Bandits, Query Learning, and the Haystack Dimension

Kareem Amin Michael Kearns Umar Syed

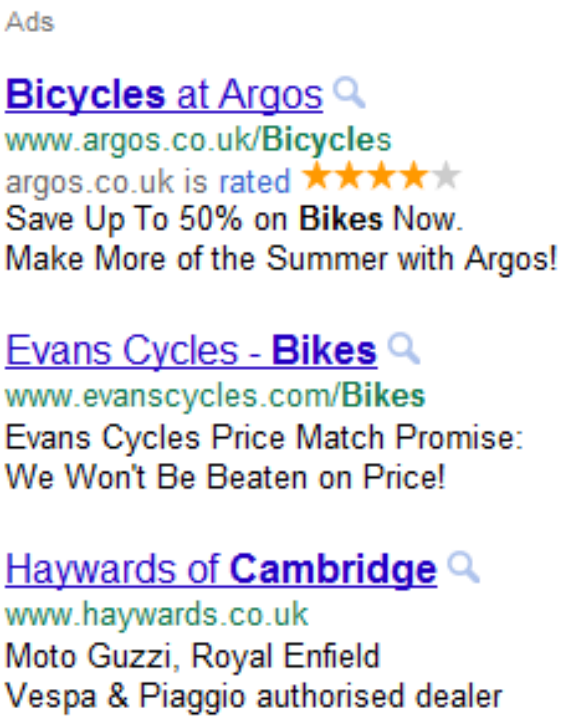
Computer and Information Science
University of Pennsylvania

Motivation

- *Problem*: Stochastic Multi-Armed Bandits with very large number of actions.
- *Motivation*: Search engine must choose results to display from very large set of advertisements.

Motivation

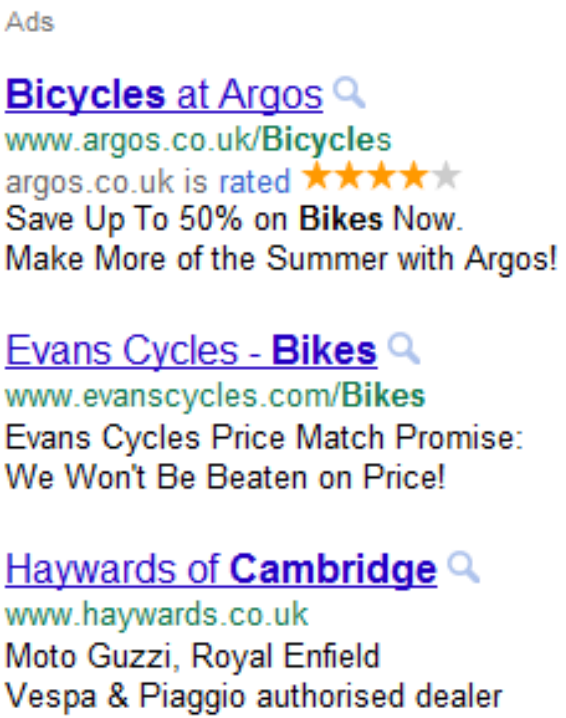
- *Problem*: Stochastic Multi-Armed Bandits with very large number of actions.
- *Motivation*: Search engine must choose results to display from very large set of advertisements.



- Assumptions are required to acheive sublinear regret in the number of *actions*.
- *Goal*: Characterize difficulty of acheiving sublinear regret.

Motivation

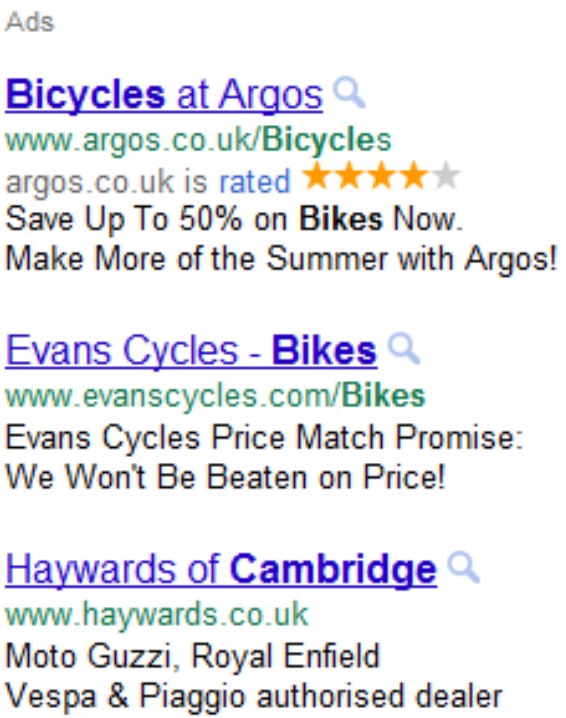
- *Problem*: Stochastic Multi-Armed Bandits with very large number of actions.
- *Motivation*: Search engine must choose results to display from very large set of advertisements.



- Assumptions are required to acheive sublinear regret in the number of *actions*.
- *Goal*: Characterize difficulty of acheiving sublinear regret.

Motivation

- *Problem*: Stochastic Multi-Armed Bandits with very large number of actions.
- *Motivation*: Search engine must choose results to display from very large set of advertisements.



- Assumptions are required to acheive sublinear regret in the number of *actions*.
- *Goal*: Characterize difficulty of acheiving sublinear regret.

Problem Formulation

- Functional Multi-Armed Bandits (fMAB)
- Expected payoff of each action $x \in \mathcal{X}$ is given by an unknown function $f^*(x)$.
- Assume that f^* belongs to a *known* function class $\mathcal{F}$.
- Previous works have studied particular classes $\mathcal{F}$.
  - Lipschitz functions. [Kleinberg, Slivkins, Upfal 2008], [Bubeck, Munos, Stoltz, Szepesvari 2009], [Maillard, Munos 2010]
  - Convex. [Flaxman, Kalai, McMahan 2004], [Kleinberg 2004]
  - Linear. [Awerbuch, Kleinberg 2004], [McMahan, Blum 2004], [Dani, Hayes 2006] [Abernethy, Hazan, Rakhlin 2008]
  - Drawn from a Gaussian Process. [Srinivas, Krause, Kakade, Seeger 2010]
- *Our Contribution*: Upper and lower bounds on regret for any $\mathcal{F}$.

Problem Formulation

- Functional Multi-Armed Bandits (fMAB)
- Expected payoff of each action $x \in \mathcal{X}$ is given by an unknown function $f^*(x)$.
- Assume that f^* belongs to a *known* function class $\mathcal{F}$.
- Previous works have studied particular classes $\mathcal{F}$.
  - Lipschitz functions. [Kleinberg, Slivkins, Upfal 2008], [Bubeck, Munos, Stoltz, Szepesvari 2009], [Maillard, Munos 2010]
  - Convex. [Flaxman, Kalai, McMahan 2004], [Kleinberg 2004]
  - Linear. [Awerbuch, Kleinberg 2004], [McMahan, Blum 2004], [Dani, Hayes 2006] [Abernethy, Hazan, Rakhlin 2008]
  - Drawn from a Gaussian Process. [Srinivas, Krause, Kakade, Seeger 2010]
- *Our Contribution*: Upper and lower bounds on regret for any $\mathcal{F}$.

Problem Formulation

- Functional Multi-Armed Bandits (fMAB)
- Expected payoff of each action $x \in \mathcal{X}$ is given by an unknown function $f^*(x)$.
- Assume that f^* belongs to a *known* function class $\mathcal{F}$.
- Previous works have studied particular classes $\mathcal{F}$.
  - Lipschitz functions. [Kleinberg, Slivkins, Upfal 2008], [Bubeck, Munos, Stoltz, Szepesvari 2009], [Maillard, Munos 2010]
  - Convex. [Flaxman, Kalai, McMahan 2004], [Kleinberg 2004]
  - Linear. [Awerbuch, Kleinberg 2004], [McMahan, Blum 2004], [Dani, Hayes 2006] [Abernethy, Hazan, Rakhlin 2008]
  - Drawn from a Gaussian Process. [Srinivas, Krause, Kakade, Seeger 2010]
- *Our Contribution*: Upper and lower bounds on regret for any $\mathcal{F}$.

Problem Formulation

- Functional Multi-Armed Bandits (fMAB)
- Expected payoff of each action $x \in \mathcal{X}$ is given by an unknown function $f^*(x)$.
- Assume that f^* belongs to a *known* function class $\mathcal{F}$.
- Previous works have studied particular classes $\mathcal{F}$.
  - Lipschitz functions. [Kleinberg, Slivkins, Upfal 2008], [Bubeck, Munos, Stoltz, Szepesvari 2009], [Maillard, Munos 2010]
  - Convex. [Flaxman, Kalai, McMahan 2004], [Kleinberg 2004]
  - Linear. [Awerbuch, Kleinberg 2004], [McMahan, Blum 2004], [Dani, Hayes 2006] [Abernethy, Hazan, Rakhlin 2008]
  - Drawn from a Gaussian Process. [Srinivas, Krause, Kakade, Seeger 2010]
- *Our Contribution*: Upper and lower bounds on regret for any $\mathcal{F}$.

Problem Formulation

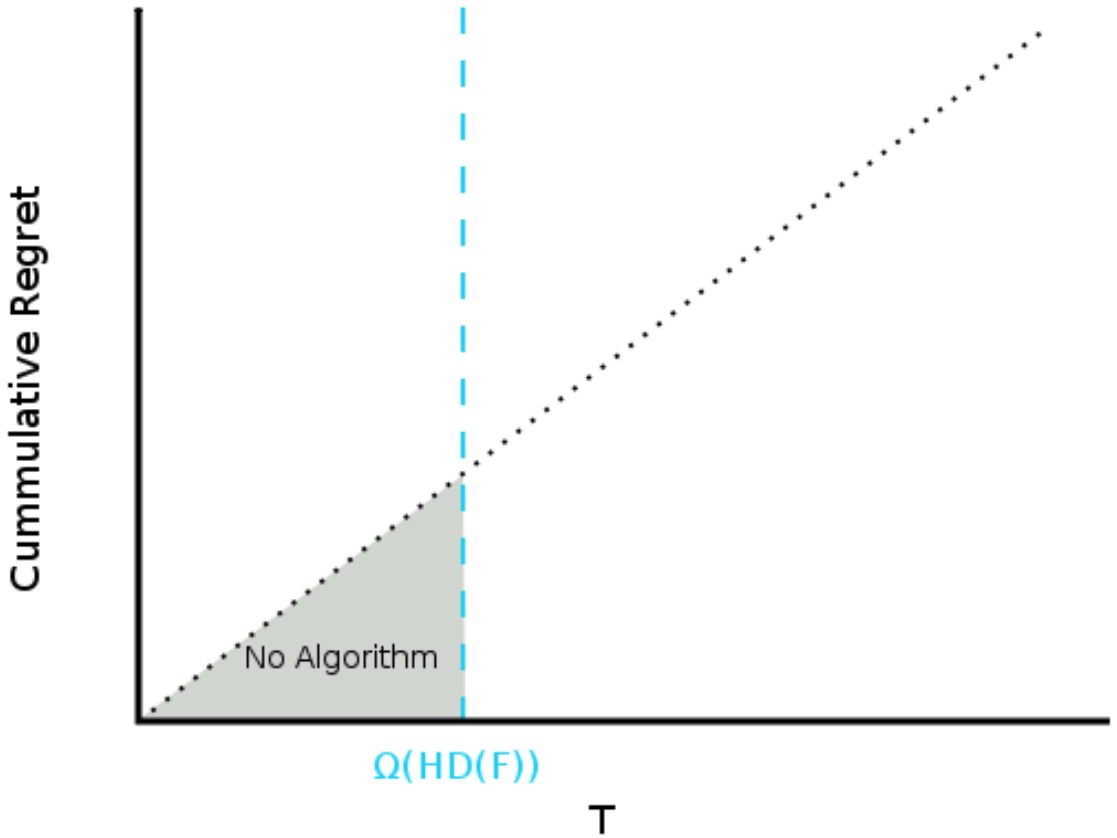
- Functional Multi-Armed Bandits (fMAB)
- Expected payoff of each action $x \in \mathcal{X}$ is given by an unknown function $f^*(x)$.
- Assume that f^* belongs to a *known* function class $\mathcal{F}$.
- Previous works have studied particular classes $\mathcal{F}$.
  - Lipschitz functions. [Kleinberg, Slivkins, Upfal 2008], [Bubeck, Munos, Stoltz, Szepesvari 2009], [Maillard, Munos 2010]
  - Convex. [Flaxman, Kalai, McMahan 2004], [Kleinberg 2004]
  - Linear. [Awerbuch, Kleinberg 2004], [McMahan, Blum 2004], [Dani, Hayes 2006] [Abernethy, Hazan, Rakhlin 2008]
  - Drawn from a Gaussian Process. [Srinivas, Krause, Kakade, Seeger 2010]
- *Our Contribution*: Upper and lower bounds on regret for any $\mathcal{F}$.

Problem Formulation

- Functional Multi-Armed Bandits (fMAB)
- Expected payoff of each action $x \in \mathcal{X}$ is given by an unknown function $f^*(x)$.
- Assume that f^* belongs to a *known* function class $\mathcal{F}$.
- Previous works have studied particular classes $\mathcal{F}$.
  - Lipschitz functions. [Kleinberg, Slivkins, Upfal 2008], [Bubeck, Munos, Stoltz, Szepesvari 2009], [Maillard, Munos 2010]
  - Convex. [Flaxman, Kalai, McMahan 2004], [Kleinberg 2004]
  - Linear. [Awerbuch, Kleinberg 2004], [McMahan, Blum 2004], [Dani, Hayes 2006] [Abernethy, Hazan, Rakhlin 2008]
  - Drawn from a Gaussian Process. [Srinivas, Krause, Kakade, Seeger 2010]
- *Our Contribution*: Upper and lower bounds on regret for any $\mathcal{F}$.

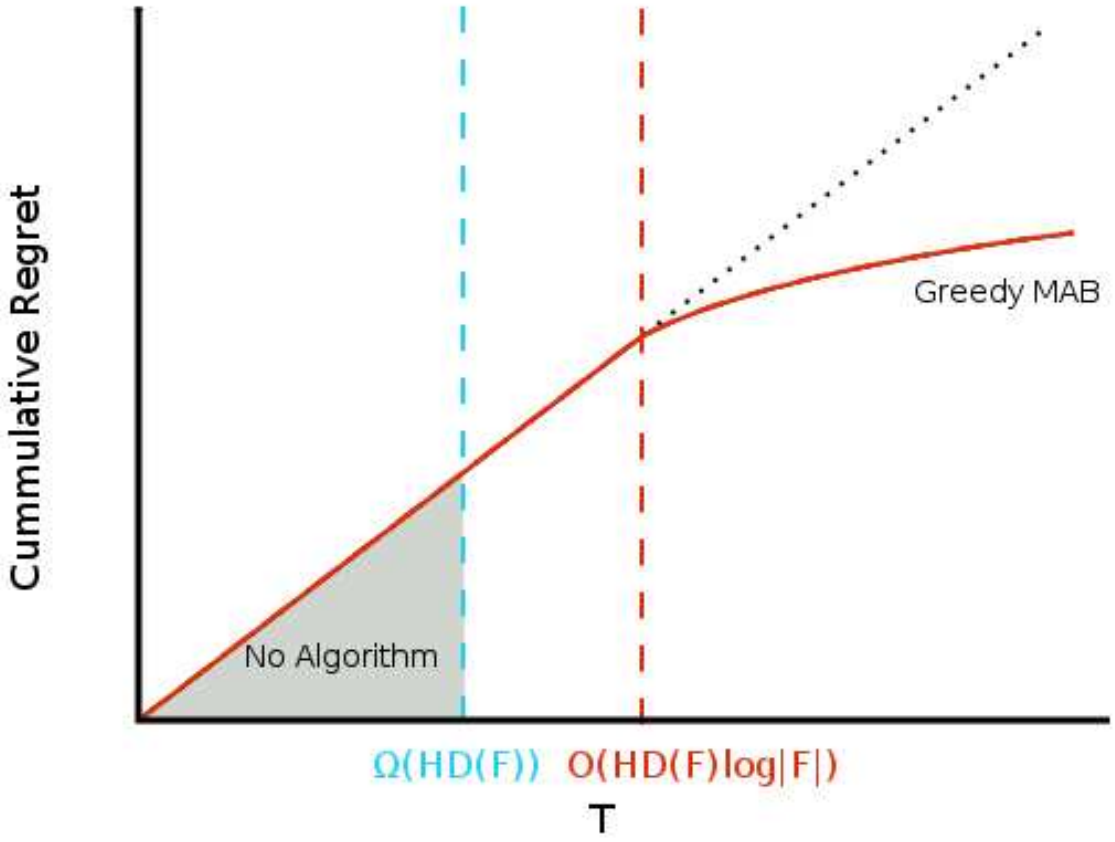
Main Results, Pictorially

We introduce a new measure of complexity called the *Haystack Dimension*, denoted $HD(\mathcal{F})$.



Main Results, Pictorially

For finite $\mathcal{F}$, results are tight.



Maximizing from Queries

- MAB Problem embeds an instance of Maximizing from Queries (MAX) problem.
- Exact payoff given by unknown f^* from known function class $\mathcal{F}$.
- Algorithm's Goal: minimize rounds needed to select action in X^{f^*}.
  - X^{f^*} are actions that maximize f^*. ($X^{f^*} \neq \emptyset$).
- *Worst-case query complexity* for MAX related to the difficulty of achieving no regret in MAB.
- Until MAB algorithm plays an action in X^{f^*} at least once, it suffers regret.

Maximizing from Queries

- MAB Problem embeds an instance of Maximizing from Queries (MAX) problem.
- Exact payoff given by unknown f^* from known function class $\mathcal{F}$.
- Algorithm's Goal: minimize rounds needed to select action in X^{f^*}.
  - X^{f^*} are actions that maximize f^*. ($X^{f^*} \neq \emptyset$).
- *Worst-case query complexity* for MAX related to the difficulty of achieving no regret in MAB.
- Until MAB algorithm plays an action in X^{f^*} at least once, it suffers regret.

Maximizing from Queries

- MAB Problem embeds an instance of Maximizing from Queries (MAX) problem.
- Exact payoff given by unknown f^* from known function class $\mathcal{F}$.
- Algorithm's Goal: minimize rounds needed to select action in X^{f^*}.
  - X^{f^*} are actions that maximize f^*. ($X^{f^*} \neq \emptyset$).
- *Worst-case query complexity* for MAX related to the difficulty of achieving no regret in MAB.
- Until MAB algorithm plays an action in X^{f^*} at least once, it suffers regret.

Maximizing from Queries

- MAB Problem embeds an instance of Maximizing from Queries (MAX) problem.
- Exact payoff given by unknown f^* from known function class $\mathcal{F}$.
- Algorithm's Goal: minimize rounds needed to select action in X^{f^*}.
  - X^{f^*} are actions that maximize f^*. ($X^{f^*} \neq \emptyset$).
- *Worst-case query complexity* for MAX related to the difficulty of achieving no regret in MAB.
- Until MAB algorithm plays an action in X^{f^*} at least once, it suffers regret.

Maximizing from Queries

- MAB Problem embeds an instance of Maximizing from Queries (MAX) problem.
- Exact payoff given by unknown f^* from known function class $\mathcal{F}$.
- Algorithm's Goal: minimize rounds needed to select action in X^{f^*}.
  - X^{f^*} are actions that maximize f^*. ($X^{f^*} \neq \emptyset$).
- *Worst-case query complexity* for MAX related to the difficulty of achieving no regret in MAB.
- Until MAB algorithm plays an action in X^{f^*} at least once, it suffers regret.

Haystack Dimension Intuition

Intuitions for the *Haystack Dimension* $HD(\mathcal{F})$

- MAX Problem
- Some function classes $\mathcal{F}$ don't require learning f^* completely.
- Example: $\arg\max_x f(x)$ is the same for all $f \in \mathcal{F}$.
- But sometimes identifying the true function f^* is necessary.
- In general, we show that an optimal algorithm must balance these two strategies.
- The Haystack Dimension characterizes how to balance these strategies.

Haystack Dimension Intuition

Intuitions for the *Haystack Dimension* $HD(\mathcal{F})$

- MAX Problem
- Some function classes $\mathcal{F}$ don't require learning f^* completely.
- Example: $\arg\max_x f(x)$ is the same for all $f \in \mathcal{F}$.
- But sometimes identifying the true function f^* is necessary.
- In general, we show that an optimal algorithm must balance these two strategies.
- The Haystack Dimension characterizes how to balance these strategies.

Haystack Dimension Intuition

Intuitions for the *Haystack Dimension* $HD(\mathcal{F})$

- MAX Problem
- Some function classes $\mathcal{F}$ don't require learning f^* completely.
- Example: $\arg\max_x f(x)$ is the same for all $f \in \mathcal{F}$.
- But sometimes identifying the true function f^* is necessary.
- In general, we show that an optimal algorithm must balance these two strategies.
- The Haystack Dimension characterizes how to balance these strategies.

Haystack Dimension Intuition

Intuitions for the *Haystack Dimension* $HD(\mathcal{F})$

- MAX Problem
- Some function classes $\mathcal{F}$ don't require learning f^* completely.
- Example: $\arg\max_x f(x)$ is the same for all $f \in \mathcal{F}$.
- But sometimes identifying the true function f^* is necessary.
- In general, we show that an optimal algorithm must balance these two strategies.
- The Haystack Dimension characterizes how to balance these strategies.

Haystack Dimension Intuition

Intuitions for the *Haystack Dimension* $HD(\mathcal{F})$

- MAX Problem
- Some function classes $\mathcal{F}$ don't require learning f^* completely.
- Example: $\arg\max_x f(x)$ is the same for all $f \in \mathcal{F}$.
- But sometimes identifying the true function f^* is necessary.
- In general, we show that an optimal algorithm must balance these two strategies.
- The Haystack Dimension characterizes how to balance these strategies.

Haystack Dimension Intuition

Intuitions for the *Haystack Dimension* $HD(\mathcal{F})$

- MAX Problem
- Some function classes $\mathcal{F}$ don't require learning f^* completely.
- Example: $\arg\max_x f(x)$ is the same for all $f \in \mathcal{F}$.
- But sometimes identifying the true function f^* is necessary.
- In general, we show that an optimal algorithm must balance these two strategies.
- The Haystack Dimension characterizes how to balance these strategies.

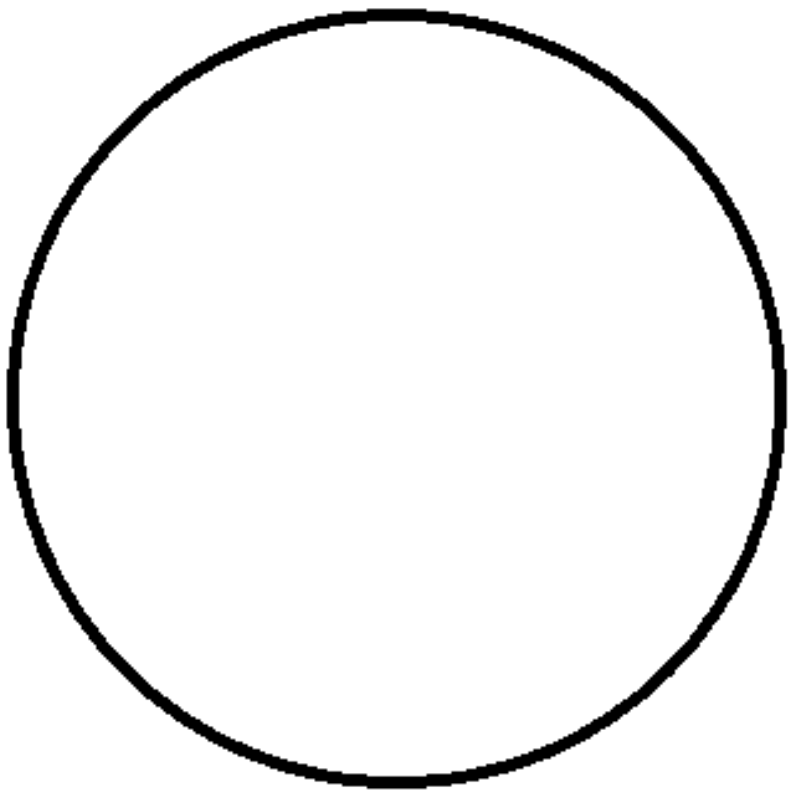
Haystack Dimension Intuition

Intuitions for the *Haystack Dimension* $HD(\mathcal{F})$

- MAX Problem
- Some function classes $\mathcal{F}$ don't require learning f^* completely.
- Example: $\arg\max_x f(x)$ is the same for all $f \in \mathcal{F}$.
- But sometimes identifying the true function f^* is necessary.
- In general, we show that an optimal algorithm must balance these two strategies.
- The Haystack Dimension characterizes how to balance these strategies.

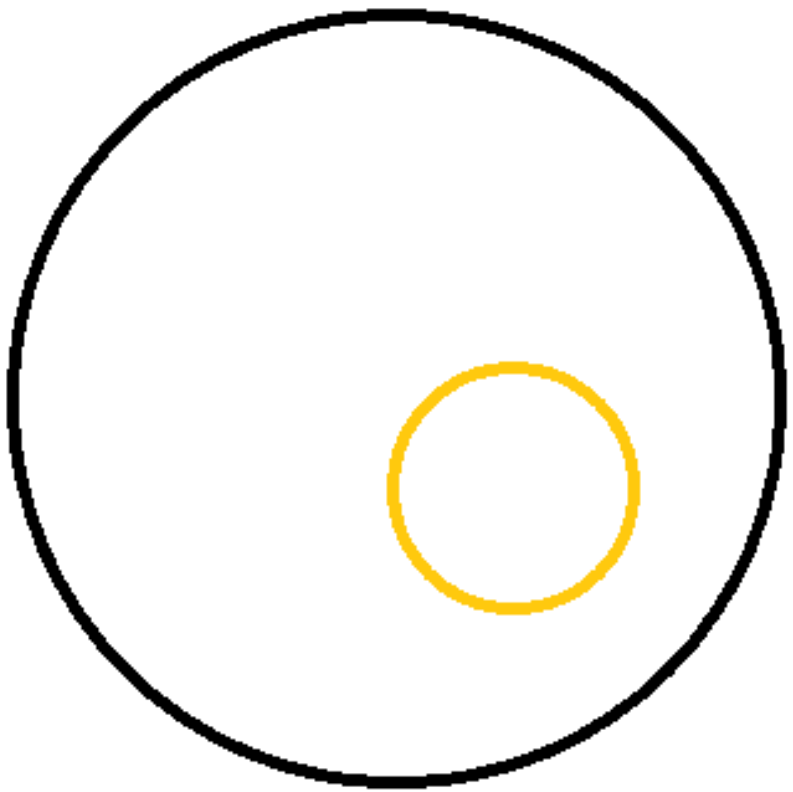
More Haystack Dimension Intuition

Consider $\mathcal{F}$:



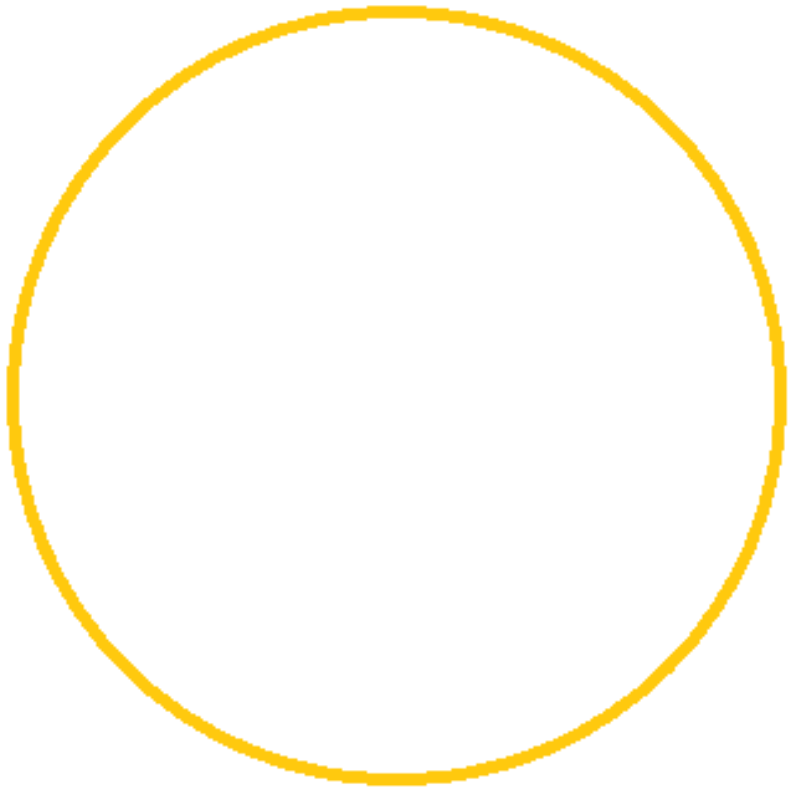
More Haystack Dimension Intuition

Consider a finite subset $G \subseteq \mathcal{F}$:



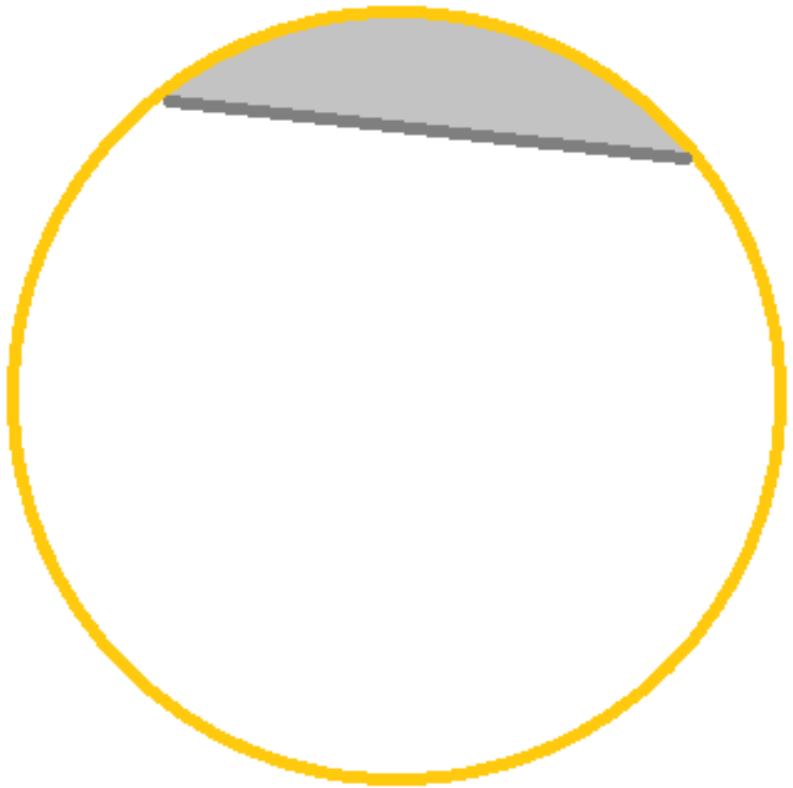
More Haystack Dimension Intuition

Focus on G and an action x:



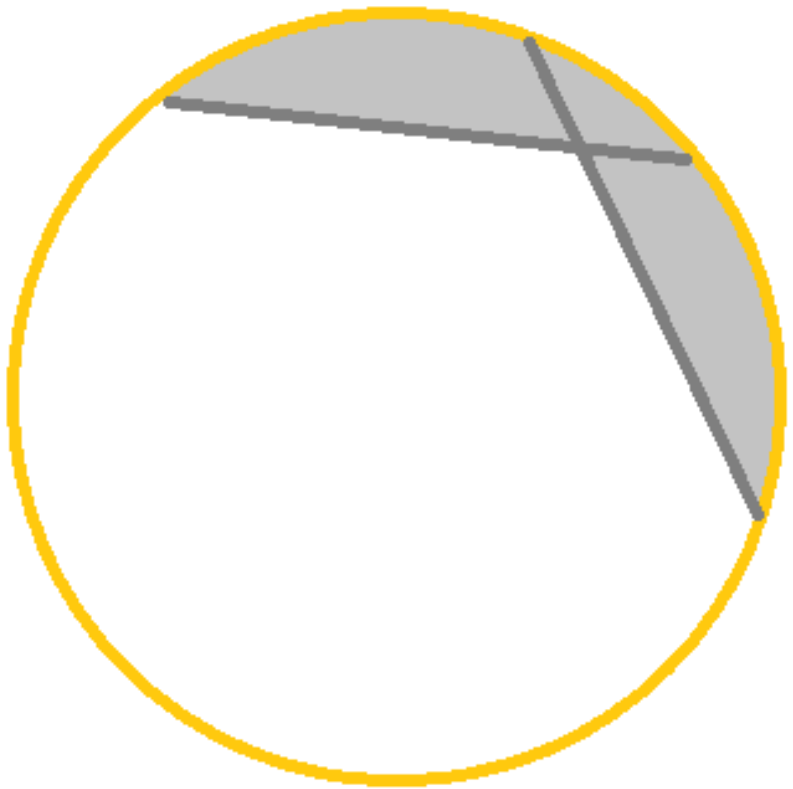
More Haystack Dimension Intuition

x maximizes some functions in G:



More Haystack Dimension Intuition

Upon observing $f^*(x)$, x also eliminates some functions in G as inconsistent:



More Haystack Dimension Intuition

- Shaded region for some $G \subseteq \mathcal{F}$ is small, G is "tricky."
- *Haystack Dimension*: Identifies the trickiest such subset.

The Haystack Dimension, Precisely

- Fix a finite $G \subset \mathcal{F}$.
- $G(x)$ is the *maximized set*, the functions in G that would be maximized by playing x.
  - $G(x) \triangleq \{f \in G : x \in X^f\}$.
- $G(\langle x, y \rangle)$ is the *inconsistent set* for action-value pair $\langle x, y \rangle$.
  - $G(\langle x, y \rangle) \triangleq \{f \in G : f(x) \neq y\}$.

The Haystack Dimension, Precisely

- Fix a finite $G \subset \mathcal{F}$.
- $G(x)$ is the *maximized set*, the functions in G that would be maximized by playing x.
  - $G(x) \triangleq \{f \in G : x \in X^f\}$.
- $G(\langle x, y \rangle)$ is the *inconsistent set* for action-value pair $\langle x, y \rangle$.
  - $G(\langle x, y \rangle) \triangleq \{f \in G : f(x) \neq y\}$.

The Haystack Dimension, Precisely

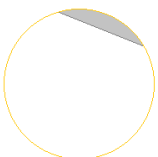
- Fix a finite $G \subset \mathcal{F}$.
- $G(x)$ is the *maximized set*, the functions in G that would be maximized by playing x.
  - $G(x) \triangleq \{f \in G : x \in X^f\}$.
- $G(\langle x, y \rangle)$ is the *inconsistent set* for action-value pair $\langle x, y \rangle$.
  - $G(\langle x, y \rangle) \triangleq \{f \in G : f(x) \neq y\}$.

The Haystack Dimension, Precisely

- Fix a finite $G \subset \mathcal{F}$.
- $G(x)$ is the *maximized set*, the functions in G that would be maximized by playing x.
  - $G(x) \triangleq \{f \in G : x \in X^f\}$.
- $G(\langle x, y \rangle)$ is the *inconsistent set* for action-value pair $\langle x, y \rangle$.
  - $G(\langle x, y \rangle) \triangleq \{f \in G : f(x) \neq y\}$.

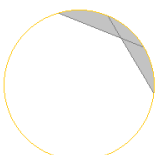
The Haystack Dimension, Precisely

- Fix a finite $G \subset \mathcal{F}$.
- $G(x)$ is the *maximized set*, the functions in G that would be maximized by playing x.



  - $G(x) \triangleq \{f \in G : x \in X^f\}$.

- $G(\langle x, y \rangle)$ is the *inconsistent set* for action-value pair $\langle x, y \rangle$.



  - $G(\langle x, y \rangle) \triangleq \{f \in G : f(x) \neq y\}$.

The Haystack Dimension, Precisely

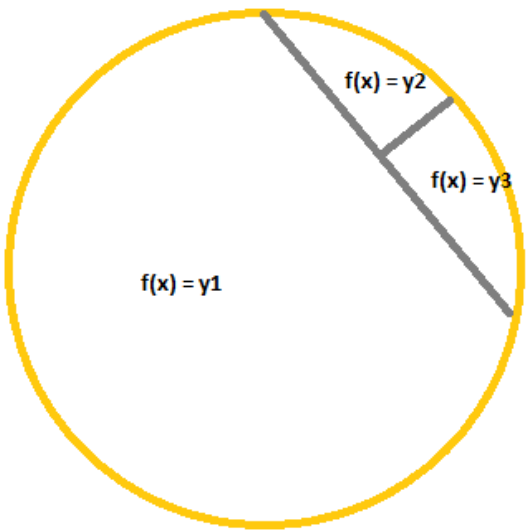
- After playing x and observing y, an algorithm can effectively *eliminate* the functions in $G(x) \cup G(\langle x, y \rangle)$.
- Let $y^*_G(x)$ be the worst-case y.
  - $y^*_G(x) \triangleq \arg\inf_y |G(x) \cup G(\langle x, y \rangle)|$

The Haystack Dimension, Precisely

- After playing x and observing y, an algorithm can effectively *eliminate* the functions in $G(x) \cup G(\langle x, y \rangle)$.
- Let $y^*_G(x)$ be the worst-case y.
  - $y^*_G(x) \triangleq \arg\inf_y |G(x) \cup G(\langle x, y \rangle)|$

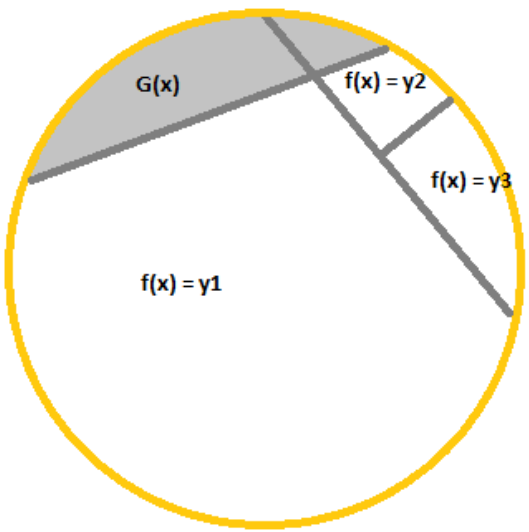
The Haystack Dimension, Precisely

- After playing x and observing y, an algorithm can effectively *eliminate* the functions in $G(x) \cup G(\langle x, y \rangle)$.
- Let $y^*_G(x)$ be the worst-case y.
  - $y^*_G(x) \triangleq \arg\inf_y |G(x) \cup G(\langle x, y \rangle)|$



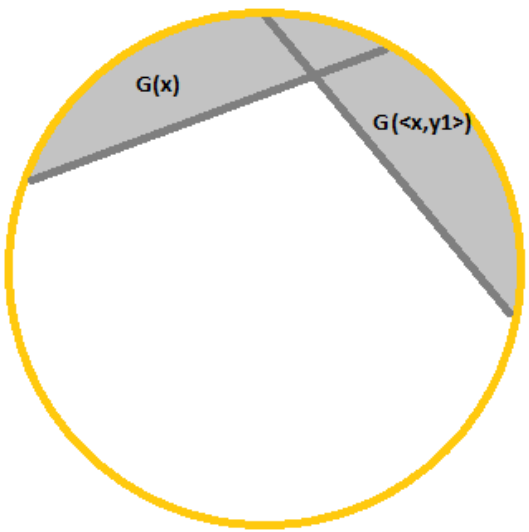
The Haystack Dimension, Precisely

- After playing x and observing y, an algorithm can effectively *eliminate* the functions in $G(x) \cup G(\langle x, y \rangle)$.
- Let $y^*_G(x)$ be the worst-case y.
  - $y^*_G(x) \triangleq \arg\inf_y |G(x) \cup G(\langle x, y \rangle)|$



The Haystack Dimension, Precisely

- After playing x and observing y, an algorithm can effectively *eliminate* the functions in $G(x) \cup G(\langle x, y \rangle)$.
- Let $y^*_G(x)$ be the worst-case y.
  - $y^*_G(x) \triangleq \arg\inf_y |G(x) \cup G(\langle x, y \rangle)|$

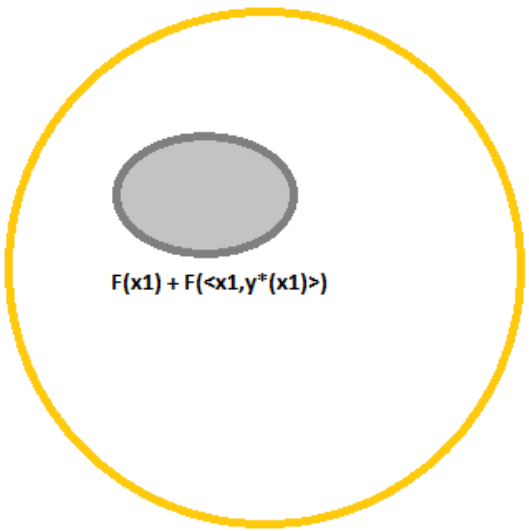


The Haystack Dimension, Precisely

- $\rho(G,x) \triangleq \frac{|G(x) \cup G(\langle x, y^*(x) \rangle)|}{|G|}$ is the fraction of G guaranteed to be eliminated by x (i.e. for worst-case y).
- $\rho(G) \triangleq \sup_{x \in \mathcal{X}} \rho(G,x)$ is the largest fraction of G that can be eliminated in this sense.

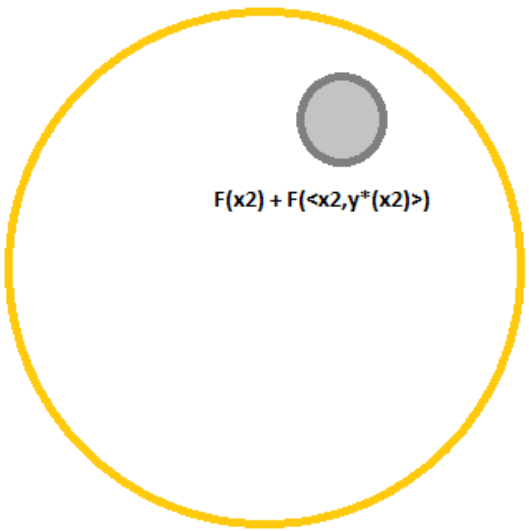
The Haystack Dimension, Precisely

- $\rho(G,x) \triangleq \frac{|G(x) \cup G(\langle x, y^*(x) \rangle)|}{|G|}$ is the fraction of G guaranteed to be eliminated by x (i.e. for worst-case y).
- $\rho(G) \triangleq \sup_{x \in \mathcal{X}} \rho(G,x)$ is the largest fraction of G that can be eliminated in this sense.



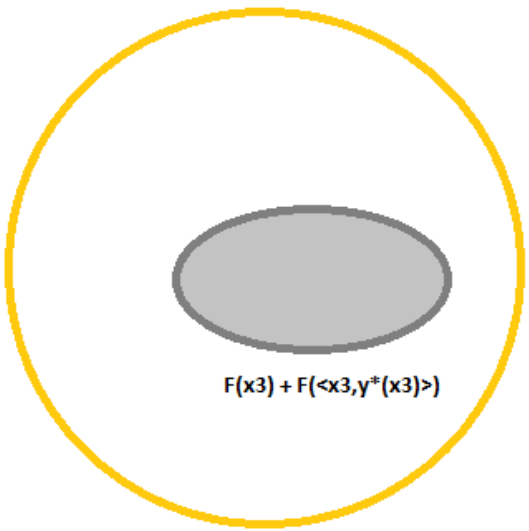
The Haystack Dimension, Precisely

- $\rho(G,x) \triangleq \frac{|G(x) \cup G(\langle x, y^*(x) \rangle)|}{|G|}$ is the fraction of G guaranteed to be eliminated by x (i.e. for worst-case y).
- $\rho(G) \triangleq \sup_{x \in \mathcal{X}} \rho(G,x)$ is the largest fraction of G that can be eliminated in this sense.



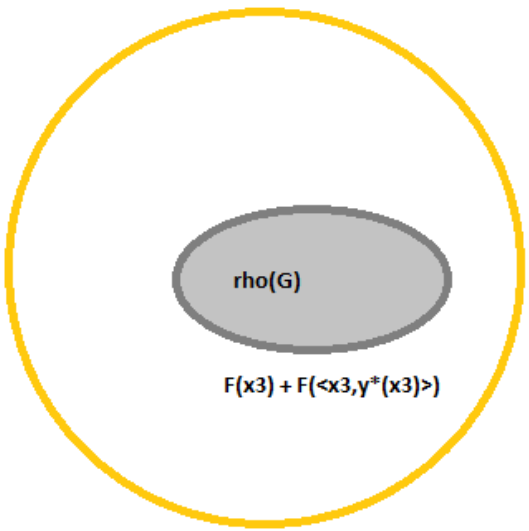
The Haystack Dimension, Precisely

- $\rho(G,x) \triangleq \frac{|G(x) \cup G(\langle x, y^*(x) \rangle)|}{|G|}$ is the fraction of G guaranteed to be eliminated by x (i.e. for worst-case y).
- $\rho(G) \triangleq \sup_{x \in \mathcal{X}} \rho(G,x)$ is the largest fraction of G that can be eliminated in this sense.



The Haystack Dimension, Precisely

- $\rho(G,x) \triangleq \frac{|G(x) \cup G(\langle x, y^*(x) \rangle)|}{|G|}$ is the fraction of G guaranteed to be eliminated by x (i.e. for worst-case y).
- $\rho(G) \triangleq \sup_{x \in \mathcal{X}} \rho(G,x)$ is the largest fraction of G that can be eliminated in this sense.



The Haystack Dimension, Precisely

- Suppose $\rho(G)$ is small for some $G \subseteq \mathcal{F}$, and f^* selected uniformly from G.
- Small probability that x is likely to belong to X^{f^*}.
- Hard to differentiate f^* from rest of G.
- $\theta = \inf_{G \in \mathcal{F}} \rho(G)$.
- Haystack Dimension: $HD(\mathcal{F}) = 1/\theta_{\mathcal{F}}$.

The Haystack Dimension, Precisely

- Suppose $\rho(G)$ is small for some $G \subseteq \mathcal{F}$, and f^* selected uniformly from G.
- Small probability that x is likely to belong to X^{f^*}.
- Hard to differentiate f^* from rest of G.
- $\theta = \inf_{G \in \mathcal{F}} \rho(G)$.
- Haystack Dimension: $HD(\mathcal{F}) = 1/\theta_{\mathcal{F}}$.

The Haystack Dimension, Precisely

- Suppose $\rho(G)$ is small for some $G \subseteq \mathcal{F}$, and f^* selected uniformly from G.
- Small probability that x is likely to belong to X^{f^*}.
- Hard to differentiate f^* from rest of G.
- $\theta = \inf_{G \in \mathcal{F}} \rho(G)$.
- Haystack Dimension: $HD(\mathcal{F}) = 1/\theta_{\mathcal{F}}$.

The Haystack Dimension, Precisely

- Suppose $\rho(G)$ is small for some $G \subseteq \mathcal{F}$, and f^* selected uniformly from G.
- Small probability that x is likely to belong to X^{f^*}.
- Hard to differentiate f^* from rest of G.
- $\theta = \inf_{G \in \mathcal{F}} \rho(G)$.
- Haystack Dimension: $HD(\mathcal{F}) = 1/\theta_{\mathcal{F}}$.

The Haystack Dimension, Precisely

- Suppose $\rho(G)$ is small for some $G \subseteq \mathcal{F}$, and f^* selected uniformly from G.
- Small probability that x is likely to belong to X^{f^*}.
- Hard to differentiate f^* from rest of G.
- $\theta = \inf_{G \in \mathcal{F}} \rho(G)$.
- Haystack Dimension: $HD(\mathcal{F}) = 1/\theta_{\mathcal{F}}$.

Lower Bound for MAX Problem

- In the worst case, any randomized algorithm needs $\Omega(HD(\mathcal{F}))$ rounds to solve the MAX problem on $\mathcal{F}$.
- *Proof Sketch*:
- Let G_θ be the subset of $\mathcal{F}$ which realizes the Haystack Dimension. **Recall that** $\theta = 1/HD(\mathcal{F})$.
- Consider f^* chosen uniformly at random from G_θ.
- Let $x_1, x_2, x_3, ...$ be actions played by algorithm A.

Lower Bound for MAX Problem

- In the worst case, any randomized algorithm needs $\Omega(HD(\mathcal{F}))$ rounds to solve the MAX problem on $\mathcal{F}$.
- *Proof Sketch*:
- Let G_θ be the subset of $\mathcal{F}$ which realizes the Haystack Dimension. **Recall that** $\theta = 1/HD(\mathcal{F})$.
- Consider f^* chosen uniformly at random from G_θ.
- Let $x_1, x_2, x_3, ...$ be actions played by algorithm A.

Lower Bound for MAX Problem

- In the worst case, any randomized algorithm needs $\Omega(HD(\mathcal{F}))$ rounds to solve the MAX problem on $\mathcal{F}$.
- *Proof Sketch*:
- Let G_θ be the subset of $\mathcal{F}$ which realizes the Haystack Dimension. **Recall that** $\theta = 1/HD(\mathcal{F})$.
- Consider f^* chosen uniformly at random from G_θ.
- Let $x_1, x_2, x_3, \ldots$ be actions played by algorithm A.

Lower Bound for MAX Problem

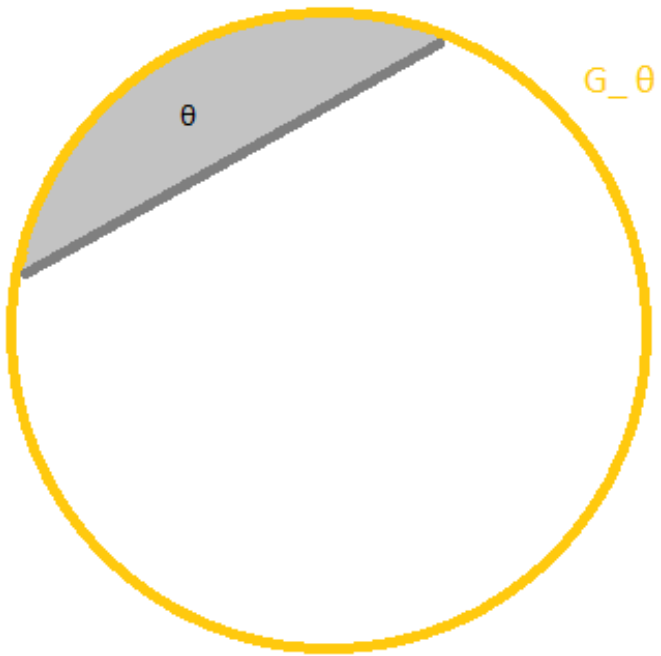
- In the worst case, any randomized algorithm needs $\Omega(HD(\mathcal{F}))$ rounds to solve the MAX problem on $\mathcal{F}$.
- *Proof Sketch*:
- Let G_θ be the subset of $\mathcal{F}$ which realizes the Haystack Dimension. **Recall that** $\theta = 1/HD(\mathcal{F})$.
- Consider f^* chosen uniformly at random from G_θ.
- Let $x_1, x_2, x_3, \ldots$ be actions played by algorithm A.

Lower Bound for MAX Problem

- In the worst case, any randomized algorithm needs $\Omega(HD(\mathcal{F}))$ rounds to solve the MAX problem on $\mathcal{F}$.
- *Proof Sketch*:
- Let G_θ be the subset of $\mathcal{F}$ which realizes the Haystack Dimension. **Recall that** $\theta = 1/HD(\mathcal{F})$.
- Consider f^* chosen uniformly at random from G_θ.
- Let $x_1, x_2, x_3, \ldots$ be actions played by algorithm A.

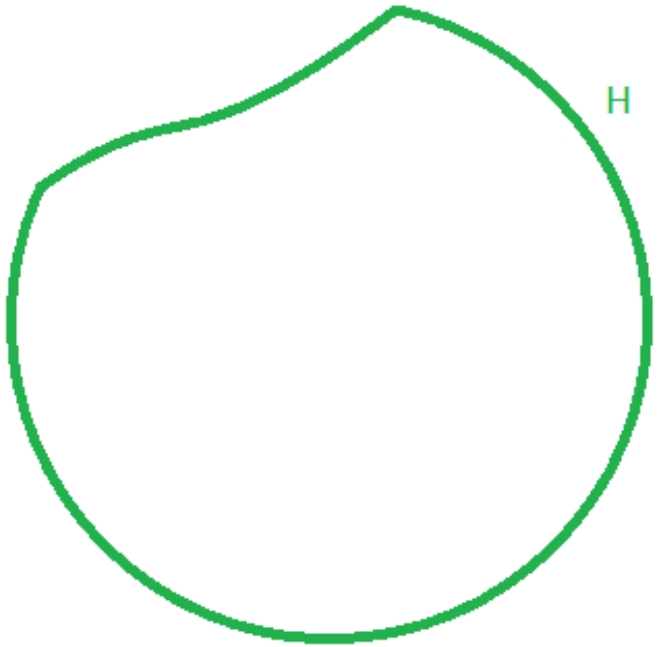
Lower Bound Proof Sketch

By definition of G_θ, the probability that x_1 is a maximizing action is no more than θ, the largest fraction that can be *eliminated* from F_θ.



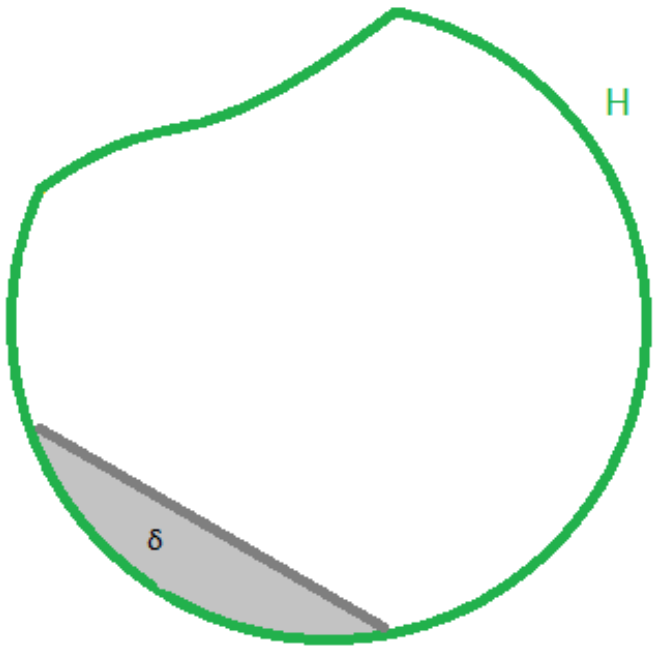
Lower Bound Proof Sketch

Claim: On round 2, we can think of f^* as being drawn uniformly from $H = G_\theta \setminus (G_\theta(x_1) \cup G_\theta(\langle x_1, y^*(x_1)))$.



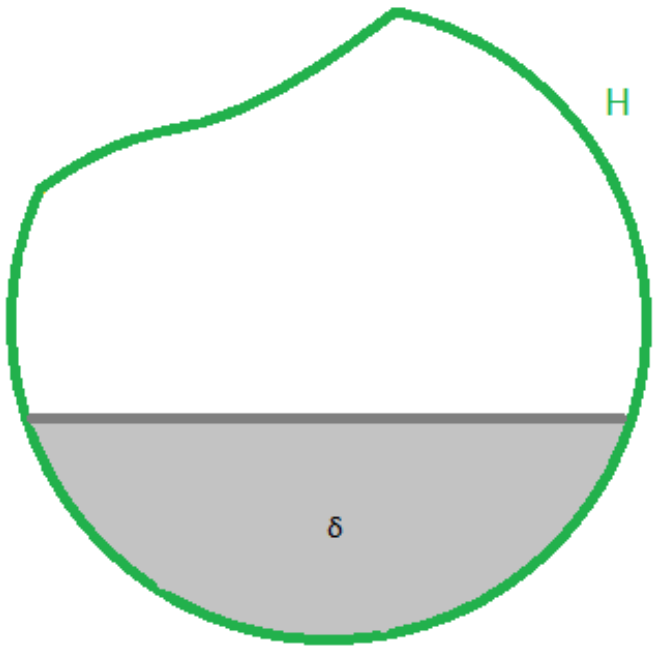
Lower Bound Proof Sketch

Let δ be the fraction of functions that can be eliminated from H.



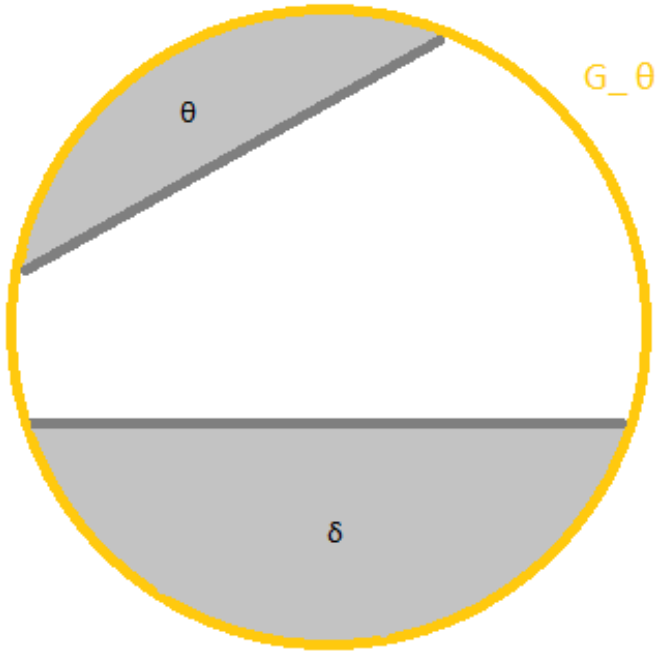
Lower Bound Proof Sketch

Can have $\delta > \theta$. Would like $\delta \approx \theta$ to repeat the argument of round 1.



Lower Bound Proof Sketch

However, δ cannot be too much bigger than θ or else on round 1 there was an action that eliminated a fraction of G_θ that was larger than θ, a contradiction.



Lower Bound Proof Sketch

- Can show the probability of maximizing a function on round t, δ_t is $O(\theta)$ when $t < \frac{1}{\theta}$.
- Until t is $\Omega(\frac{1}{\theta} = HD(\mathcal{F}))$, constant probability $x_1, ..., x_t \notin X^{f^*}$

Greedy MAX

- Is lower bound tight? Use the ideas from the lower bound.
- Define A_t to be the *attention space* at time t
  - A_t set of consistent function with observations $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$ (version space).
  - With any functions maximized by $x_1, ..., x_{t-1}$ removed.
- On round t, greedily select action that guarantees the greatest reduction in the attention space.
- That is, $x_t = \arg\max_{x \in \mathcal{X}} \rho(A_{t-1}, x)$
- Emptying attention space is proof that maximizing arm was played.
- For finite $\mathcal{F}$, query complexity is $O(HD(\mathcal{F}) \log |\mathcal{F}|)$

Greedy MAX

- Is lower bound tight? Use the ideas from the lower bound.
- Define A_t to be the *attention space* at time t
  - A_t set of consistent function with observations $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$ (version space).
  - With any functions maximized by $x_1, ..., x_{t-1}$ removed.
- On round t, greedily select action that guarantees the greatest reduction in the attention space.
- That is, $x_t = \arg\max_{x \in \mathcal{X}} \rho(A_{t-1}, x)$
- Emptying attention space is proof that maximizing arm was played.
- For finite $\mathcal{F}$, query complexity is $O(HD(\mathcal{F}) \log |\mathcal{F}|)$

Greedy MAX

- Is lower bound tight? Use the ideas from the lower bound.
- Define A_t to be the *attention space* at time t
  - A_t set of consistent function with observations $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$ (version space).
    - With any functions maximized by $x_1, ..., x_{t-1}$ removed.
- On round t, greedily select action that guarantees the greatest reduction in the attention space.
- That is, $x_t = \arg\max_{x \in \mathcal{X}} \rho(A_{t-1}, x)$
- Emptying attention space is proof that maximizing arm was played.
- For finite $\mathcal{F}$, query complexity is $O(HD(\mathcal{F}) \log |\mathcal{F}|)$

Greedy MAX

- Is lower bound tight? Use the ideas from the lower bound.
- Define A_t to be the *attention space* at time t
  - A_t set of consistent function with observations $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$ (version space).
  - With any functions maximized by $x_1, ..., x_{t-1}$ removed.
- On round t, greedily select action that guarantees the greatest reduction in the attention space.
- That is, $x_t = \arg\max_{x \in \mathcal{X}} \rho(A_{t-1}, x)$
- Emptying attention space is proof that maximizing arm was played.
- For finite $\mathcal{F}$, query complexity is $O(HD(\mathcal{F}) \log |\mathcal{F}|)$

Greedy MAX

- Is lower bound tight? Use the ideas from the lower bound.
- Define A_t to be the *attention space* at time t
  - A_t set of consistent function with observations $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$ (version space).
  - With any functions maximized by $x_1, ..., x_{t-1}$ removed.
- On round t, greedily select action that guarantees the greatest reduction in the attention space.
- That is, $x_t = \arg\max_{x \in \mathcal{X}} \rho(A_{t-1}, x)$
- Emptying attention space is proof that maximizing arm was played.
- For finite $\mathcal{F}$, query complexity is $O(HD(\mathcal{F}) \log |\mathcal{F}|)$

Greedy MAX

- Is lower bound tight? Use the ideas from the lower bound.
- Define A_t to be the *attention space* at time t
  - A_t set of consistent function with observations $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$ (version space).
  - With any functions maximized by $x_1, ..., x_{t-1}$ removed.
- On round t, greedily select action that guarantees the greatest reduction in the attention space.
- That is, $x_t = \arg\max_{x \in \mathcal{X}} \rho(A_{t-1}, x)$
- Emptying attention space is proof that maximizing arm was played.
- For finite $\mathcal{F}$, query complexity is $O(HD(\mathcal{F}) \log |\mathcal{F}|)$

Greedy MAX

- Is lower bound tight? Use the ideas from the lower bound.
- Define A_t to be the *attention space* at time t
  - A_t set of consistent function with observations $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$ (version space).
  - With any functions maximized by $x_1, ..., x_{t-1}$ removed.
- On round t, greedily select action that guarantees the greatest reduction in the attention space.
- That is, $x_t = \arg\max_{x \in \mathcal{X}} \rho(A_{t-1}, x)$
- Emptying attention space is proof that maximizing arm was played.
- For finite $\mathcal{F}$, query complexity is $O(HD(\mathcal{F}) \log |\mathcal{F}|)$

Greedy MAX

- Is lower bound tight? Use the ideas from the lower bound.
- Define A_t to be the *attention space* at time t
  - A_t set of consistent function with observations $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$ (version space).
  - With any functions maximized by $x_1, ..., x_{t-1}$ removed.
- On round t, greedily select action that guarantees the greatest reduction in the attention space.
- That is, $x_t = \arg\max_{x \in \mathcal{X}} \rho(A_{t-1}, x)$
- Emptying attention space is proof that maximizing arm was played.
- For finite $\mathcal{F}$, query complexity is $O(HD(\mathcal{F}) \log |\mathcal{F}|)$

Returning to MAB Setting

- MAB Setting
- For any finite $\mathcal{F}$ and MAB algorithm A, $R_A(T) = \Omega(\Delta \min\{T, HD(\mathcal{F})\})$
- Follows directly from MAX setting.
- Can give Greedy MAB algorithm that uses Greedy MAX algorithm as subroutine.
- Regret Bound: $\tilde{O}(\frac{HD(\mathcal{F})}{\epsilon^2} \log T)$
- $\epsilon = \min_{f,f' \in \mathcal{F}} \inf_{x:f(x) \neq f'(x)} |f(x) - f'(x)|$
- Conjecture gap can be removed.

Returning to MAB Setting

- MAB Setting
- For any finite $\mathcal{F}$ and MAB algorithm A, $R_A(T) = \Omega(\Delta \min\{T, HD(\mathcal{F})\})$
- Follows directly from MAX setting.
- Can give Greedy MAB algorithm that uses Greedy MAX algorithm as subroutine.
- Regret Bound: $\tilde{O}(\frac{HD(\mathcal{F})}{\epsilon^2} \log T)$
- $\epsilon = \min_{f,f' \in \mathcal{F}} \inf_{x:f(x) \neq f'(x)} |f(x) - f'(x)|$
- Conjecture gap can be removed.

Returning to MAB Setting

- MAB Setting
- For any finite $\mathcal{F}$ and MAB algorithm A, $R_A(T) = \Omega(\Delta \min\{T, HD(\mathcal{F})\})$
- Follows directly from MAX setting.
- Can give Greedy MAB algorithm that uses Greedy MAX algorithm as subroutine.
- Regret Bound: $\tilde{O}(\frac{HD(\mathcal{F})}{\epsilon^2} \log T)$
- $\epsilon = \min_{f,f' \in \mathcal{F}} \inf_{x: f(x) \neq f'(x)} |f(x) - f'(x)|$
- Conjecture gap can be removed.

Returning to MAB Setting

- MAB Setting
- For any finite $\mathcal{F}$ and MAB algorithm A, $R_A(T) = \Omega(\Delta \min\{T, HD(\mathcal{F})\})$
- Follows directly from MAX setting.
- Can give Greedy MAB algorithm that uses Greedy MAX algorithm as subroutine.
- Regret Bound: $\tilde{O}(\frac{HD(\mathcal{F})}{\epsilon^2} \log T)$
- $\epsilon = \min_{f,f' \in \mathcal{F}} \inf_{x: f(x) \neq f'(x)} |f(x) - f'(x)|$
- Conjecture gap can be removed.

Returning to MAB Setting

- MAB Setting
- For any finite $\mathcal{F}$ and MAB algorithm A, $R_A(T) = \Omega(\Delta \min\{T, HD(\mathcal{F})\})$
- Follows directly from MAX setting.
- Can give Greedy MAB algorithm that uses Greedy MAX algorithm as subroutine.
- Regret Bound: $\tilde{O}(\frac{HD(\mathcal{F})}{\epsilon^2} \log T)$
- $\epsilon = \min_{f,f' \in \mathcal{F}} \inf_{x: f(x) \neq f'(x)} |f(x) - f'(x)|$
- Conjecture gap can be removed.

Returning to MAB Setting

- MAB Setting
- For any finite $\mathcal{F}$ and MAB algorithm A, $R_A(T) = \Omega(\Delta \min\{T, HD(\mathcal{F})\})$
- Follows directly from MAX setting.
- Can give Greedy MAB algorithm that uses Greedy MAX algorithm as subroutine.
- Regret Bound: $\tilde{O}(\frac{HD(\mathcal{F})}{\epsilon^2} \log T)$
- $\epsilon = \min_{f,f' \in \mathcal{F}} \inf_{x: f(x) \neq f'(x)} |f(x) - f'(x)|$
- Conjecture gap can be removed.

Returning to MAB Setting

- MAB Setting
- For any finite $\mathcal{F}$ and MAB algorithm A, $R_A(T) = \Omega(\Delta \min\{T, HD(\mathcal{F})\})$
- Follows directly from MAX setting.
- Can give Greedy MAB algorithm that uses Greedy MAX algorithm as subroutine.
- Regret Bound: $\tilde{O}(\frac{HD(\mathcal{F})}{\epsilon^2} \log T)$
- $\epsilon = \min_{f,f' \in \mathcal{F}} \inf_{x: f(x) \neq f'(x)} |f(x) - f'(x)|$
- Conjecture gap can be removed.

Infinite F

- Everything so far has been for finite $\mathcal{F}$.
- For infinite $\mathcal{F}$, we have some results.
- $\mathcal{F}$ can be uniformly well-approximated by finite cover.
- However, introduces an additional gap in results.
  - For certain function classes this gap can be controlled.
  - e.g. Linear functions: gap is the dimensionality.
  - Certain function classes, bounds are arbitrarily loose.

Infinite F

- Everything so far has been for finite $\mathcal{F}$.
- For infinite $\mathcal{F}$, we have some results.
- $\mathcal{F}$ can be uniformly well-approximated by finite cover.
- However, introduces an additional gap in results.
  - For certain function classes this gap can be controlled.
  - e.g. Linear functions: gap is the dimensionality.
  - Certain function classes, bounds are arbitrarily loose.

Infinite F

- Everything so far has been for finite $\mathcal{F}$.
- For infinite $\mathcal{F}$, we have some results.
- $\mathcal{F}$ can be uniformly well-approximated by finite cover.
- However, introduces an additional gap in results.
  - For certain function classes this gap can be controlled.
  - e.g. Linear functions: gap is the dimensionality.
  - Certain function classes, bounds are arbitrarily loose.

Infinite F

- Everything so far has been for finite $\mathcal{F}$.
- For infinite $\mathcal{F}$, we have some results.
- $\mathcal{F}$ can be uniformly well-approximated by finite cover.
- However, introduces an additional gap in results.
  - For certain function classes this gap can be controlled.
  - e.g. Linear functions: gap is the dimensionality.
  - Certain function classes, bounds are arbitrarily loose.

Infinite F

- Everything so far has been for finite $\mathcal{F}$.
- For infinite $\mathcal{F}$, we have some results.
- $\mathcal{F}$ can be uniformly well-approximated by finite cover.
- However, introduces an additional gap in results.
  - For certain function classes this gap can be controlled.
  - e.g. Linear functions: gap is the dimensionality.
  - Certain function classes, bounds are arbitrarily loose.

Infinite F

- Everything so far has been for finite $\mathcal{F}$.
- For infinite $\mathcal{F}$, we have some results.
- $\mathcal{F}$ can be uniformly well-approximated by finite cover.
- However, introduces an additional gap in results.
  - For certain function classes this gap can be controlled.
  - e.g. Linear functions: gap is the dimensionality.
  - Certain function classes, bounds are arbitrarily loose.

Infinite F

- Everything so far has been for finite $\mathcal{F}$.
- For infinite $\mathcal{F}$, we have some results.
- $\mathcal{F}$ can be uniformly well-approximated by finite cover.
- However, introduces an additional gap in results.
  - For certain function classes this gap can be controlled.
  - e.g. Linear functions: gap is the dimensionality.
  - Certain function classes, bounds are arbitrarily loose.

Conclusions

- Analogous to VC-Dimension but incomparable.
  - $\mathcal{F}$ can have small VC-Dimension, Large Haystack Dimension and vice-versa.
- Generalizes Extended Teaching Dimension.

Conclusions

- Analogous to VC-Dimension but incomparable.
  - $\mathcal{F}$ can have small VC-Dimension, Large Haystack Dimension and vice-versa.
- Generalizes Extended Teaching Dimension.

Thanks!

Thanks!