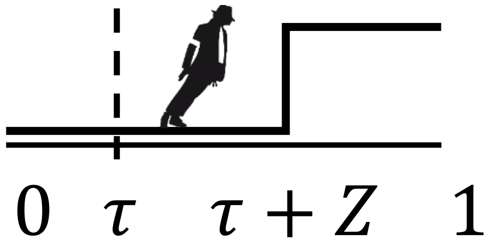
Adversary chooses thresholds $u_t: [0,1] \to \{0,1\}$.



Smooth adversaries and dispersion

Adversary chooses thresholds $u_t: [0,1] \to \{0,1\}$.

Discontinuity τ "smoothed" by adding $Z \sim N(0, \sigma^2)$.



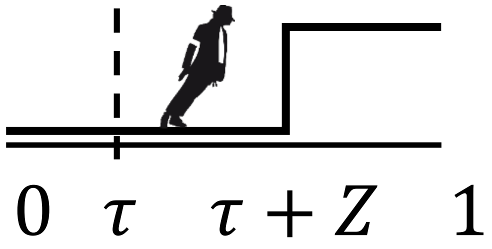
Lemma: W.h.p., $\forall w$, $u_1, \ldots, u_T$ are $\left(w, \tilde{O}\left(\frac{Tw}{\sigma} + \sqrt{T}\right)\right)$-dispersed.

Corollary: $w = \frac{\sigma}{\sqrt{T}} \Rightarrow$ **Full information regret =** $\boldsymbol{O\left(\sqrt{T \log \frac{T}{\sigma}}\right)}$.

Smooth adversaries and dispersion

Adversary chooses thresholds $u_t: [0,1] \to \{0,1\}$.

Discontinuity τ “smoothed” by adding $Z \sim N(0, \sigma^2)$.



Lemma: W.h.p., $\forall w$, $u_1, \ldots, u_T$ are $\left(w, \tilde{O}\left(\frac{Tw}{\sigma} + \sqrt{T}\right)\right)$-dispersed.

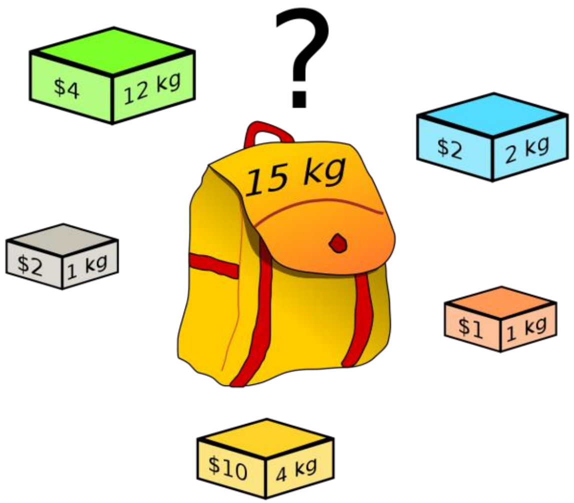
Proof idea: For any width-w interval, $\mathbb{E}[\text{\#discontinuities}] = O\left(\frac{Tw}{\sigma}\right)$.

- VC-dim $\Rightarrow$ w.h.p., every interval has $\tilde{O}\left(\frac{Tw}{\sigma} + \sqrt{T}\right)$ discontinuities.

Simple example: knapsack

Problem instance:

- n items; Item i has value v_i and size s_i
- Knapsack with capacity K



Goal: find most valuable items that fit

Algorithm (parameterized by $\rho \geq 0$)**:**
Add items in decreasing order of $\frac{v_i}{s_i^{\rho}}$

[Gupta and Roughgarden, ‘17]

Simple example: knapsack

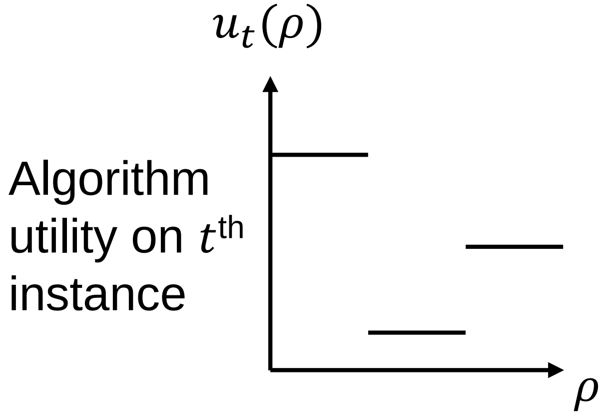
Problem instance:

- n items; Item i has value v_i and size s_i
- Knapsack with capacity K

Goal: find most valuable items that fit

Algorithm (parameterized by $\rho \geq 0$)**:**
Add items in decreasing order of $\frac{v_i}{s_i^\rho}$

[Gupta and Roughgarden, '17]

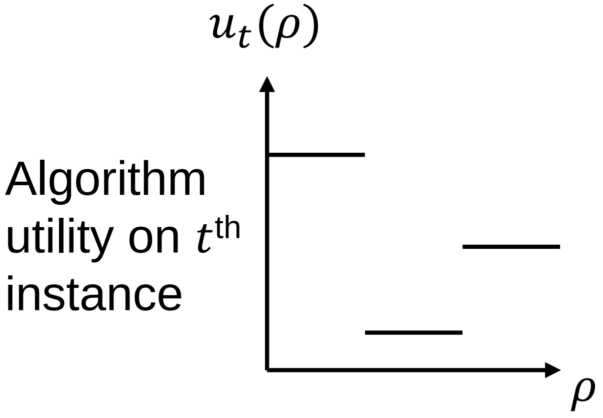


Dispersion for knapsack

Theorem: If instances randomly distributed s.t. on each round:

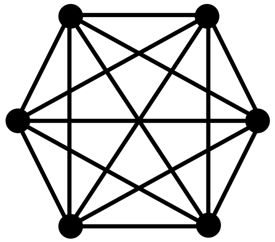
1. Each v_i independent from s_i
2. All (v_i, v_j) have κ-bounded joint density,

W.h.p., for any $\alpha \geq \frac{1}{2}$, $u_1, \ldots, u_T$ are $\left(\tilde{O}\left(\frac{T^{1-\alpha}}{\kappa}\right), \tilde{O}\left((\#\text{ items})^2 T^{\alpha}\right)\right)$-dispersed.



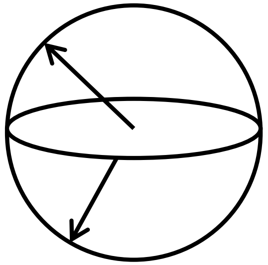
Corollary: Full information regret = $\tilde{O}\left((\#\text{ items})^2\sqrt{T}\right)$.

More Results for Algorithm Configuration



Prove dispersion under **smoothness** assumptions for:

- Maximum weight independent set



Under **no assumptions**, we show dispersion for:

- Integer quadratic programming approximation algos
  - Based on semi-definite programming relaxations
    - s-linear rounding [Feige & Langberg '06]
    - Outward rotations [Zwick '99]
      - Both generalizations of Goemans-Williamson max-cut algorithm ['95].

Outline

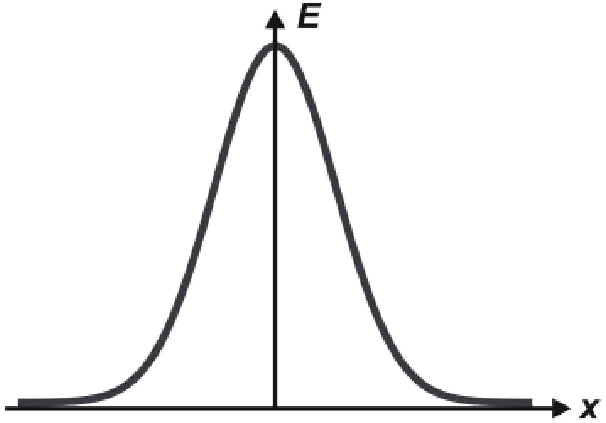
1. Online learning setup
2. Dispersion
3. Regret bounds
4. Examples of dispersion
5. **Other applications of dispersion**
6. Conclusion

Uniform convergence for batch learning

Theorem: If $u_1, \ldots, u_T : \mathbb{R}^d \to [0,1]$ are:

1. Independently drawn from a distribution $\mathcal{D}$
2. Piecewise L-Lipschitz
3. Globally (w, k)-dispersed,

W.h.p., for every $\boldsymbol{\rho} \in \mathbb{R}^d$,



$$\left|\frac{1}{T}\sum_{t=1}^{T} u_t(\boldsymbol{\rho}) - \mathbb{E}_{u\sim\mathcal{D}}[u(\boldsymbol{\rho})]\right| = \tilde{O}\left(\sqrt{\frac{d}{T}\log\frac{1}{w}} + Lw + \frac{k}{T}\right).$$

Differentially private optimization

Given $u_1, \ldots, u_T \colon B_d(\mathbf{0}, 1) \to [0,1]$ up front.

Goal:

- Find (approximate) maximizer of $\frac{1}{T}\sum_{t=1}^{T} u_t$.
- Preserve ϵ-DP w.r.t. changing any one function.

Exponential mechanism [McSherry-Talwar '07] has suboptimality

$$\tilde{O}\left(\frac{d}{T\epsilon}\log\frac{1}{w} + Lw + \frac{k}{T}\right).$$

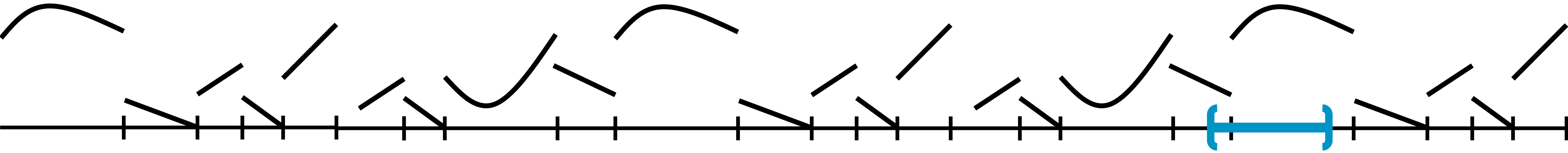
Matching lower bounds!

Outline

1. Online learning setup
2. Dispersion
3. Regret bounds
4. Examples of dispersion
5. Other applications of dispersion
6. **Conclusion**

Conclusions and open questions

- Introduced dispersion.
  - Measures concentration of discontinuities of PWL functions.
  - Implies regret bounds for online optimization of PWL functions.
  - Batch learning and private optimization guarantees.
- Examples of dispersion in real problems.



Conclusions and open questions

- Introduced dispersion.
  - Measures concentration of discontinuities of PWL functions.
  - Implies regret bounds for online optimization of PWL functions.
  - Batch learning and private optimization guarantees.
- Examples of dispersion in real problems.

Open Questions:

- Bad properties beyond discontinuities?
- Config. between full-info and bandit. Can we provide algos?