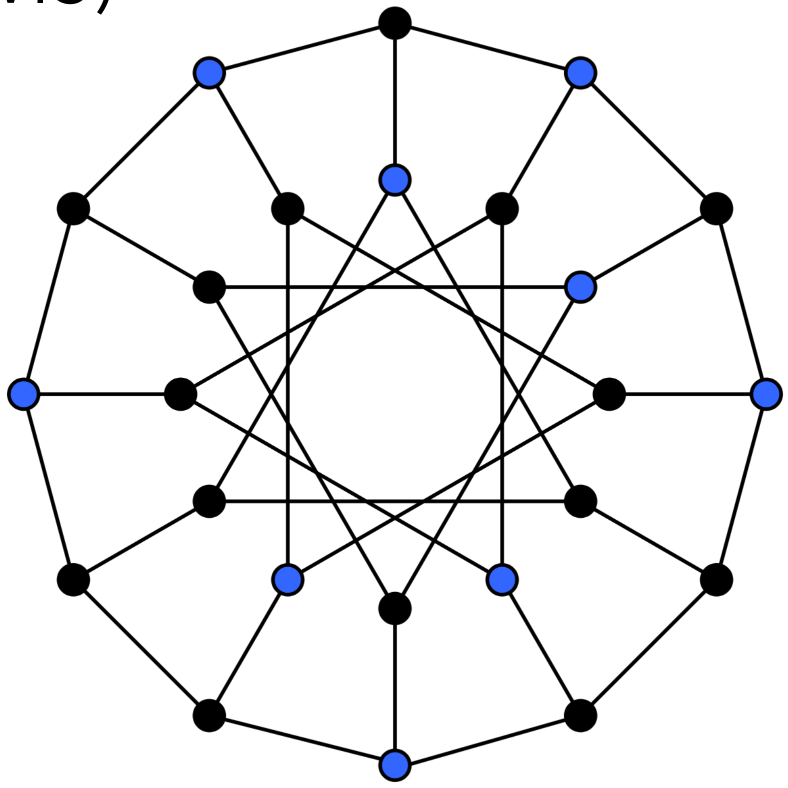
An ML-theory lens on algorithm configuration

Outline

1. **Statistical learning theory**
2. Online learning

Running example

Maximum weight independent set (MWIS)



Problem instance:

- Graph $G = (V, E)$
- n vertices with weights $w_1, \dots, w_n \geq 0$

Goal: find subset $S \subseteq [n]$

- Maximizing $\sum_{i \in S} w_i$
- No nodes $i, j \in S$ are connected: $(i, j) \notin E$

Running example: MWIS

Greedy heuristic:

Greedily add vertices v in decreasing order of $\frac{w_v}{(1+\deg(v))}$

Maintaining independence

Parameterized heuristic [Gupta, Roughgarden, ITCS'16]**:**

Greedily add nodes in decreasing order of $\frac{w_v}{(1+\deg(v))^\rho}$, $\rho \geq 0$

[Inspired by knapsack heuristic by Lehmann et al., JACM'02]

Question: How to choose ρ?

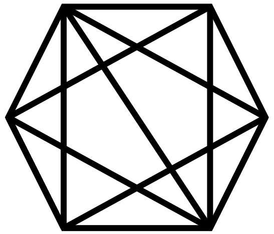
General model

$\mathbb{R}^d$: Set of all parameters

E.g., MWIS parameter $\rho \in \mathbb{R}$, CPLEX parameters, …

$\mathcal{X}$: Set of all inputs

E.g., graphs, integer programs, …



One element $x \in \mathcal{X}$

Algorithmic performance

$u_{\boldsymbol{\rho}}(x)$ = utility of algorithm parameterized by $\boldsymbol{\rho} \in \mathbb{R}^d$ on input x

E.g., runtime, solution quality, memory usage, …

MWIS: If algorithm returns set S, $u_{\boldsymbol{\rho}}(x) = \sum_{i \in S} w_i$

Assume $u_{\boldsymbol{\rho}}(x) \in [-H, H]$

Automated configuration procedure

1. Fix parameterized algorithm
2. Receive set of "typical" inputs sampled from unknown $\mathcal{D}$

Problem instance 1

Problem instance 2

• • •

3. Return parameter setting $\hat{\rho}$ with good avg performance

Runtime, solution quality, etc.

Automated configuration procedure

Statistical question: Will $\hat{\rho}$ have good **future** performance?

More formally: Is the expected performance of $\hat{\rho}$ also good?

Generalization bounds

Key question: For any parameter setting $\boldsymbol{\rho}$,
is **average** utility on training set close to **expected** utility?

Formally: Given samples $x_1, \ldots, x_N \sim \mathcal{D}$, for any $\boldsymbol{\rho}$,

Generalization bounds

Key question: For any parameter setting $\boldsymbol{\rho}$,
is **average** utility on training set close to **expected** utility?

Formally: Given samples $x_1, \ldots, x_N \sim \mathcal{D}$, for any $\boldsymbol{\rho}$,

$$\left| \underbrace{\frac{1}{N} \sum_{i=1}^{N} u_{\boldsymbol{\rho}}(x_i)}_{\text{Empirical average utility}} - \mathbb{E}_{x\sim\mathcal{D}}\left[u_{\boldsymbol{\rho}}(x)\right] \right| \leq \textbf{?}$$

Generalization bounds

Key question: For any parameter setting $\boldsymbol{\rho}$,
is **average** utility on training set close to **expected** utility?

Formally: Given samples $x_1, \ldots, x_N \sim \mathcal{D}$, for any $\boldsymbol{\rho}$,

$$\left| \frac{1}{N} \sum_{i=1}^{N} u_{\boldsymbol{\rho}}(x_i) - \underbrace{\mathbb{E}_{x \sim \mathcal{D}}\left[u_{\boldsymbol{\rho}}(x)\right]}_{\text{Expected utility}} \right| \leq \textbf{?}$$

Generalization bounds

Key question: For any parameter setting $\boldsymbol{\rho}$,
is **average** utility on training set close to **expected** utility?

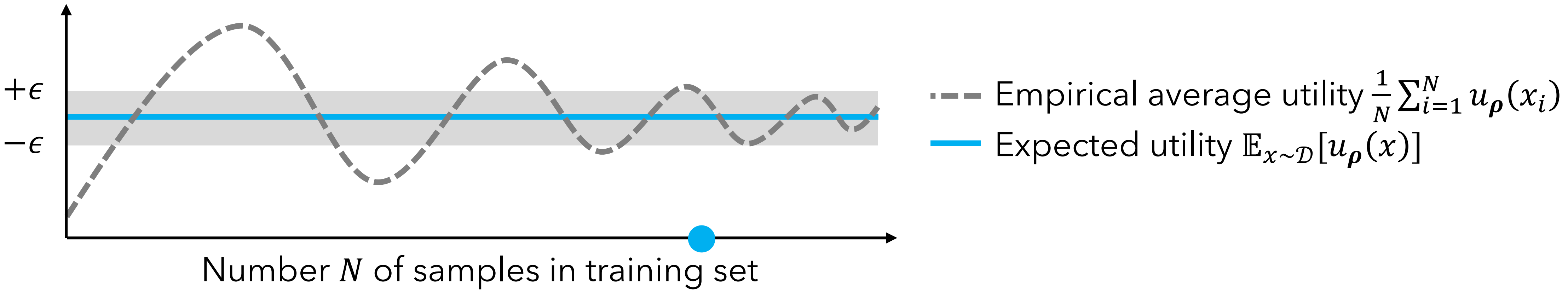
Formally: Given samples $x_1, \ldots, x_N \sim \mathcal{D}$, for any $\boldsymbol{\rho}$,

$$\left| \frac{1}{N} \sum_{i=1}^{N} u_{\boldsymbol{\rho}}(x_i) - \mathbb{E}_{x \sim \mathcal{D}}\left[u_{\boldsymbol{\rho}}(x)\right] \right| \leq \textbf{?}$$

Good **average empirical** utility ➡ Good **expected** utility

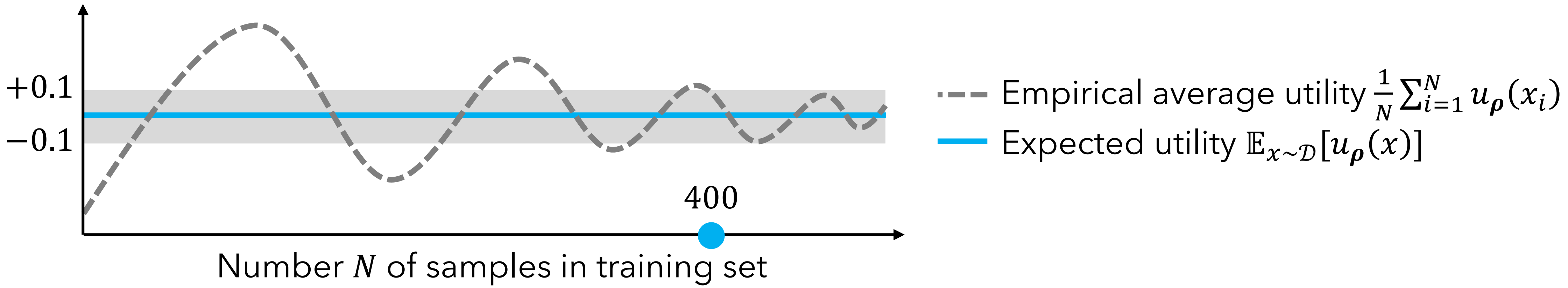
Convergence

Key question: For any parameter setting $\boldsymbol{\rho}$,
is **average** utility on training set close to **expected** utility?



Convergence

Key question: For any parameter setting $\boldsymbol{\rho}$,
is **average** utility on training set close to **expected** utility?



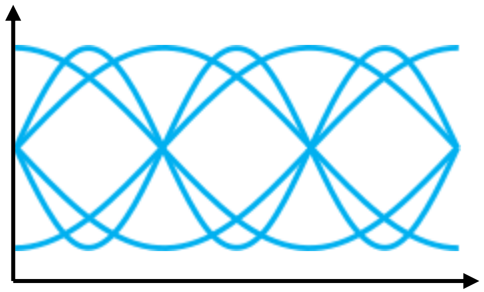
Outline

1. Statistical learning theory
   i. Generalization bounds
   ii. Measures of "intrinsic complexity"
   iii. Pseudo-dimension of MWIS heuristic
2. Online learning

Intrinsic complexity

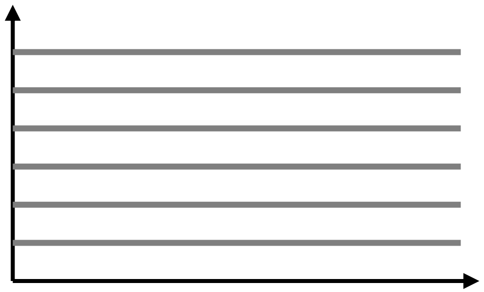
"Intrinsic complexity" of function class $\mathcal{G}$

- Measures how well functions in $\mathcal{G}$ fit complex patterns
- Specific ways to quantify "intrinsic complexity":
  - VC dimension
  - Pseudo-dimension



More complex

Less complex

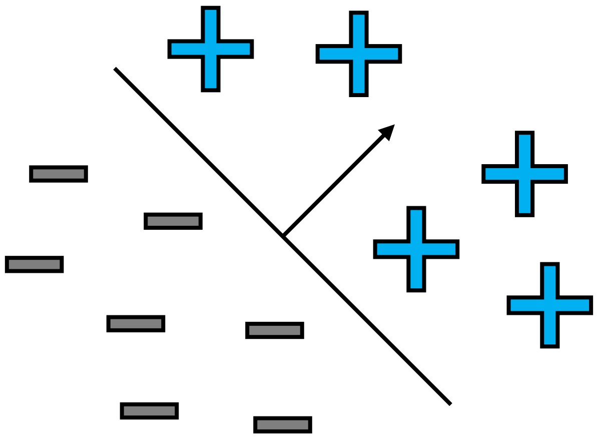


VC dimension

Complexity measure for binary-valued function classes $\mathcal{F}$

(Classes of functions $f: \mathcal{Y} \to \{-1,1\}$)

E.g., linear separators



VC dimension of $\mathcal{F}$

Size of the largest set $\mathcal{S} \subseteq \mathcal{Y}$
that can be labeled in all $2^{|\mathcal{S}|}$ ways by functions in $\mathcal{F}$

Example: $\mathcal{F}$ = Intervals on the real line $f_{a,b}(x) = \begin{cases} 1 & \text{if } x \in (a,b) \\ 0 & \text{else} \end{cases}$

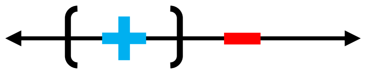
VCdim($\mathcal{F}$) ≥ 2

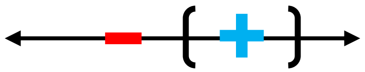
VC dimension of $\mathcal{F}$

Size of the largest set $\mathcal{S} \subseteq \mathcal{Y}$
that can be labeled in all $2^{|\mathcal{S}|}$ ways by functions in $\mathcal{F}$

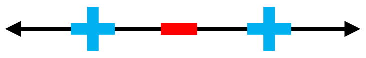
Example: $\mathcal{F}$ = Intervals on the real line $f_{a,b}(x) = \begin{cases} 1 & \text{if } x \in (a,b) \\ 0 & \text{else} \end{cases}$

$\text{VCdim}(\mathcal{F}) \geq 2$





$\text{VCdim}(\mathcal{F}) \leq 2$



Sample complexity using VC dimension

Theorem [Vapnik, Chervonenkis, '71]:

- For $\epsilon, \delta \in (0,1)$, let $N = O\left(\frac{\mathrm{VCdim}(\mathcal{F})}{\epsilon^2} \log \frac{1}{\delta}\right)$
- $\mathcal{D}$ is an unknown distribution over $\mathcal{Y}$
- $f^*: \mathcal{Y} \to \{0,1\}$ is an unknown target function
- Let $\{(y_1, f^*(y_1)), \dots, (y_N, f^*(y_N))\}$ be the training set
- With probability at least $1 - \delta$ over $y_1, \dots, y_N \sim \mathcal{D}, \forall f \in \mathcal{F}$,

$$\left| \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\{f(y_i) \neq f^*(y_i)\}} - \mathbb{P}_{y \sim \mathcal{D}}[f(y) \neq f^*(y)] \right| \leq \epsilon$$

Sample complexity using VC dimension

Theorem [Vapnik, Chervonenkis, '71]: (alternate formulation)

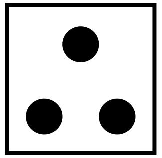
- For $\epsilon, \delta \in (0,1)$, let $N = O\left(\frac{\mathrm{VCdim}(\mathcal{F})}{\epsilon^2} \log \frac{1}{\delta}\right)$
- $\mathcal{D}$ is an unknown distribution over $\mathcal{Y}$
- With probability at least $1 - \delta$ over $y_1, \ldots, y_N \sim \mathcal{D}, \forall f \in \mathcal{F}$,

$$\left| \frac{1}{N} \sum_{i=1}^{N} f(y_i) - \mathbb{E}_{y \sim \mathcal{D}}[f(y)] \right| \leq \epsilon$$

VC dimension of $\mathcal{F}$

Size of the largest set $\mathcal{S} \subseteq \mathcal{Y}$
that can be labeled in all $2^{|\mathcal{S}|}$ ways by functions in $\mathcal{F}$

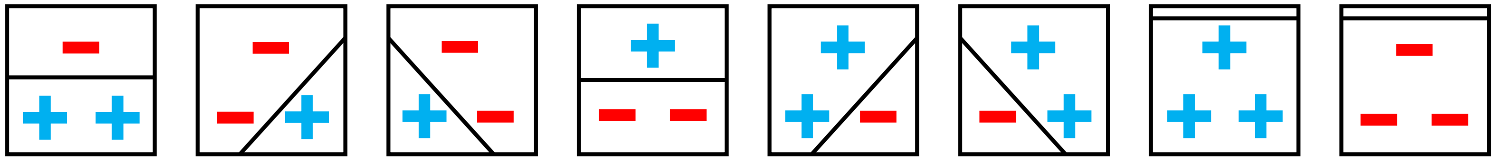
Example: $\mathcal{F}$ = Linear separators in $\mathbb{R}^2$ VCdim($\mathcal{F}$) ≥ 3



VC dimension of $\mathcal{F}$

Size of the largest set $\mathcal{S} \subseteq \mathcal{Y}$
that can be labeled in all $2^{|\mathcal{S}|}$ ways by functions in $\mathcal{F}$

Example: $\mathcal{F}$ = Linear separators in $\mathbb{R}^2$ VCdim($\mathcal{F}$) ≥ 3



VCdim($\mathcal{F}$) ≤ 3

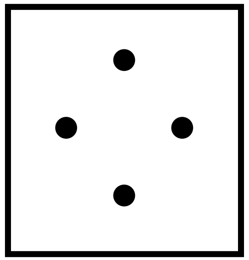
VCdim({Linear separators in $\mathbb{R}^d$}) $= d + 1$

VC dimension of $\mathcal{F}$

Size of the largest set $\mathcal{S} \subseteq \mathcal{Y}$
 that can be labeled in all $2^{|\mathcal{S}|}$ ways by functions in $\mathcal{F}$

Example: $\mathcal{F}$ = Axis-aligned rectangles

$\mathrm{VCdim}(\mathcal{F}) \geq 4$

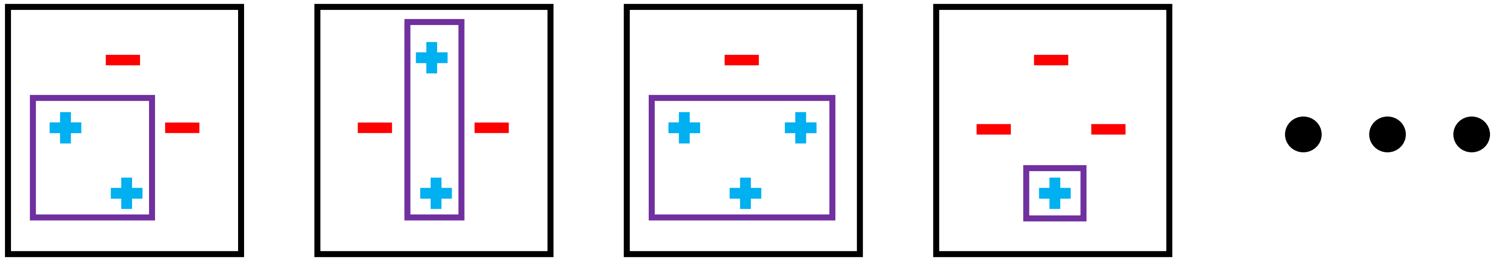


VC dimension of $\mathcal{F}$

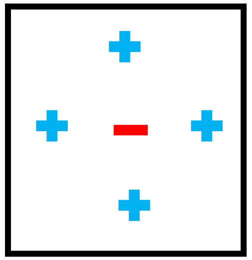
Size of the largest set $\mathcal{S} \subseteq \mathcal{Y}$
that can be labeled in all $2^{|\mathcal{S}|}$ ways by functions in $\mathcal{F}$

Example: $\mathcal{F}$ = Axis-aligned rectangles

$\text{VCdim}(\mathcal{F}) \geq 4$



$\text{VCdim}(\mathcal{F}) \leq 4$



VC dimension of $\mathcal{F}$

Size of the largest set $\mathcal{S} \subseteq \mathcal{Y}$
that can be labeled in all $2^{|\mathcal{S}|}$ ways by functions in $\mathcal{F}$

Mathematically, for $\mathcal{S} = \{y_1, \dots, y_N\}$,

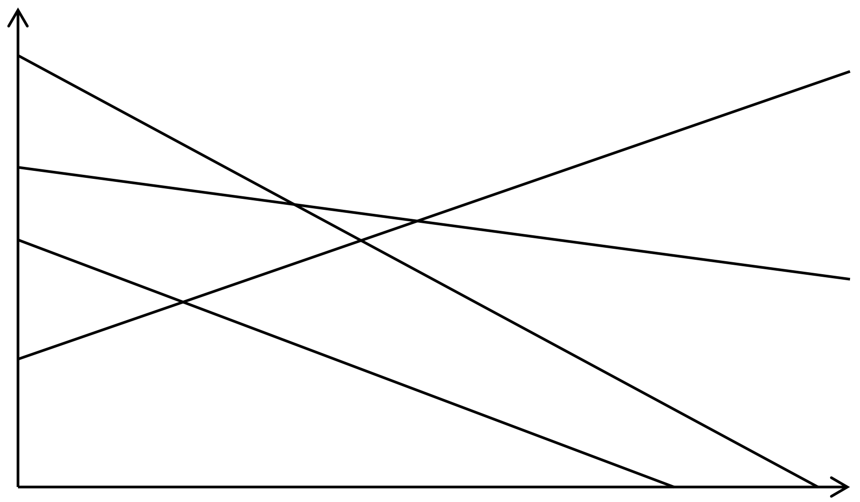
$$\left| \left\{ \begin{pmatrix} f(y_1) \\ \vdots \\ f(y_N) \end{pmatrix} : f \in \mathcal{F} \right\} \right| = 2^N$$

Pseudo-dimension

Complexity measure for real-valued function classes $\mathcal{G}$

(Classes of functions $g: \mathcal{Y} \to [-H, H]$)

E.g., affine functions



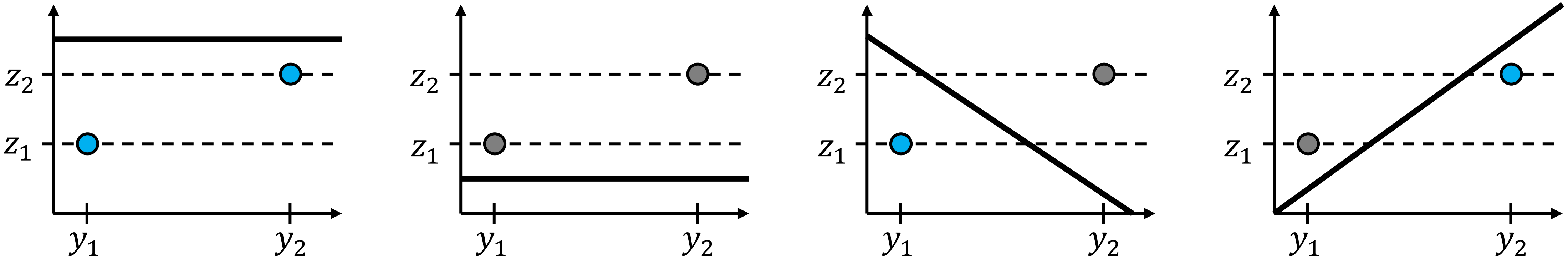
Pseudo-dimension of $\mathcal{G}$

Size of the largest set $\{y_1, \dots, y_N\} \subseteq \mathcal{Y}$ s.t.:

for some *targets* $z_1, \dots, z_N \in \mathbb{R}$,

all 2^N above/below patterns achieved by functions in $\mathcal{G}$

Example: $\mathcal{G}$ = Affine functions in $\mathbb{R}$ $\mathrm{Pdim}(\mathcal{G}) \geq 2$



Can also show that $\mathrm{Pdim}(\mathcal{G}) \leq 2$

Pseudo-dimension of $\mathcal{G}$

Size of the largest set $\{y_1, \ldots, y_N\} \subseteq \mathcal{Y}$ s.t.:

for some *targets* $z_1, \ldots, z_N \in \mathbb{R}$,

all 2^N above/below patterns achieved by functions in $\mathcal{G}$

Mathematically,

$$\left|\left\{\begin{pmatrix} \mathbf{1}_{\{g(y_1) \geq z_1\}} \\ \vdots \\ \mathbf{1}_{\{g(y_N) \geq z_N\}} \end{pmatrix} : g \in \mathcal{G}\right\}\right| = 2^N$$

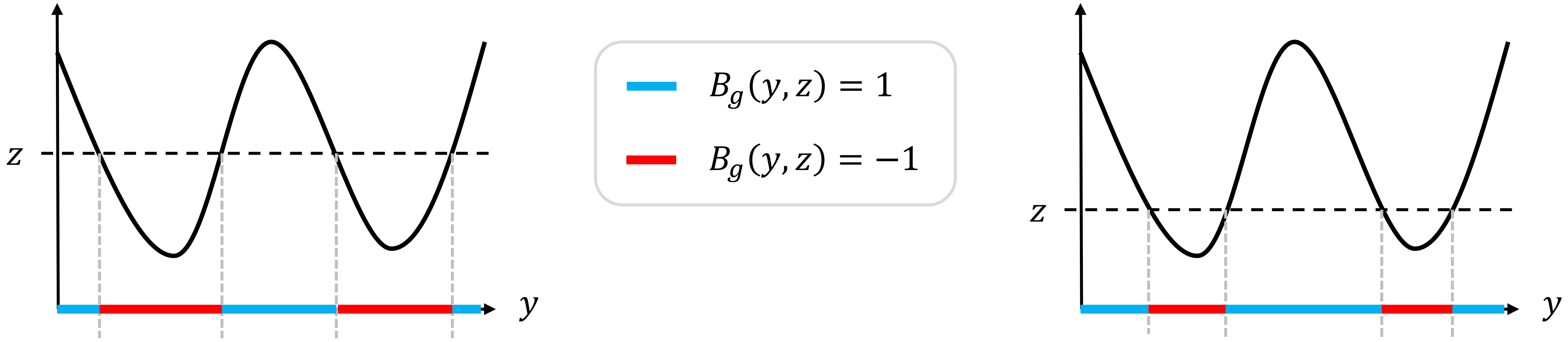
Another interpretation of pseudo-dim

For any $g \in \mathcal{G}$:

B_g = indicator function of the region below the graph of g

$$B_g(y, z) = \operatorname{sgn}(g(y) - z)$$

Illustration of $B_g(y, z)$ with a fixed z and varying y:



Another interpretation of pseudo-dim

For any $g \in \mathcal{G}$:

B_g = indicator function of the region below the graph of g

$$B_g(y, z) = \operatorname{sgn}(g(y) - z)$$

Fact: $\mathrm{Pdim}(\mathcal{G}) = \mathrm{VCdim}\big(\{B_g : g \in \mathcal{G}\}\big)$

Sample complexity using pseudo-dim

Theorem [Pollard, '84]:

- For $\epsilon, \delta \in (0,1)$, let $N = O\left(\frac{\mathrm{Pdim}(\mathcal{G})}{\epsilon^2} \log \frac{1}{\delta}\right)$
- $\mathcal{D}$ is an unknown distribution over $\mathcal{Y}$
- With probability at least $1 - \delta$ over $y_1, \ldots, y_N \sim \mathcal{D}, \forall g \in \mathcal{G}$,

$$\left| \frac{1}{N} \sum_{i=1}^{N} g(y_i) - \mathbb{E}_{y \sim \mathcal{D}}[g(y)] \right| \leq \epsilon H$$

Sample complexity using pseudo-dim

In the context of **algorithm configuration**:

- $\mathcal{U} = \{u_{\boldsymbol{\rho}} : \boldsymbol{\rho} \in \mathbb{R}^d\}$ measure algorithm **performance**
- For $\epsilon, \delta \in (0,1)$, let $N = O\left(\frac{\mathrm{Pdim}(\mathcal{U})}{\epsilon^2} \log \frac{1}{\delta}\right)$
- With probability at least $1 - \delta$ over $x_1, \ldots, x_N \sim \mathcal{D}, \forall \boldsymbol{\rho} \in \mathbb{R}^d$,

$$\left| \underbrace{\frac{1}{N} \sum_{i=1}^{N} u_{\boldsymbol{\rho}}(x_i)}_{\text{Empirical average utility}} - \underbrace{\mathbb{E}_{x \sim \mathcal{D}}\left[u_{\boldsymbol{\rho}}(x)\right]}_{\text{Expected utility}} \right| \leq \epsilon H$$

Outline

1. Statistical learning theory
    i. Generalization bounds
    ii. Measures of “intrinsic complexity”
    iii. **Pseudo-dimension of MWIS heuristic**
2. Online learning

Pseudo-dimension of MWIS heuristic

- N MWIS instances $x_1, \ldots, x_N$, each with n vertices
- N targets $z_1, \ldots, z_N \in \mathbb{R}$
- How many above-below patterns can we make?

$$\left| \left\{ \begin{pmatrix} \mathbf{1}_{\{u_\rho(x_1) \geq z_1\}} \\ \vdots \\ \mathbf{1}_{\{u_\rho(x_N) \geq z_N\}} \end{pmatrix} : \rho \in \mathbb{R} \right\} \right| \leq \textbf{?}$$

Theorem [Gupta, Roughgarden, ITCS'16]**:** at most Nn^2

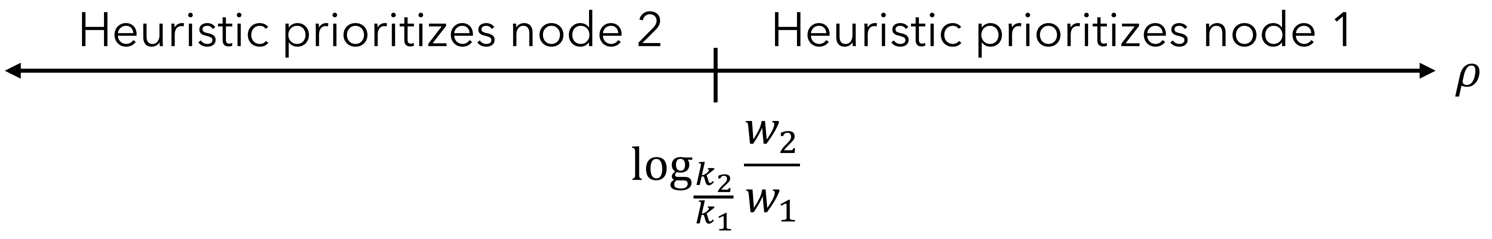
Pseudo-dimension of MWIS heuristic

Let's start with a single instance:

- Weights $w_1, \dots, w_n \geq 0$
- $\deg(i) + 1 = k_i$

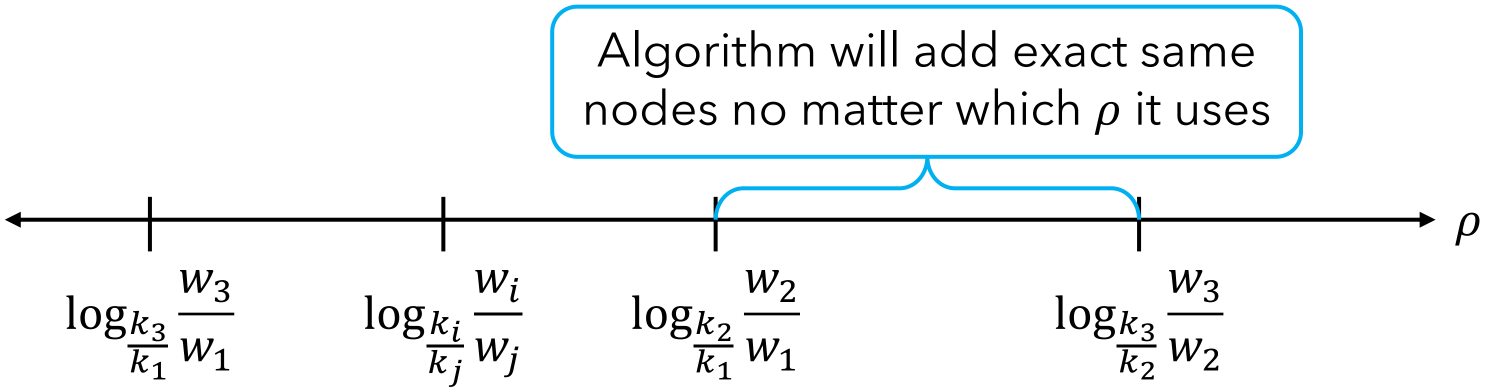
Algorithm parameterized by ρ would add **node 1** before **2** if:

$$\frac{w_1}{k_1^\rho} \geq \frac{w_2}{k_2^\rho} \qquad \Leftrightarrow \qquad \rho \geq \log_{\frac{k_2}{k_1}} \frac{w_2}{w_1}$$



Pseudo-dimension of MWIS heuristic

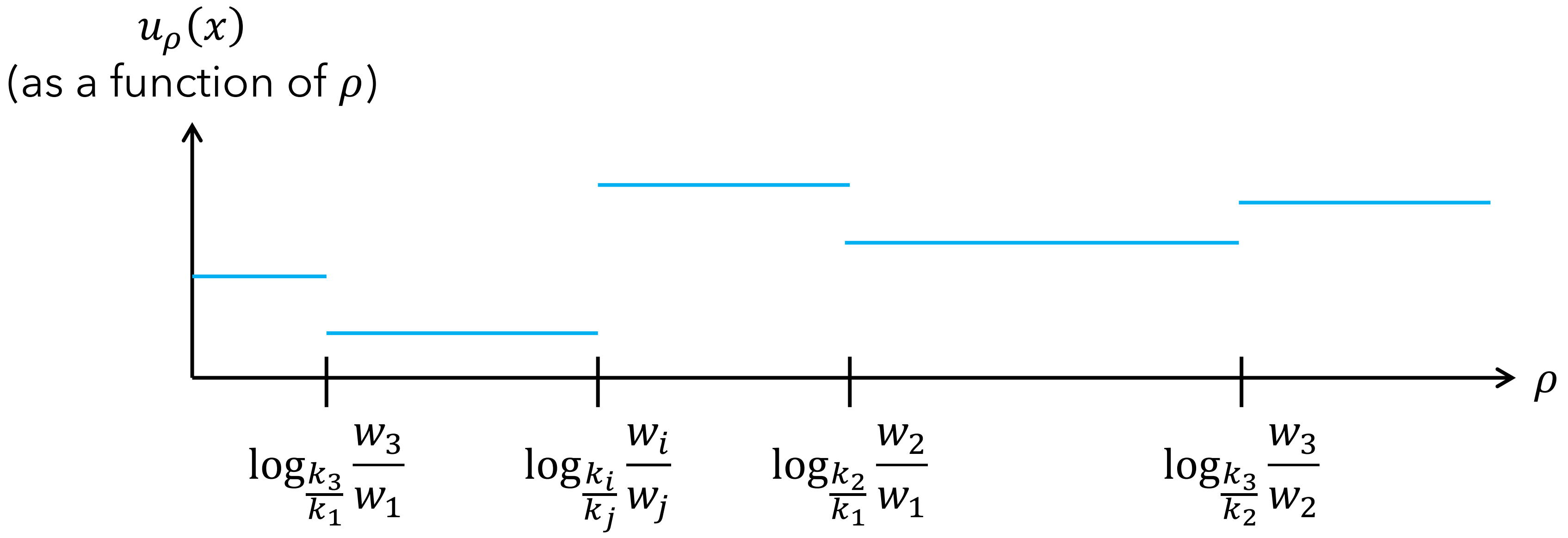
- $\binom{n}{2}$ thresholds per instance
- Partition $\mathbb{R}$ into regions where algorithm's output is fixed



Pseudo-dimension of MWIS heuristic

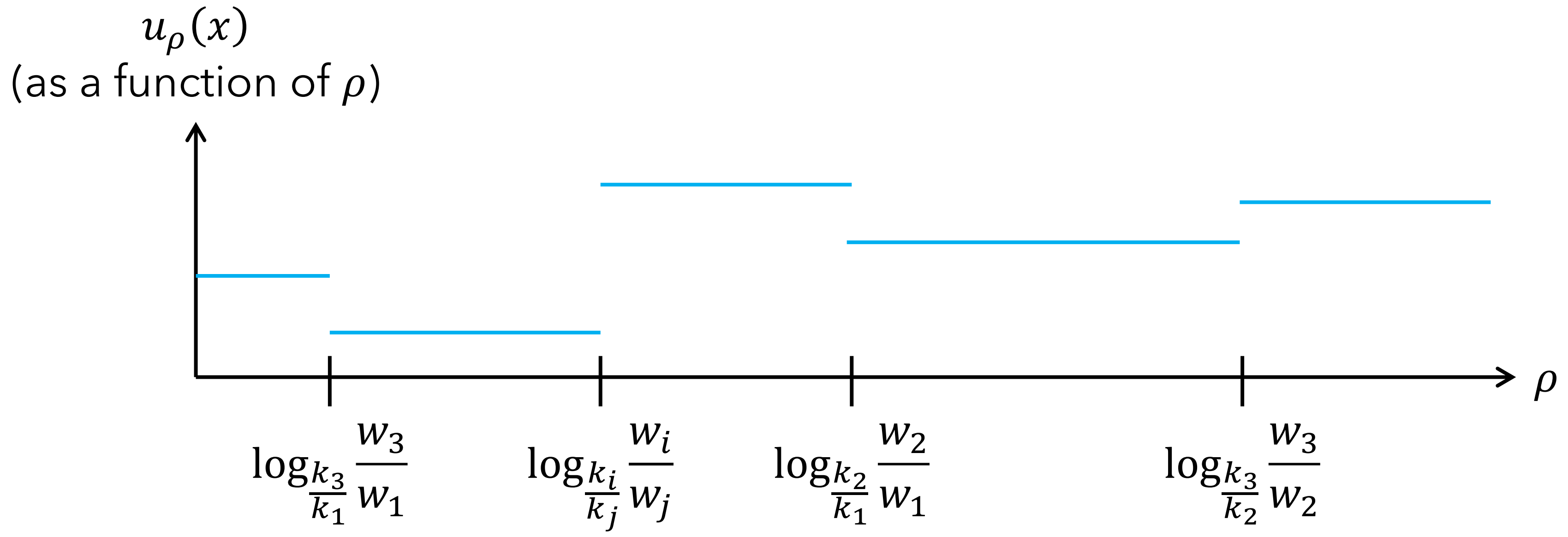
- $\binom{n}{2}$ thresholds per instance
- Partition $\mathbb{R}$ into regions where algorithm's output is fixed

$\Rightarrow u_\rho(x)$ is constant



Pseudo-dimension of MWIS heuristic

- For N instances $x_1, \dots, x_N$, total of $N\binom{n}{2}$ thresholds
- Partition $\mathbb{R}$ into $N\binom{n}{2} + 1$ regions where $u_\rho(x_i)$ is constant $\forall i$



Pseudo-dimension of MWIS heuristic

- For N instances $x_1, \dots, x_N$, total of $N\binom{n}{2}$ thresholds
- Partition $\mathbb{R}$ into $N\binom{n}{2} + 1$ regions where $u_\rho(x_i)$ is constant $\forall i$

$$\Rightarrow \left| \left\{ \begin{pmatrix} \mathbf{1}_{\{u_\rho(x_1) \geq z_1\}} \\ \vdots \\ \mathbf{1}_{\{u_\rho(x_N) \geq z_N\}} \end{pmatrix} : \rho \in \mathbb{R} \right\} \right| \leq N\binom{n}{2} + 1$$

  - If ρ_1, ρ_2 from same region, $u_{\rho_1}(x_i) = u_{\rho_2}(x_i)\ \forall i$,

$$\Rightarrow \begin{pmatrix} \mathbf{1}_{\{u_{\rho_1}(x_1) \geq z_1\}} \\ \vdots \\ \mathbf{1}_{\{u_{\rho_1}(x_N) \geq z_N\}} \end{pmatrix} = \begin{pmatrix} \mathbf{1}_{\{u_{\rho_2}(x_1) \geq z_1\}} \\ \vdots \\ \mathbf{1}_{\{u_{\rho_2}(x_N) \geq z_N\}} \end{pmatrix}$$

Pseudo-dimension of MWIS heuristic

If all 2^N above/below patterns achievable,

$$2^N = \left| \left\{ \begin{pmatrix} \mathbf{1}_{\{u_\rho(x_1) \geq z_1\}} \\ \vdots \\ \mathbf{1}_{\{u_\rho(x_N) \geq z_N\}} \end{pmatrix} : \rho \in \mathbb{R} \right\} \right| \leq N \binom{n}{2} + 1$$

Implies that $N = O(\log n)$, so $\text{Pdim}(\mathcal{U}) = O(\log n)$

MWIS sample complexity

For $\epsilon, \delta \in (0,1)$, let $N = O\left(\frac{\log n}{\epsilon^2} \log \frac{1}{\delta}\right)$

With probability at least $1 - \delta$ over $x_1, \ldots, x_N \sim \mathcal{D}$, $\forall \rho \in \mathbb{R}$,

$$\left| \underbrace{\frac{1}{N} \sum_{i=1}^{N} u_\rho(x_i)}_{\textbf{Empirical average utility}} - \underbrace{\mathbb{E}_{x \sim \mathcal{D}}\left[u_\rho(x)\right]}_{\textbf{Expected utility}} \right| \leq \epsilon H$$

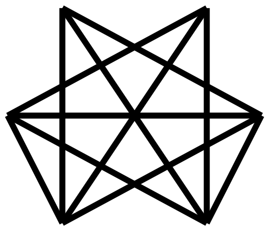
Outline

1. Statistical learning theory
2. **Online learning**

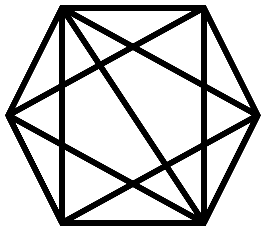
Online algorithm configuration

What if inputs are not i.i.d., but even adversarial?

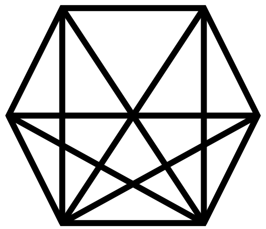
Day 1: $\boldsymbol{\rho}_1$



Day 2: $\boldsymbol{\rho}_2$



Day 3: $\boldsymbol{\rho}_3$



Goal: Compete with best parameter setting in hindsight

- Impossible in the worst case
- Under what conditions is online configuration possible?

Setup

$$\overbrace{\text{To start: finite \# of algorithms (can be generalized)}}$$

Timestep	Algorithm 1	Algorithm 2	Algorithm 3	⋯	Algorithm k
1					

Setup

E.g., independent set weight

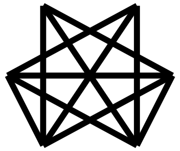
Solution quality

Timestep	**Algorithm 1**	**Algorithm 2**	**Algorithm 3**	$\cdots$	**Algorithm k**
1					

Setup

	Solution quality				
Timestep	**Algorithm 1**	**Algorithm 2**	**Algorithm 3**	...	**Algorithm k**
1					

Setup

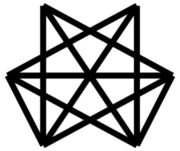
	Solution quality				
Timestep	**Algorithm 1**	**Algorithm 2**	**Algorithm 3**	$\cdots$	**Algorithm k**
1 	2.8	9.3	0.3	$\cdots$	1.4

Full information: Learner sees all solution qualities

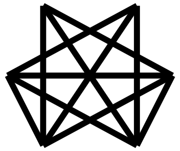
Focus of this lecture (for simplicity)

Will discuss other models in a few slides

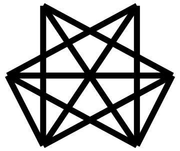
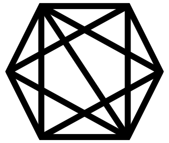
Setup

	Solution quality				
Timestep	**Algorithm 1**	**Algorithm 2**	**Algorithm 3**	⋯	**Algorithm k**
1 	2.8	9.3	0.3	⋯	1.4
2					

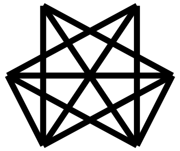
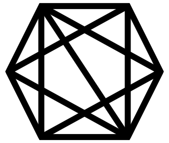
Setup

	Solution quality				
Timestep	**Algorithm 1**	**Algorithm 2**	**Algorithm 3**	…	**Algorithm k**
1 	2.8	9.3	0.3	…	1.4
2					

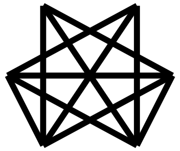
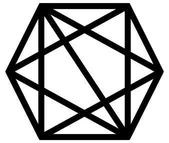
Setup

Timestep		Solution quality				
		Algorithm 1	Algorithm 2	Algorithm 3	⋯	Algorithm k
1		2.8	9.3	0.3	⋯	1.4
2		3.7	4.3	5.8	⋯	1.0

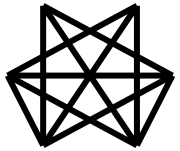
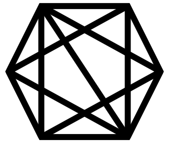
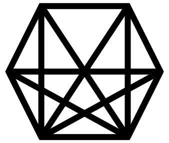
Setup

	Solution quality				
Timestep	**Algorithm 1**	**Algorithm 2**	**Algorithm 3**	⋯	**Algorithm k**
1 	2.8	9.3	0.3	⋯	1.4
2 	3.7	4.3	5.8	⋯	1.0
⋮	⋮	⋮	⋮	⋱	⋮
T					

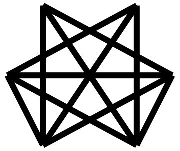
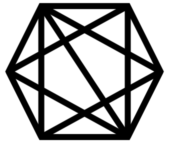
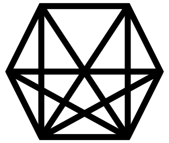
Setup

Timestep		Solution quality				
		Algorithm 1	Algorithm 2	Algorithm 3	⋯	Algorithm k
1		2.8	9.3	0.3	⋯	1.4
2		3.7	4.3	5.8	⋯	1.0
	⋮	⋮	⋮	⋮	⋱	⋮
T						

Setup

		Solution quality				
Timestep		**Algorithm 1**	**Algorithm 2**	**Algorithm 3**	$\cdots$	**Algorithm k**
1		2.8	9.3	0.3	$\cdots$	1.4
2		3.7	4.3	5.8	$\cdots$	1.0
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
T		9.9	5.0	3.9	$\cdots$	2.8

Setup

Timestep	Algorithm 1	Best in hindsight Algorithm 2	Algorithm 3	⋯	Algorithm k
1 	2.8	9.3	0.3	⋯	1.4
2 	3.7	4.3	5.8	⋯	1.0
T 	9.9	5.0	3.9	⋯	2.8

Regret = (solution quality of best alg in hindsight) – (learner's reward)

$$= (9.3 + 4.3 + \cdots + 5.0) - (2.8 + 4.3 + \cdots + 2.8)$$

Regret

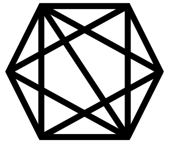
Regret = (solution quality of best alg in hindsight) – (learner's reward)
$= (9.3 + 4.3 + \cdots + 5.0) - (2.8 + 4.3 + \cdots + 2.8)$

Goal: $\frac{1}{T} \cdot$ (Regret) $\rightarrow 0$ as $T \rightarrow \infty$

On average, competing with best algorithm in hindsight

(Of course, model applies beyond algorithm selection as well)

Setup

	Solution quality				
Timestep	**Algorithm 1**	**Algorithm 2**	**Algorithm 3**	$\cdots$	**Algorithm k**
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
t 	$u_t(1)$	$u_t(2)$	$u_t(3)$	$\cdots$	$u_t(k)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$

$\boldsymbol{u}_t = \big(u_t(1), \ldots, u_t(k)\big) \in [0,1]^k$ (normalized for simplicity)

Outline

1. Statistical learning theory
2. Online learning
    i. Problem setup
    ii. Hedge algorithm
    iii. Online learning for MWIS
    iv. Additional learning models

Hedge algorithm [Freund, Schapire, JCSS'97]

input: Learning rate $\eta > 0$

initialization: $\boldsymbol{U}_0 = (0, \dots, 0)$ is the all-zeros vector of length k

for $t = 1, \dots, T$:

choose distribution $\boldsymbol{p}_t \in [0,1]^k$ such that $\underline{p_t(i) \propto \exp\big(\eta U_{t-1}(i)\big)}$

Initially, $\boldsymbol{p}_1 = \left(\frac{1}{k}, \dots, \frac{1}{k}\right)$

choose algorithm $\underline{i_t \sim \boldsymbol{p}_t\text{, receive reward } u_t(i_t)}$

Expected reward is $\langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle$

observe reward vector $\boldsymbol{u}_t$

update $\boldsymbol{U}_t = \boldsymbol{U}_{t-1} + u_t$

Hedge algorithm [Freund, Schapire, JCSS'97]

input: Learning rate $\eta > 0$

initialization: $\boldsymbol{U}_0 = (0, \dots, 0)$ is the all-zeros vector of length k

for $t = 1, \dots, T$:

choose distribution $\boldsymbol{p}_t \in [0,1]^k$ such that $\underline{p_t(i) \propto \exp\left(\eta U_{t-1}(i)\right)}$

Exponentially upweight high-reward algorithms

choose algorithm $\underline{i_t \sim \boldsymbol{p}_t, \text{ receive reward } u_t(i_t)}$

Expected reward is $\langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle$

observe reward vector $\boldsymbol{u}_t$

update $\boldsymbol{U}_t = \boldsymbol{U}_{t-1} + u_t$

Regret

Regret = (sol quality of best alg in hindsight) – (learner's reward)

$$= \max_{i \in [k]} \sum_{t=1}^{T} u_t(i) - \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle$$

$$i^* = \operatorname*{argmax}_{i \in [k]} \sum_{t=1}^{T} u_t(i)$$

Theorem: The regret of the Hedge algorithm is $\leq 3\sqrt{T \ln k}$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$W_t = \sum_{i=1}^{k} \exp\left(\eta U_t(i)\right)$$

$$\left(U_t(i) = \sum_{\tau=1}^{t} u_\tau(i)\right)$$

$$\frac{W_t}{W_{t-1}} = \frac{\sum_{i=1}^{k} \exp\left(\eta U_t(i)\right)}{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right)}$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$W_t = \sum_{i=1}^{k} \exp\left(\eta U_t(i)\right)$$

$$\left(U_t(i) = \sum_{\tau=1}^{t} u_\tau(i)\right)$$

$$\frac{W_t}{W_{t-1}} = \frac{\sum_{i=1}^{k} \exp\left(\eta U_t(i)\right)}{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right)}$$

$$= \frac{\sum_{i=1}^{k} \exp\left(\eta\left(U_{t-1}(i) + u_t(i)\right)\right)}{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right)}$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{W_t}{W_{t-1}} = \frac{\sum_{i=1}^{k} \exp\left(\eta\left(U_{t-1}(i) + u_t(i)\right)\right)}{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right)}$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{W_t}{W_{t-1}} = \frac{\sum_{i=1}^{k} \exp\left(\eta\left(U_{t-1}(i) + u_t(i)\right)\right)}{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right)}$$

$$= \frac{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right) \exp\left(\eta u_t(i)\right)}{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right)}$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{W_t}{W_{t-1}} = \frac{\sum_{i=1}^{k} \exp\left(\eta\left(U_{t-1}(i) + u_t(i)\right)\right)}{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right)}$$

$$= \frac{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right) \exp\left(\eta u_t(i)\right)}{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right)}$$

Remember: $p_t(i) \propto \exp\left(\eta U_{t-1}(i)\right)$, so $p_t(i) = \frac{\exp\left(\eta U_{t-1}(i)\right)}{\sum_{i=1}^{k} \exp\left(\eta U_{t-1}(i)\right)}$

$$\frac{W_t}{W_{t-1}} = \sum_{i=1}^{k} p_t(i) \exp\left(\eta u_t(i)\right)$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{W_t}{W_{t-1}} = \sum_{i=1}^{k} p_t(i) \exp\left(\eta u_t(i)\right)$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{W_t}{W_{t-1}} = \sum_{i=1}^{k} p_t(i) \exp\left(\eta u_t(i)\right)$$

Useful inequality: For $u \in [0,1]$ and $\eta > 0$, $e^{\eta u} \leq 1 + (e^{\eta} - 1)u$

$$\frac{W_t}{W_{t-1}} \leq \sum_{i=1}^{k} p_t(i)\left(1 + (e^{\eta} - 1)u_t(i)\right)$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{W_t}{W_{t-1}} = \sum_{i=1}^{k} p_t(i) \exp\left(\eta u_t(i)\right)$$

Useful inequality: For $u \in [0,1]$ and $\eta > 0$, $e^{\eta u} \leq 1 + (e^{\eta} - 1)u$

$$\frac{W_t}{W_{t-1}} \leq \sum_{i=1}^{k} p_t(i)\left(1 + (e^{\eta} - 1)u_t(i)\right)$$

$$= 1 + (e^{\eta} - 1)\langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{W_t}{W_{t-1}} \leq 1 + (e^\eta - 1)\langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle$$

Useful inequality: $1 + z \leq e^z, \forall z \in \mathbb{R}$

$$\frac{W_t}{W_{t-1}} \leq \exp\left((e^\eta - 1)\langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle\right)$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{W_t}{W_{t-1}} \leq 1 + (e^\eta - 1)\langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle$$

Useful inequality: $1 + z \leq e^z, \forall z \in \mathbb{R}$

$$\frac{W_t}{W_{t-1}} \leq \exp\left((e^\eta - 1)\langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle\right)$$

$$\frac{W_T}{W_0} = \frac{W_1}{W_0} \cdot \frac{W_2}{W_1} \cdots \frac{W_T}{W_{T-1}} \leq \exp\left((e^\eta - 1) \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle \right)$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{W_T}{W_0} \leq \exp\left((e^\eta - 1) \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle \right)$$

$$W_T = \sum_{i=1}^{k} \exp\big(\eta U_T(i)\big) \geq \exp\big(\eta U_T(i^*)\big)$$

$$W_0 = \sum_{i=1}^{k} \exp\big(\eta U_0(i)\big) = \sum_{i=1}^{k} \exp(\eta \cdot 0) = k$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{\exp\left(\eta U_T(i^*)\right)}{k} \leq \frac{W_T}{W_0} \leq \exp\left((e^\eta - 1) \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle \right)$$

$$W_T = \sum_{i=1}^{k} \exp\left(\eta U_T(i)\right) \geq \exp\left(\eta U_T(i^*)\right)$$

$$W_0 = \sum_{i=1}^{k} \exp\left(\eta U_0(i)\right) = \sum_{i=1}^{k} \exp(\eta \cdot 0) = k$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{\exp\left(\eta U_T(i^*)\right)}{k} \leq \frac{W_T}{W_0} \leq \exp\left((e^\eta - 1) \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle \right)$$

$$U_T(i^*) \leq \frac{e^\eta - 1}{\eta} \cdot \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle + \frac{\ln k}{\eta}$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\frac{\exp\left(\eta U_T(i^*)\right)}{k} \leq \frac{W_T}{W_0} \leq \exp\left((e^\eta - 1)\sum_{t=1}^{T}\langle \boldsymbol{p}_t, \boldsymbol{u}_t\rangle\right)$$

$$U_T(i^*) \leq \frac{e^\eta - 1}{\eta} \cdot \sum_{t=1}^{T}\langle \boldsymbol{p}_t, \boldsymbol{u}_t\rangle + \frac{\ln k}{\eta}$$

$$\sum_{t=1}^{T} u_t(i^*) \leq \frac{e^\eta - 1}{\eta} \cdot \sum_{t=1}^{T}\langle \boldsymbol{p}_t, \boldsymbol{u}_t\rangle + \frac{\ln k}{\eta}$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\sum_{t=1}^{T} u_t(i^*) \leq \frac{e^\eta - 1}{\eta} \cdot \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle + \frac{\ln k}{\eta}$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\sum_{t=1}^{T} u_t(i^*) \leq \frac{e^\eta - 1}{\eta} \cdot \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle + \frac{\ln k}{\eta}$$

$$\text{regret} = \sum_{t=1}^{T} u_t(i^*) - \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle \leq \frac{e^\eta - 1 - \eta}{\eta} \cdot \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle + \frac{\ln k}{\eta}$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\sum_{t=1}^{T} u_t(i^*) \leq \frac{e^\eta - 1}{\eta} \cdot \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle + \frac{\ln k}{\eta}$$

$$\text{regret} = \sum_{t=1}^{T} u_t(i^*) - \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle \leq \frac{e^\eta - 1 - \eta}{\eta} \cdot \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle + \frac{\ln k}{\eta}$$

$$\leq \frac{e^\eta - 1 - \eta}{\eta} \cdot T + \frac{\ln k}{\eta}$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\text{regret} = \sum_{t=1}^{T} u_t(i^*) - \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle \leq \frac{e^{\eta} - 1 - \eta}{\eta} \cdot T + \frac{\ln k}{\eta}$$

Proof that Hedge's regret is $O\left(\sqrt{T \ln k}\right)$

$$\text{regret} = \sum_{t=1}^{T} u_t(i^*) - \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{u}_t \rangle \leq \frac{e^\eta - 1 - \eta}{\eta} \cdot T + \frac{\ln k}{\eta}$$

Useful inequality: For $\eta \in [0,1]$, $e^\eta - 1 \leq 2\eta$

$$\text{regret} \leq 2\eta T + \frac{\ln k}{\eta}$$

Setting $\eta = \sqrt{\frac{\ln k}{T}}$, we have that $\textbf{regret} \leq \mathbf{3}\sqrt{\boldsymbol{T} \ln \boldsymbol{k}}$

Outline

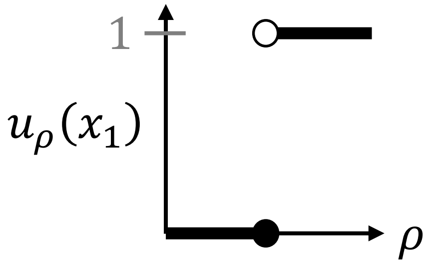
1. Statistical learning theory
2. Online learning
    i. Problem setup
    ii. Hedge algorithm
    iii. **Online learning for MWIS**
    iv. Additional learning models

Worst-case MWIS instance

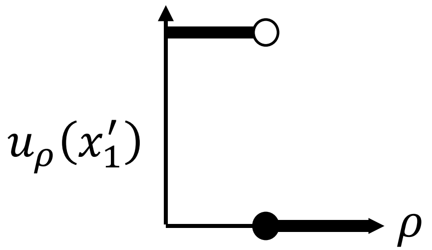
Exists adversary choosing MWIS instances s.t.:

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

Round 1:



Utility on instance x_1 as a function of ρ



Utility on instance x_1' as a function of ρ

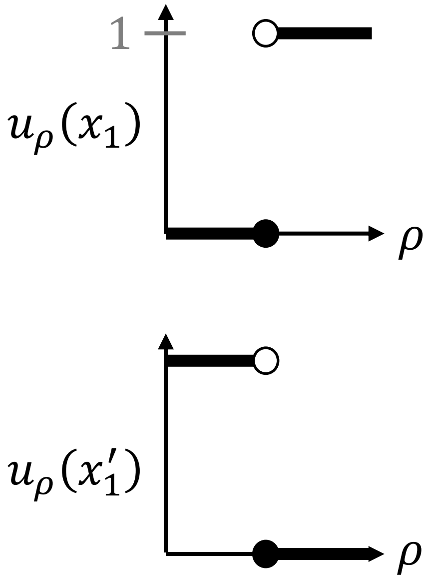
Gupta and Roughgarden, ITCS'16

Worst-case MWIS instance

Exists adversary choosing MWIS instances s.t.:

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

Round 1:



Adversary chooses x_1 or x_1' with equal probability

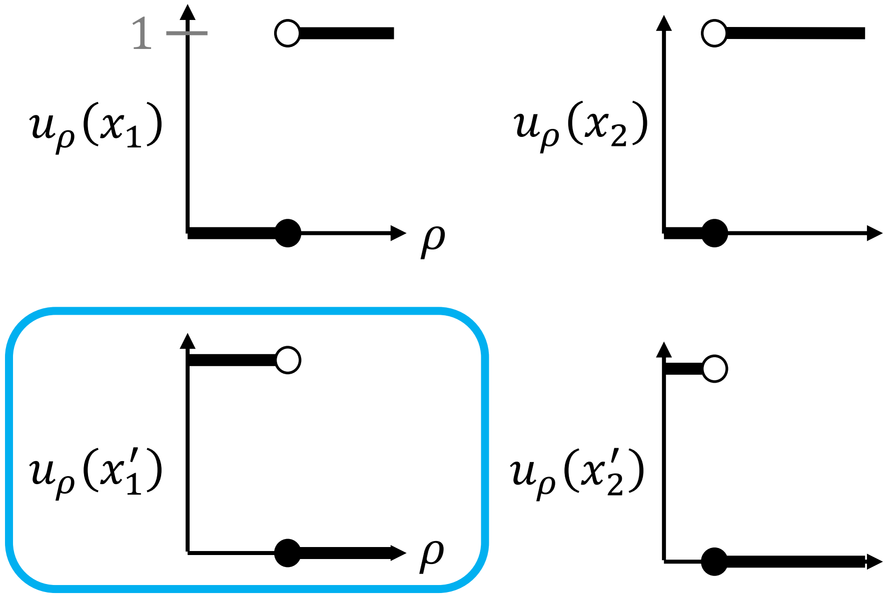
Gupta and Roughgarden, ITCS'16

Worst-case MWIS instance

Exists adversary choosing MWIS instances s.t.:

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

Round 1: Round 2:



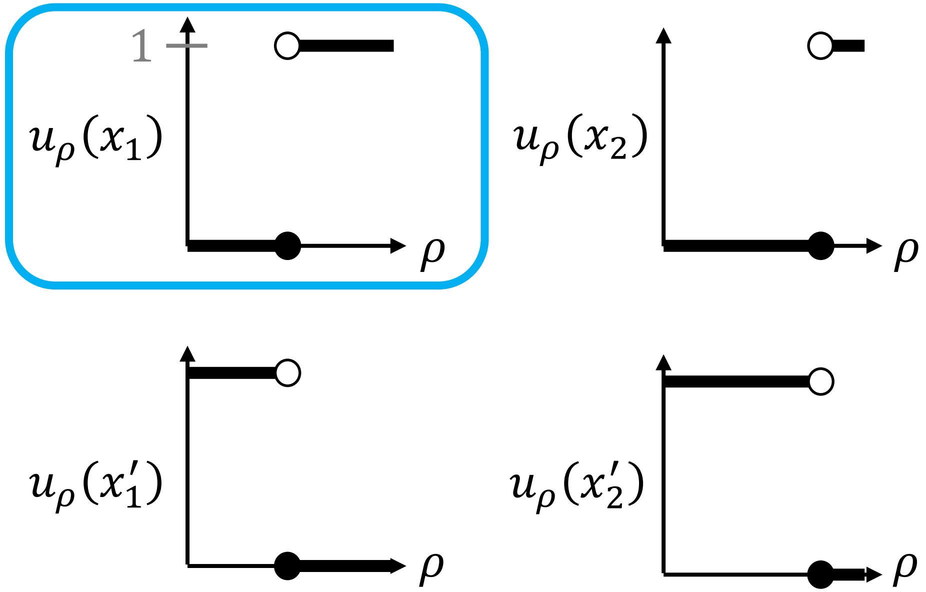
Gupta and Roughgarden, ITCS'16

Worst-case MWIS instance

Exists adversary choosing MWIS instances s.t.:

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

Round 1: Round 2:



Repeatedly halves optimal region

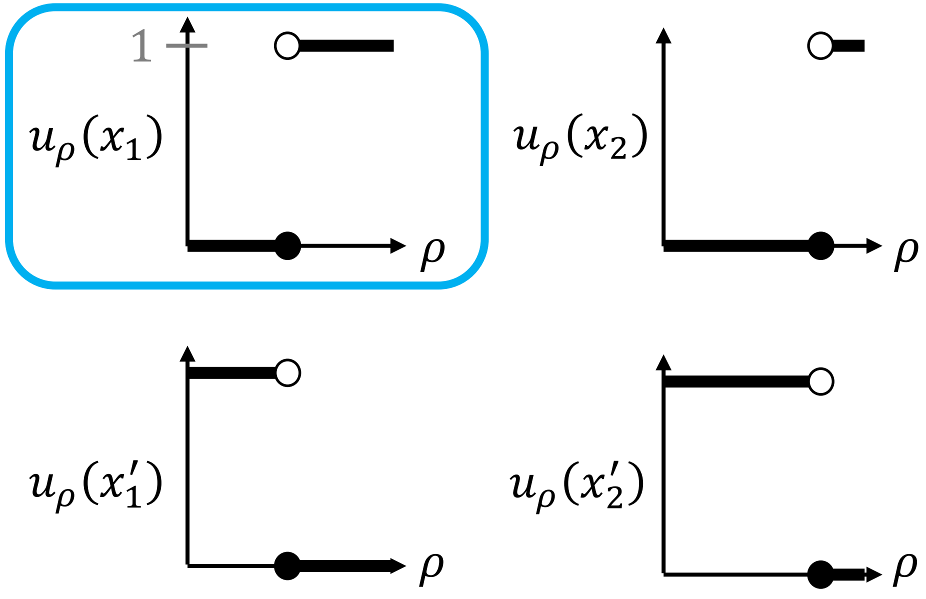
Gupta and Roughgarden, ITCS'16

Worst-case MWIS instance

Exists adversary choosing MWIS instances s.t.:

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

Round 1: Round 2:



Repeatedly halves optimal region

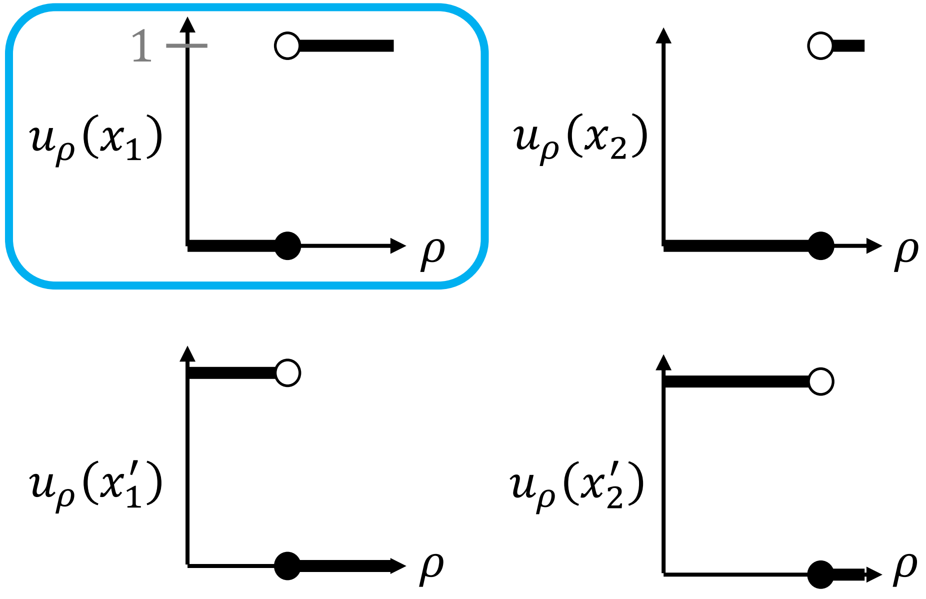
Gupta and Roughgarden, ITCS'16

Worst-case MWIS instance

Exists adversary choosing MWIS instances 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 ρ in hindsight: T

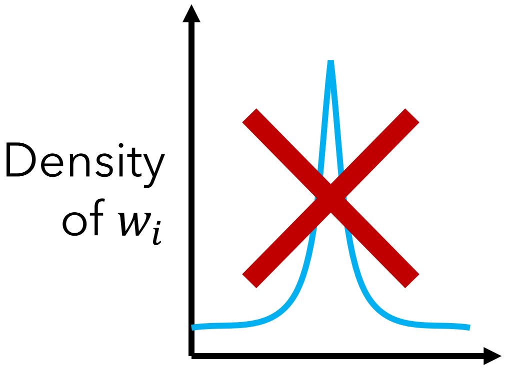
Expected regret $= \frac{T}{2}$

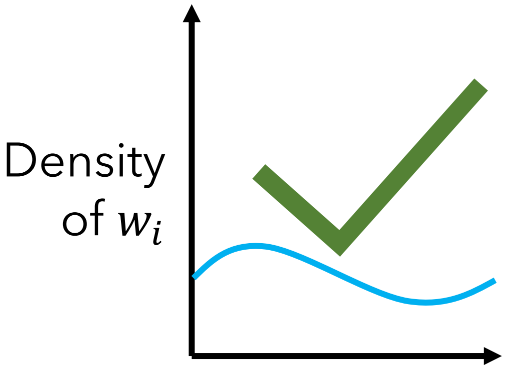
Gupta and Roughgarden, ITCS'16

Smoothed adversary

Sub-linear regret is possible if adversary has a "shaky hand":

- $w_1, \ldots, w_n, k_1, \ldots, k_n$ are stochastic
- Joint density of (w_i, w_j, k_i, k_j) is bounded





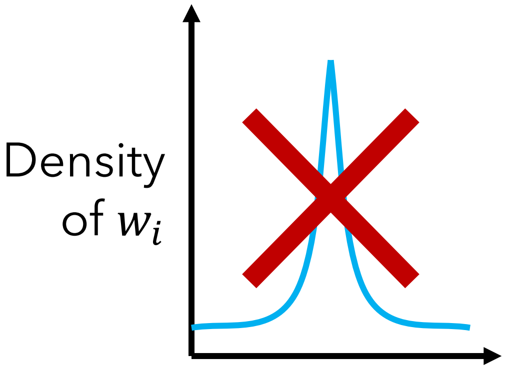
In this case, discretize and run Hedge

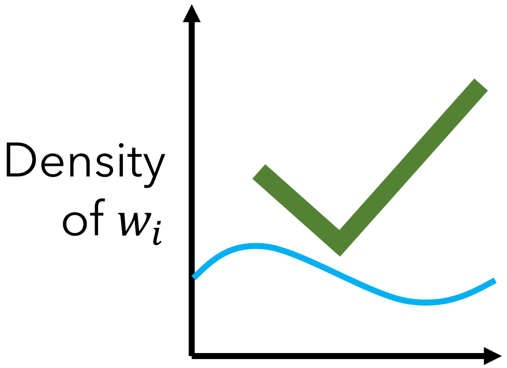
Gupta and Roughgarden, ITCS'16

Smoothed adversary

Sub-linear regret is possible if adversary has a "shaky hand":

- $w_1, \ldots, w_n, k_1, \ldots, k_n$ are stochastic
- Joint density of (w_i, w_j, k_i, k_j) is bounded





Later generalized by Cohen-Addad, Kanade [AISTATS, '17]; Balcan, Dick, Vitercik [FOCS'18]; Balcan et al. [UAI'20]; …

Gupta and Roughgarden, ITCS'16

Outline

1. Statistical learning theory
2. Online learning
   i. Problem setup
   ii. Hedge algorithm
   iii. Online learning for MWIS
   iv. Additional learning models

Other models

- **Full information:** Learner sees all runtimes
  - *Focus of this lecture*
- **Bandit:** Learner only sees runtime of chosen algorithm
  - E.g., Balcan, Dick, Vitercik, FOCS'18
- **Semi-bandit:** Mixture of the two
  - E.g., Balcan, Dick, Pegden, UAI'20
- **Continuous parameters** (piecewise-Lipschitz performance)
  - E.g., Gupta, Roughgarden, ITCS'16; Cohen-Addad, Kanade, AISTATS, '17; Balcan, Dick, Vitercik, FOCS'18; …