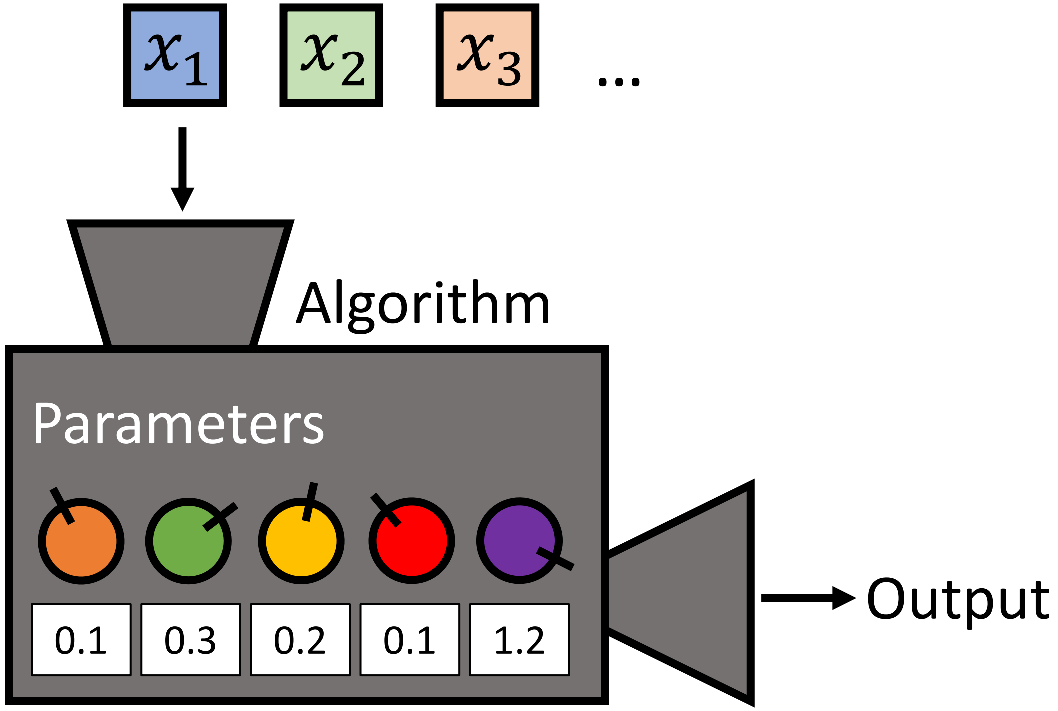
Dispersion for Data-Driven Algorithm Configuration, Online Learning, and Private Optimization

Maria-Florina Balcan, Travis Dick, Ellen Vitercik

Carnegie Mellon University

Data-Driven Algorithm Configuration

Problem instances from specific application.



Goal:

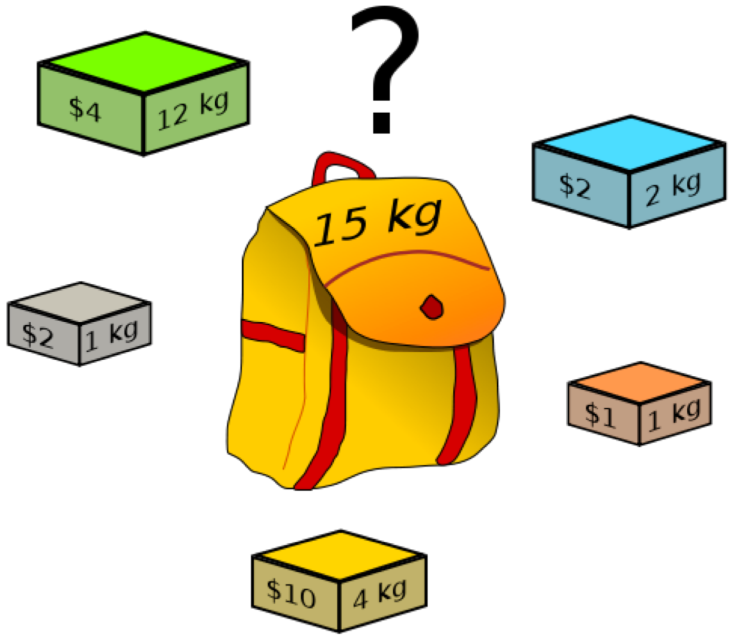
- Automatically find the best parameters for a specific application domain.
- Algorithm is run repeatedly, historic instances are training data.
- Want provable guarantees for online and private settings.

Example: Greedy Knapsack Algorithm

Problem Instance:

- Given n items
- item i has value v_i and size s_i
- a knapsack with capacity K

Find the most value subset of items that fits.



Algorithm: (Parameter $\rho \geq 0$)

Add items in decreasing order of $\mathrm{score}_\rho(i) = v_i / s_i^\rho$.

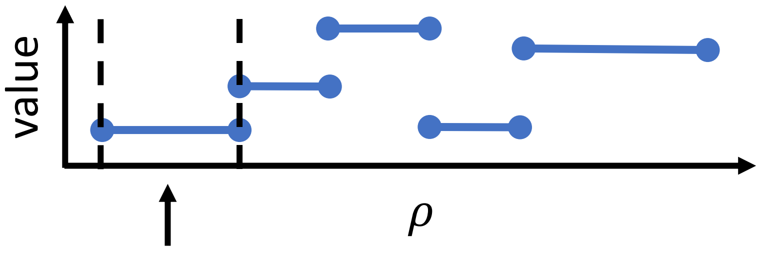
Goal: Find ρ giving highest total value for an application / source of instances.

Observation: For one instance, total value is piecewise constant in ρ.

If ρ and ρ' give the same item ordering, output is the same.

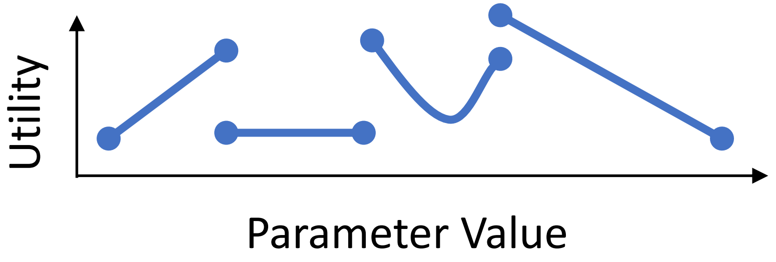
Items i and j only swap relative order at $\rho = \dfrac{\ln(v_i/v_j)}{\ln(s_i/s_j)}$.

So at most n^2 discontinuities.



Order of items fixed.

More generally, utility is often a piecewise Lipschitz function of parameters.



Learning protocol:

For each round $t = 1, \dots, T$:

1. Learner chooses point $\rho_t \in C \subset R^d$.

   (Learner chooses parameter vector ρ)

2. Adversary chooses piecewise L-Lipschitz function $u_t: C \to R$.

   (Adversary chooses problem instance x_t and sets $u_t(\rho) =$ utility of ρ for x_t)

3. Learner gets reward $u_t(\rho_t)$ and either
   - Observes entire function u_t
   - Observes the scalar $u_t(\rho_t)$

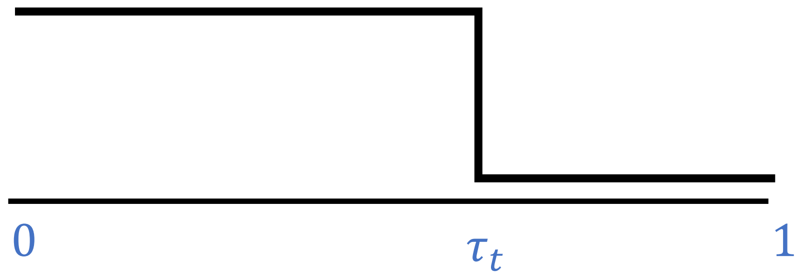
Notation: Let $U_t(\rho) = \sum_{s=1}^{t} u_t(\rho)$

Goal: Minimize regret $= \max_{\rho \in C} U_T(\rho) - \sum_{t=1}^{T} u_t(\rho_t)$.

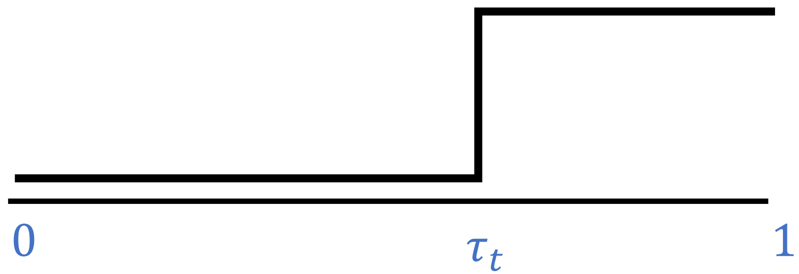
A Mean Adversary

Fact: There exists an adversary choosing piecewise constant functions from $[0,1]$ to $[0,1]$ such that **every** full information online algorithm has **linear regret**.

At round t, adversary chooses a threshold τ_t and flips a coin to choose either



or



Every learner has expected utility of $½$ per round → expected total utility $T/2$.

Let $G_t = \{\rho \in [0,1] : u_s(\rho) = 1 \text{ for all } 1 \leq s \leq t\}$

Set τ_t to be midpoint of G_{t-1} → $\max_{T} U_T(\rho) = T$.

Regret = $T/2$.

Talk Outline

1. Define a condition on collections of PWL functions called ***Dispersion***.

2. Regret bounds for Online PWL Optimization under Dispersion.

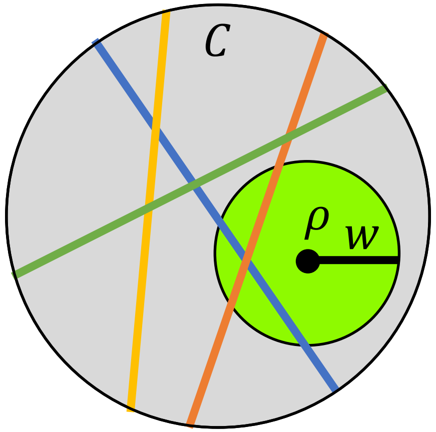
3. Dispersion in algorithm configuration under realistic assumptions.

4. Differentially private optimization of PWL functions.

Dispersion

The mean adversary concentrated discontinuities near ρ^*. Even very near points had low utility!

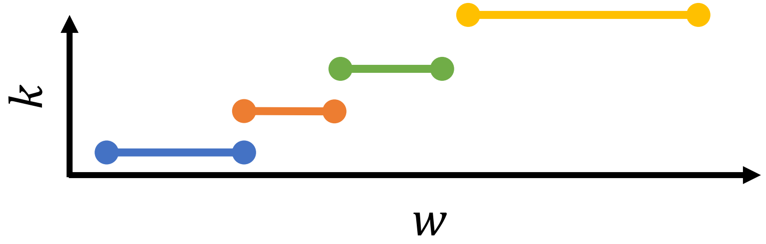
Def: Functions $u_1(\cdot), \dots, u_T(\cdot)$ are $(\boldsymbol{w}, \boldsymbol{k})$***-dispersed at point*** $\boldsymbol{\rho}$ if the ℓ_2-ball $B(\rho, w)$ contains discontinuities for at most k of $u_1 \dots, u_T$.



Each colored line is a discontinuity of one function.

Ball of radius w about ρ contains 2 discontinuities.
→ $(w, 2)$-dispersed.

Functions will satisfy a range of dispersion parameters:



Online Optimization with Dispersion

Full Information Regret Bounds

We analyze the classic Exponentially Weighted Forecaster [Cesa-Bianchi and Lugosi ’06]

Algorithm: (Parameter $\lambda > 0$)
At round t, sample ρ_t from $p_t(\rho) \propto \exp(\lambda U_{t-1}(\rho))$.

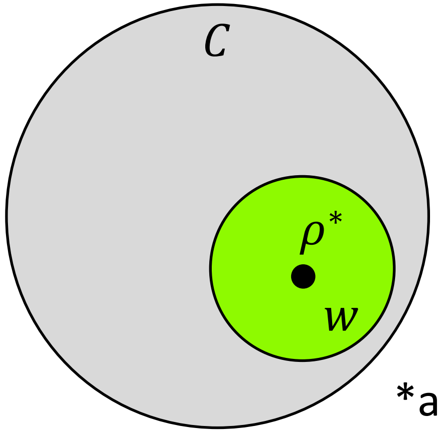
Theorem: If $u_1, \ldots, u_T : C \to [0,1]$ are piecewise L-Lipschitz and (w, k)-dispersed at ρ^*, EWF has regret $O\left(\sqrt{Td \log \frac{1}{w}} + TLw + k\right)$.

Intuition: Any ρ' in $B(\rho^*, w)$ has utility at least $U_T(\rho^*) - TLw - k$. “Many” good points.

When is this a good bound?

If $w = 1/(L\sqrt{T})$ and $k = \tilde{O}(\sqrt{T})$ regret is $\tilde{O}(\sqrt{Td})$.

Note: don’t need to know (w, k) in advance!



*assume C has radius 1.

Matching Lower Bound

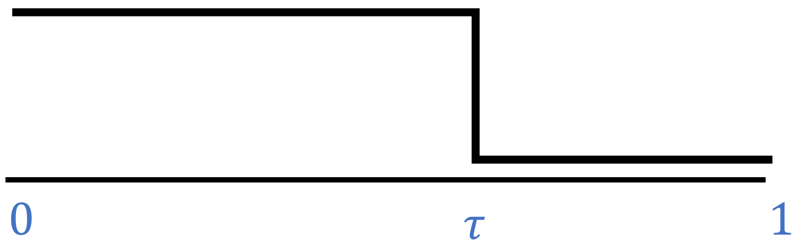
Theorem: For any algorithm A and T big enough, there are piecewise constant functions $u_1, \dots, u_T$ so that A has expected regret at least

$$\Omega\left(\inf_{(w,k)} \sqrt{Td\log\left(\frac{1}{w}\right)} + k\right)$$

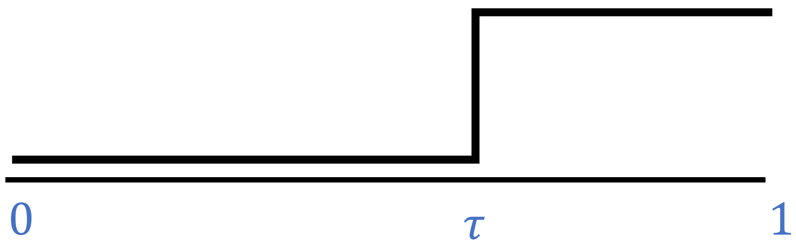
Where the infimum is over all (w,k)-dispersion parameters satisfied by $u_1, \dots, u_T$ at ρ^*.

Our upper bound in this case is $O\left(\inf_{(w,k)} \sqrt{Td\log\frac{1}{w}} + k\right)$.

Idea: Calculate dispersion parameters for worst-case lower bound. Works when $d = 1$.



or



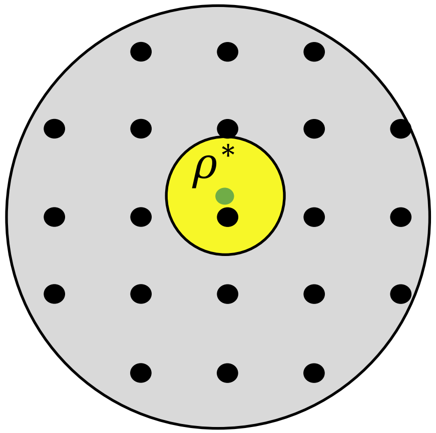
More careful construction works even when sublinear regret is possible, and in higher dimensions.

Bandit Feedback Regret Bounds

Theorem: There exists a bandit-feedback algorithm A such that, if $u_1, \ldots, u_T : C \to [0,1]$ are piecewise L-Lipschitz and (w, k)-dispersed at ρ^*, then the expected regret of A is at most $\tilde{O}\left(\sqrt{Td\left(\frac{1}{w}\right)^d} + TLw + k\right)$.

Reduction:

- Let $\rho_1, \ldots, \rho_N$ be a w-net for C (can take $N \approx 1/w^d$).
- N-armed bandit, payout for arm i at round t is $u_i(\rho_t)$.
- Use EXP3 to play this bandit → regret is $O\left(\sqrt{TN \log N}\right)$.
- Ball of radius w about ρ^* must contain some ρ_i.
- Regret of ρ_i compared to ρ^* is at most $TLw + k$.

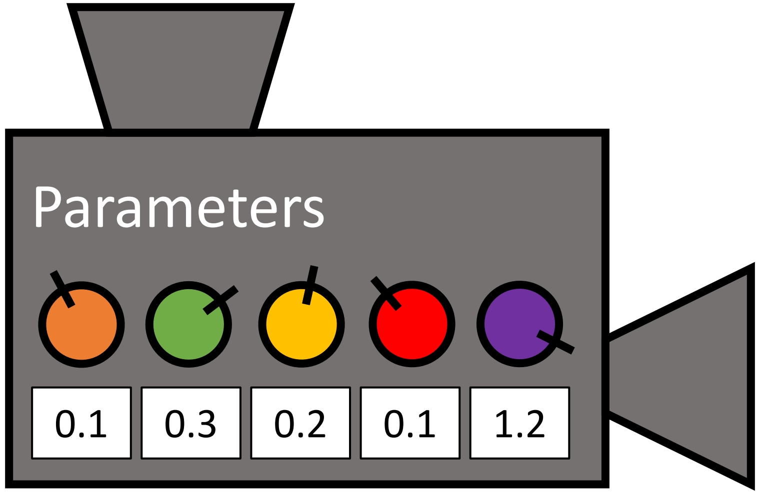


When is this a good bound?

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

Matches dependence on T of a lower bound for (globally) Lipschitz functions.

Dispersion in Algorithm Configuration

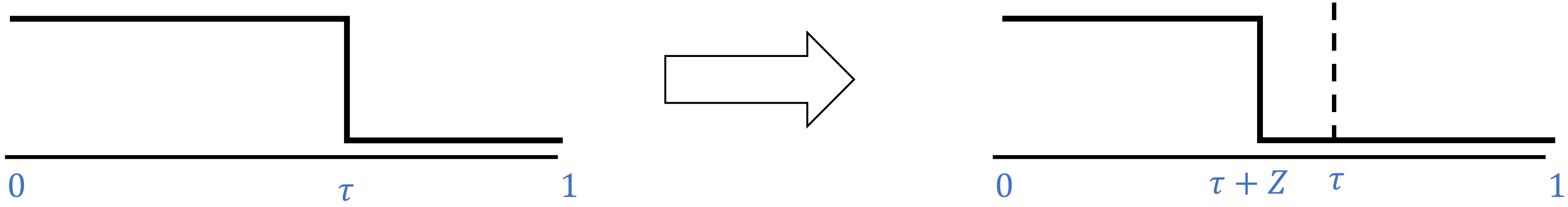


Smoothed Adversaries and Dispersion

Consider any adversary chooses threshold functions $u_1, \ldots, u_T: [0,1] \to [0,1]$:

Location $\tau \in [0,1]$
Orientation $s \in \{\pm 1\}$

Location τ corrupted by adding $Z \sim N(0, \sigma^2)$.

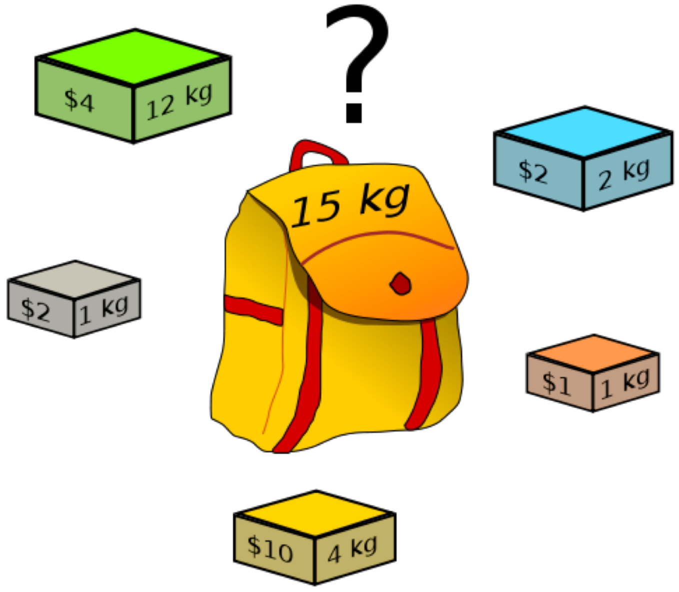


Lemma: For any $w > 0$, the functions $u_1, \ldots, u_T$ are (w, k)-dispersed for $k = \tilde{O}\left(\frac{Tw}{\sigma} + \sqrt{T}\right)$ w.h.p. For any $\alpha > \frac{1}{2}$, we can take $w = T^{\alpha - 1}\sigma$ and $k = \tilde{O}(T^{\alpha})$.

Fix any interval $I = [a, a + w]$.
Expected number of discontinuities in I is at most $T \cdot w / (\sigma\sqrt{2\pi})$.
Uniform convergence → all width w intervals have $k = \tilde{O}(Tw/\sigma + \sqrt{T})$ discontinuities w.h.p.

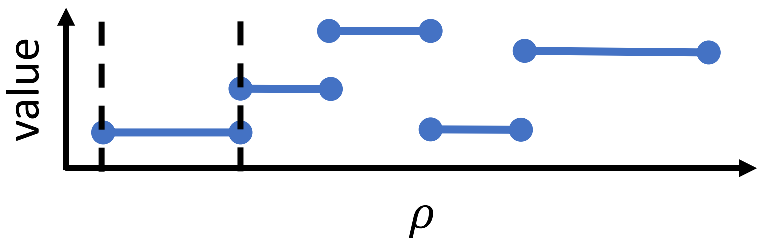
Smoothed Adversaries and Dispersion

More generally: adversary is unable to precisely pick some problem parameters (e.g. item values in knapsack).



Challenges:

- Each utility function has multiple dependent discontinuities.
- Distribution of discontinuity location depends on setting.
- How do we generalize to multiple dimensions?



$$\rho = \frac{\ln(v_i/v_j)}{\ln(s_i/s_j)}$$

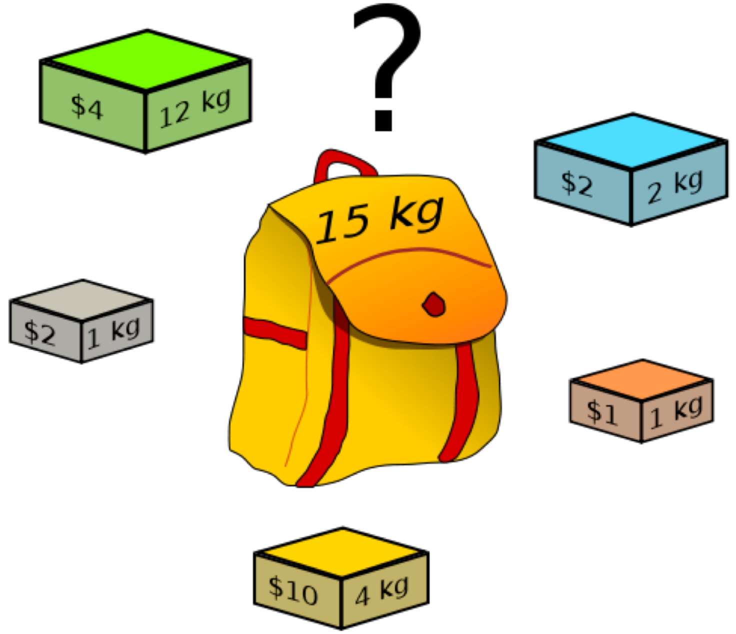
Dispersion decouples problem-specific smoothness arguments from regret bounds and private utility guarantees.

Dispersion in Knapsack

Problem Instance:

- Given n items
- item i has value v_i and size s_i
- a knapsack with capacity K

Find the most value subset of items that fits.



Algorithm: (Parameter $\rho \in [0, M]$)

Add items in decreasing order of $\mathrm{score}_\rho(i) = v_i / s_i^\rho$.

Lemma: If $v_i \in [0,1]$, $s_i \in [1,2]$, and the adversary is “smoothed” (e.g. Gaussian noise with std. dev. σ is added to each v_i) then $u_1, \dots, u_T$ are (w, k)-dispersed with $w = T^{\alpha-1}\sigma$ and $k = \tilde{O}(n^2 T^\alpha)$ for any $\alpha \geq 1/2$ with high probability.

Idea: Discontinuities for items (i, j) across t are independent → similar to noisy thresholds.
Union bound over the n^2 pairs of items.

Full information regret = $\tilde{O}(n^2\sqrt{T})$

Bandit feedback regret = $\tilde{O}(T^{\frac{2}{3}}(\sqrt{\sigma} + n^2))$

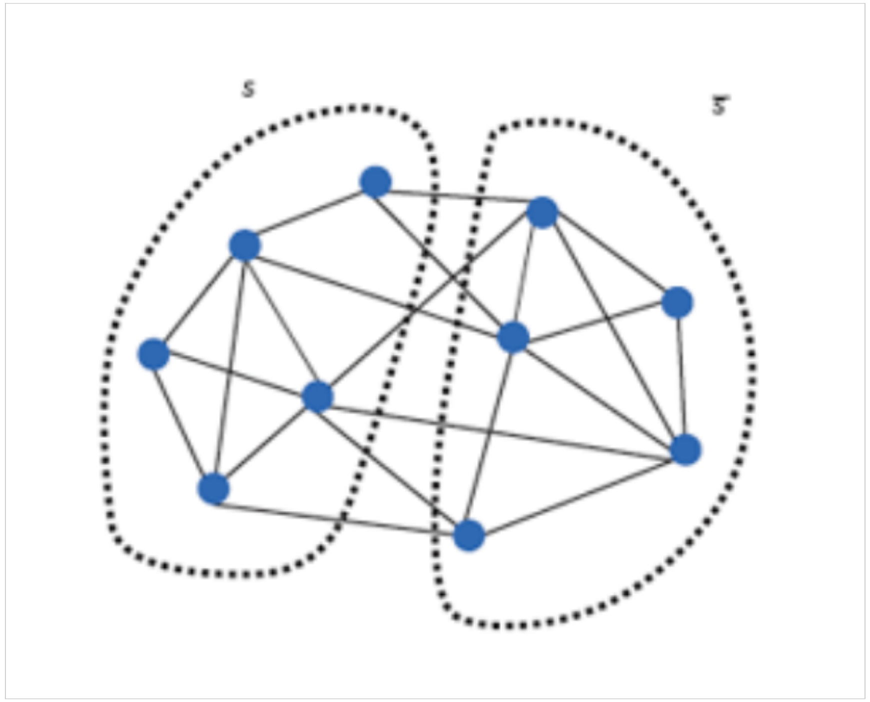
Integer Quadratic Programming

IQP: Given $A \in \mathbb{R}^{n \times n}$, solve $\max_x x^T A x = \sum_{i,j} a_{ij} x_i x_j$ s.t. $x_i \in \{\pm 1\}$ for all $i = 1, \dots, n$.

E.g.: Max cut
Given weighted graph $G(V, E)$
Find cut $S, T \subset V$ maximizing weight of edges between S, T.

x_i = which side of cut is vertex i.

$\max \sum_{(i,j) \in E} w_{ij} (1 - x_i x_j)/2$
s.t. $x_i \in \{\pm 1\}$ for all i.



Integer Quadratic Programming

IQP: Given $A \in \mathbb{R}^{n \times n}$, solve $\max_x x^T A x = \sum_{i,j} a_{ij} x_i x_j$ s.t. $x_i \in \{\pm 1\}$ for all $i = 1, \ldots, n$.

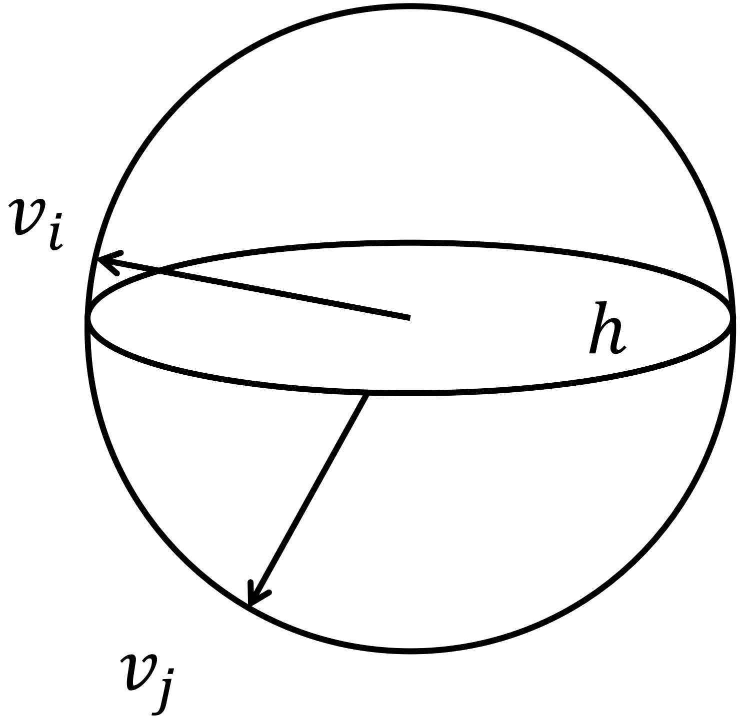
Algorithmic Approach: SDP + Rounding

1. Associate each binary variable x_i with a vector $v_i \in \mathbb{R}^n$. Solve the SDP

$$\max \sum_{i,j} a_{ij} \langle v_i, v_j \rangle$$
$$\text{s.t. } \|v_i\| = 1 \text{ for all } i.$$

2. Rounding Procedure [Goemans & Williamson '95]

- Choose a random hyperplane h
- Set x_i to $+1$ if v_i on positive side of h, -1 otherwise.



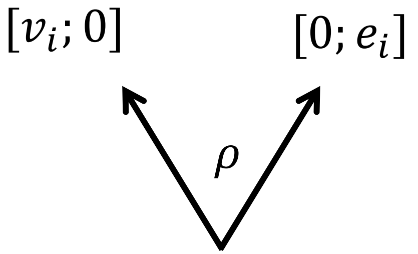
Integer Quadratic Programming: Outward Rotations

IQP: Given $A \in \mathbb{R}^{n\times n}$, solve $\max_x x^T A x = \sum_{i,j} a_{ij}\, x_i x_j$ s.t. $x_i \in \{\pm 1\}$ for all $i = 1, \dots, n$.

Outward Rotation Algorithm:

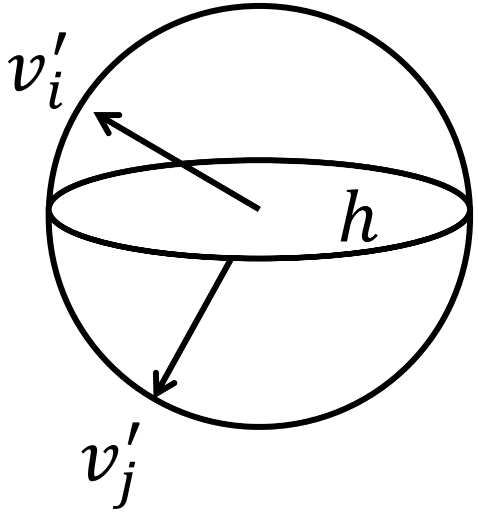
1. Associate each binary variable x_i with a vector v_i.

$$\max \sum_{i,j} a_{ij} \langle v_i, v_j \rangle$$
s.t. $\|v_i\| = 1$ for all i.



2. Outward Rotations: [Zwick '99]

- For each $i \in [n]$, let $v_i' = [\cos(\rho)\, v_i; \sin(\rho)\, e_i] \in \mathbb{R}^{2n}$.
- Pick random hyperplane h and round as in GW algorithm.



$\rho = 0$: GW algorithm.
$\rho = \pi/2$: Random assignment.
Better performance than GW with $\rho \neq 0$ for MaxCut with light cuts.

Goal: Tune parameter $\rho \in [0, \frac{\pi}{2}]$ to maximize $u_t(\rho) = x^\top A x$

Dispersion for Outward Rotations IQP

Think of the random hyperplane as part of the IQP. $u_t(\rho) = u(\rho; A_t, h_t)$.

Lemma: For every sequence of IQPs $A_1, \ldots, A_T$ and $\alpha \geq 1/2$, the corresponding utility functions $u_1, \ldots, u_T$ are (w, k)-dispersed with $w = T^{1-\alpha}$ and $k = \tilde{O}(nT^{\alpha})$ w.h.p. over the randomly chosen hyperplanes.

Idea:

- The adversary can't control the random hyperplanes.
- Discontinuities depend on the hyperplanes → dispersion for free.

Full Information Regret: $\tilde{O}\left(n\sqrt{T}\right)$

Bandit feedback regret: $\tilde{O}\left(nT^{2/3}\right)$

A similar argument holds for s-linear rounding [Feige, Langberg '06].

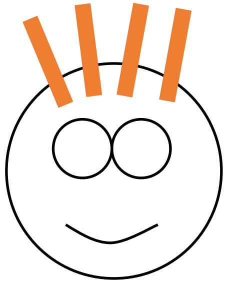
Differentially Private Optimization

Differentially Private Optimization

Goal: Given utility functions $u_1, \dots, u_T$ where each u_i encodes sensitive information about one individual, find an approximate maximizer of $\frac{1}{T}\sum_t u_t(\rho)$ without violating privacy.

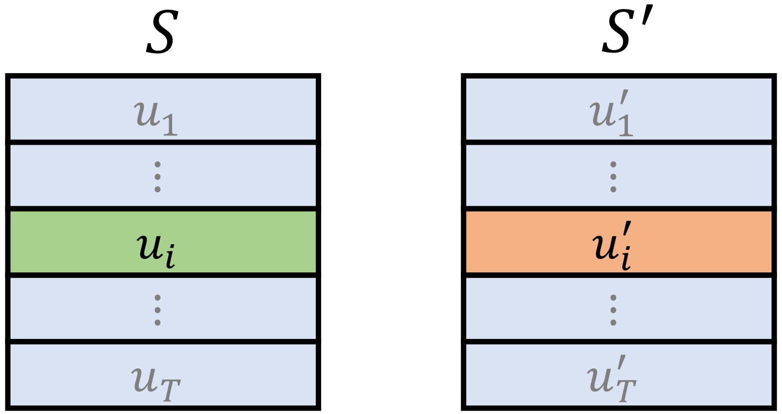
Example:

- Website solves knapsack instances.
- Each instance represents a specific user's values for some set of items.
  - Suppose a new user joins, and the website decreases ρ.
  - Scores for items were given by v_i/s_i^ρ.
  - We might guess new user highly values large items.



Differential Privacy

Def: Two collections of utility functions S and S' are *neighboring* if they differ on at most one function.



Def: A randomized alg. A is ϵ*-differentially private* if for any neighboring collections S, S' and any set C of outcomes, we have:

$$\Pr(A(S) \in C) \leq e^{\epsilon} \cdot \Pr(A(S') \in C)$$

This definition of neighboring is good when:

- Each u_i encodes information about an individual or small group.
- Individuals are not present in too many functions.

Exponential Mechanism Utility

We analyze the exponential mechanism. [McSherry and Talwar '07]

Given a collection of functions $S = \{u_1, \dots, u_T : C \to [0,1]\}$

Algorithm: For $\epsilon > 0$...

Sample ρ from $p(\rho) \propto \exp\left(\frac{\epsilon}{2\Delta} \cdot U_S(\rho)\right)$ where $U_S(\rho) = \frac{1}{T}\sum_{i=1}^{N} u_i(\rho)$.

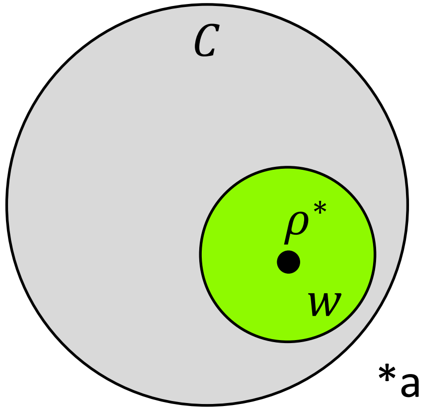
ϵ is the target privacy parameter. $\Delta = 1/N$ is the sensitivity of the average utility.

Theorem: If $u_1, \dots, u_T$ are L-Lipschitz and (w, k)-dispersed, then then with high probability, the exponential mechanism outputs $\hat{\rho}$ such that

$$U_S(\hat{\rho}) \geq \max U_S(\rho) - O\left(\frac{d}{T\varepsilon}\log\frac{1}{w} + Lw + \frac{k}{T}\right)$$

Intuition: Exponential mechanism can fail if there are many more bad points than good.

Any ρ' in $B(\rho^*, w)$ has utility at least $U_T(\rho^*) - TLw - k$.
"Many" good points.



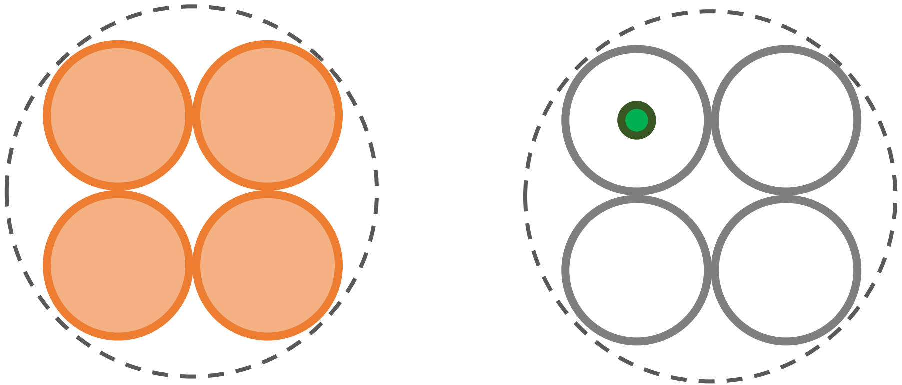
*assume C has radius 1.

Lower bound for Privacy

Theorem: For any ϵ-DP optimizer A there exists a multiset S of T piecewise constant functions from $B(0,1) \subset \mathbb{R}^d$ to $[0,1]$ such that with probability 99%, A outputs an $\Omega\left(\inf_{(w,k)} \frac{d}{N\epsilon} \log \frac{1}{w} + \frac{k}{N}\right)$ suboptimal solution.

Idea:

- Packing argument similar to De [2012].
- Construct many sets of functions whose sets of approximate maximizers are disjoint.
- Every ϵ-DP algorithm must have low utility on at least one.
- Tune the construction so that dispersion parameters match utility lower bound.



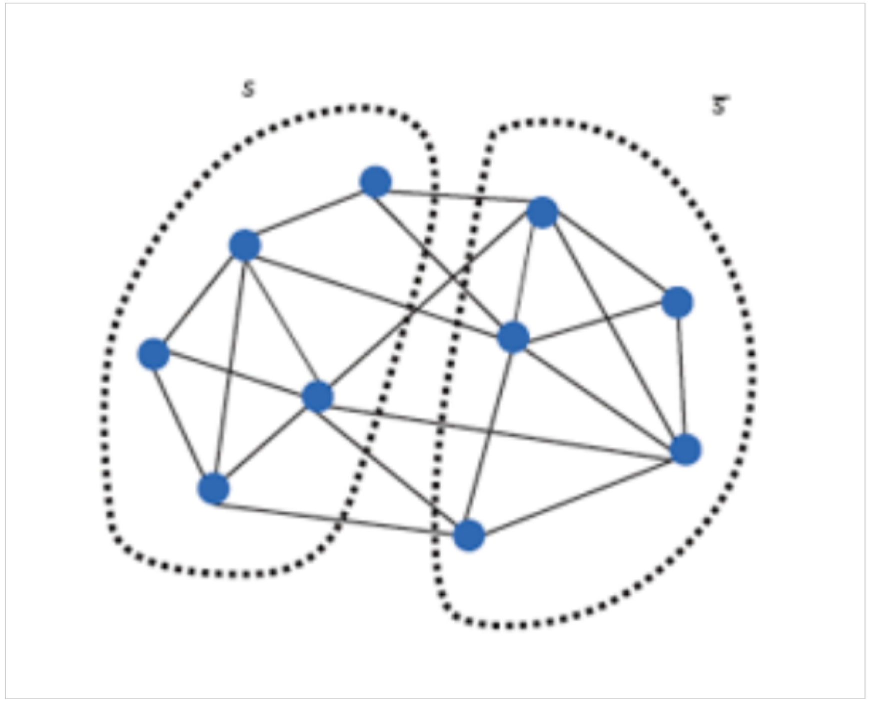
Thanks!

1. Dispersion: measuring the concentration of discontinuities.
2. Dispersion-based regret bounds for online optimization.
3. Differentially private utility guarantees for private optimization.
4. Several interesting applications where smoothness implies dispersion.

Correlation Clustering

IQP: Given $A \in \mathbb{R}^{n \times n}$, solve $\max_{x} x^T A x = \sum_{i,j} a_{ij}\, x_i x_j$ s.t. $x_i \in \{\pm 1\}$ for all $i = 1, \dots, n$.

E.g.: Correlation Clustering
Given weighted graph $G(V, E)$
Find clusters $C_1, C_2 \subset V$ maximizing
sum of weights within cluster minus
sum of weights between clusters.



x_i = which cluster i belongs to.

$\max \sum_{(i,j) \in E} w_{ij} x_i x_j$
s.t. $x_i \in \{\pm 1\}$ for all i.