# Zooming Algorithm for Lipschitz Bandits

Alex Slivkins
Microsoft Research New York City

Based on joint work
with Robert Kleinberg and Eli Upfal (STOC'08)

# Running examples

- **Dynamic pricing.**
  You release a song which customers can download for a price. What price will maximize profit?
  - Customers arrive one by one, you can update the price
- **Web advertisement.**
  Every time someone visits your site, you display an ad. There are many ads to choose from. Which one will maximize #clicks?
  - you can update your selection based on the clicks received

# Multi-Armed Bandits

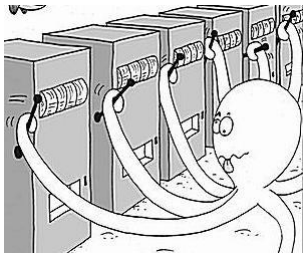
| | *arms* | *payoffs* |
|---|---|---|
| *pricing* | prices | payments |
| *web ads* | ads | clicks |

- In a (basic) MAB problem one has:
  - set $X$ of strategies (a.k.a. *arms*)
  - $\mu(x) \in [0,1]$ expected payoff for each $x \in X$ (fixed but unknown)
- In each round an algorithm
  - picks arm $x \in X$ based on past history
  - receives payoffs (money): an independent sample in $[0,1]$ from distribution $D(x)$ with expectation $\mu(x)$


$\mu=.6$


$\mu=.2$


$\mu=.4$



# Exploration vs Exploitation

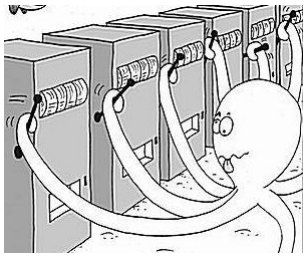
- Explore: try out new arms to get more info
  … perhaps playing low-paying arms
- Exploit: play arms that seem best based on current info
  … but maybe there is a better arm that we don't know about

- Classical setting since 1952
  - OR, Econ, CS: various versions and extensions


$\mu$=.6


$\mu$=.2


$\mu$=.4



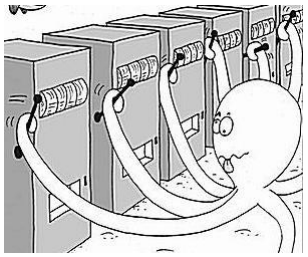
# Background

- Early work: maximize expected time-discounted payoffs w.r.t. independent bayesian priors over arms.
  Solved by the "Gittins index policy" ( *Gittins and Jones (1972)* )
- We focus on the **prior-free** version
  - arm $x \Rightarrow$ i.i.d. sample with expectation $\mu(x)$
  - benchmark: $\mu^* = \max_{x \in X} \mu(x)$
    *Regret* in $T$ rounds: $R(T) = T\,\mu^* -$ [expected total payoffs]


$\mu$=.6


$\mu$=.2


$\mu$=.4



# Background

- For small #arms (K), the problem is well-understood

( Lai & Robbins (1985), Auer at al. (2002) )

Benchmark: $\mu^* = \max_{x \in X} \mu(x)$

*Regret*: $R(T) = T\,\mu^* - [\text{expected total payoffs}]$

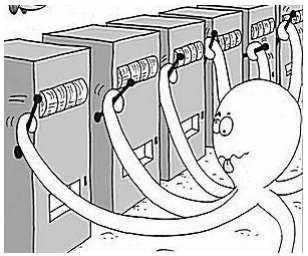
- $R(T) \le O_\mu(K \log T)$ for fixed $\mu$

  $R(T) \le O(K T \log K)^{1/2}$ in the worst case
- both optimal via relative entropy arguments

$\mu$=.6

$\mu$=.2

$\mu$=.4



# Bandits with side information

- What if the strategy set is very large? infinite?
  - needle in a haystack $\Rightarrow$ hopeless **unless** we have **side info**
- Dynamic pricing

  unlimited supply of identical digital goods, seller can update the price; arms are prices
  - numerical similarity between arms
  - known *shape* of payoff function, e.g. smoothness
- Web advertisement

  new user arrives, display one of the k ads, maximize #clicks; arms are ads
  - similarity between arms: topical taxonomy, feature vectors, etc
  - context: user profile, page features

Present scope: **similarity between arms**

# Lipschitz MAB problem

- Algorithm is given **similarity metric L** on arms
  such that

$$|\mu(x) - \mu(y)| \leq L(x, y) \quad \forall x,y \in X \qquad \textit{(Lipschitz condition)}$$

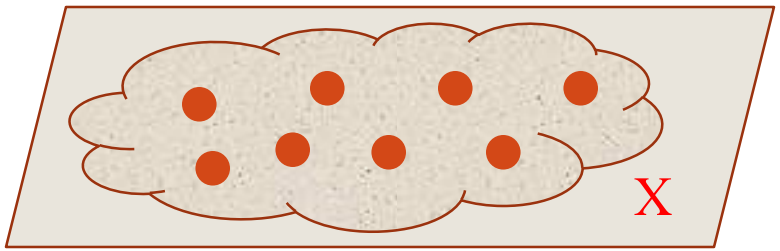
  In other words, considering *payoff function* $\mu$: $X \to \mu(x)$:

  $\mu$ is Lipschitz-continuous w.r.t. $(X,L)$

- Problem instance: (known) metric space $(X,L)$ and (unknown) $\mu$
- **How to utilize this side information?**
  **What performance guarantees (regret) can be achieved?**

# A (very) naive algorithm

- in each phase, choose K equally spaced arms (ε-net),
  use an off-the-shelf K-armed bandit algorithm
  *one of the chosen arms is close to the opt!*
- phase i lasts for $2^i$ rounds; $K = 2^{i\,d/(d+2)}$, d = CoveringDim



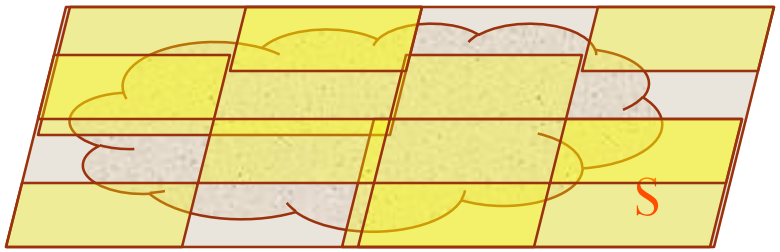
# A (very) naive algorithm

- in each phase, choose K equally spaced arms (ε-net),
  use an off-the-shelf K-armed bandit algorithm
  *one of the chosen arms is close to the opt!*
- phase i lasts for $2^i$ rounds; $K = 2^{i\,d/(d+2)}$, d = CoveringDim

Definition Covering Dimension of a metric space

- $\forall r>0$ the metric can be covered with $c\,r^{-d}$ sets of diameter $\leq r$
- c-*CovDim* = smallest such d

**Fact: CovDim ≤ DoublingDim ≤ EuclideanDim**

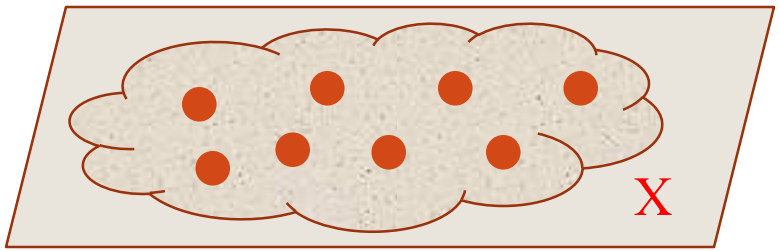


# A (very) naive algorithm

- in each phase, choose K equally spaced arms (ε-net),
  use an off-the-shelf K-armed bandit algorithm
  *one of the chosen arms is close to the opt!*
- phase i lasts for $2^i$ rounds; $K = 2^{i\, d/(d+2)}$, d = CoveringDim

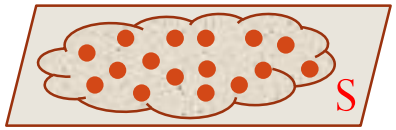
Theorem: using off-the-shelf guarantees

$$R(T) \leq O(T^{1-1/(d+2)} \log T)$$



# Is this the right algorithm ??

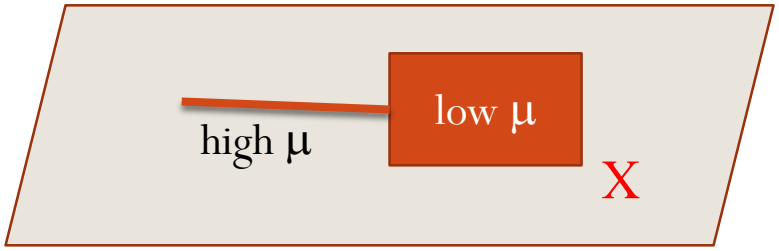
The naive algorithm seems wasteful:

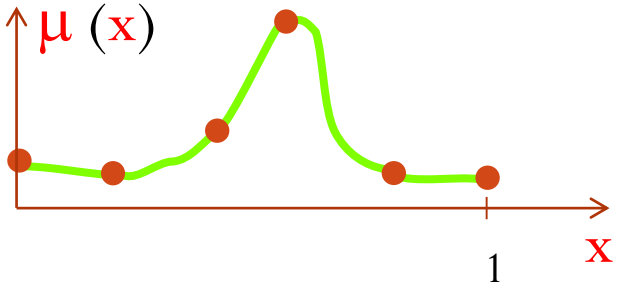



- places equally spaced probes
  (what if some regions yield better payoffs than others?)
- after the probes are placed, all similarity information is discarded

For a given metric space, **can we do better?**

- … in the worst case? **YES**
- … for a ***nice*** problem instance (payoff function)? **YES** This talk



# Better algorithm for nice instances

- **Goal**: do as well as the naive algorithm in general, but perform "better" on "nice" problem instances

# Our results: zooming algorithm

Theorem The zooming algorithm achieves regret

$$R(T) \le O(c\ T^{1-1/(d+2)} \log T)$$

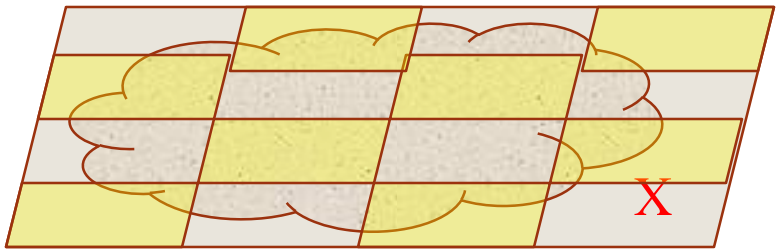
where $d$ = ~~c-*CovDim* of similarity metric **L**~~

c-Zooming Dimension of problem instance ($\mu$, **L**)

Definition ~~Covering Dimension of a metric space~~

- $\forall r>0$ the metric can be covered with $c\ r^{-d}$ sets of diameter $\le r$
- ~~c-*CovDim*~~ = smallest such $d$

c-*ZoomingDim*



# Our results: zooming algorithm

**Theorem** The zooming algorithm achieves regret

$$R(T) \le O(c\ T^{1-1/(d+2)} \log T)$$

where $d$ = ~~c-*CovDim* of similarity metric **L**~~

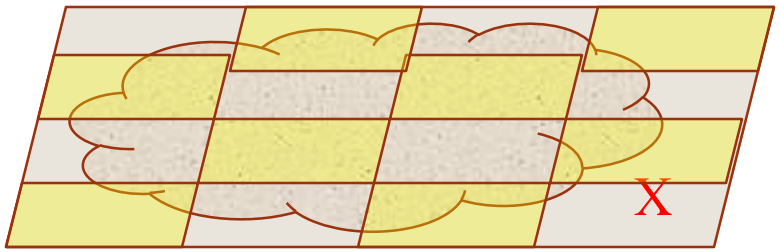
c-Zooming Dimension of problem instance $(\mu, L)$

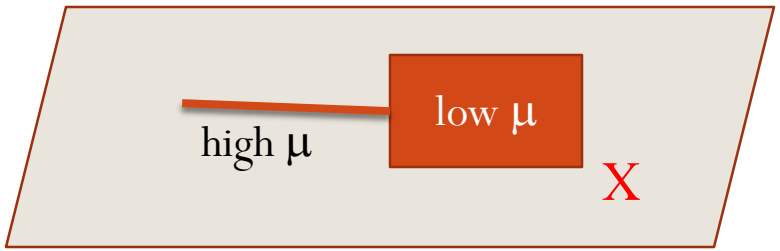
**Definition** ~~Covering Dimension of a metric space~~

$\{x: \ r/2 \le \mu^* - \mu(x) \le r\}$

- $\forall r>0$ ~~the metric~~ can be covered with $c\ r^{-d}$ sets of diameter $\le r$
- ~~c-*CovDim*~~ = smallest such $d$

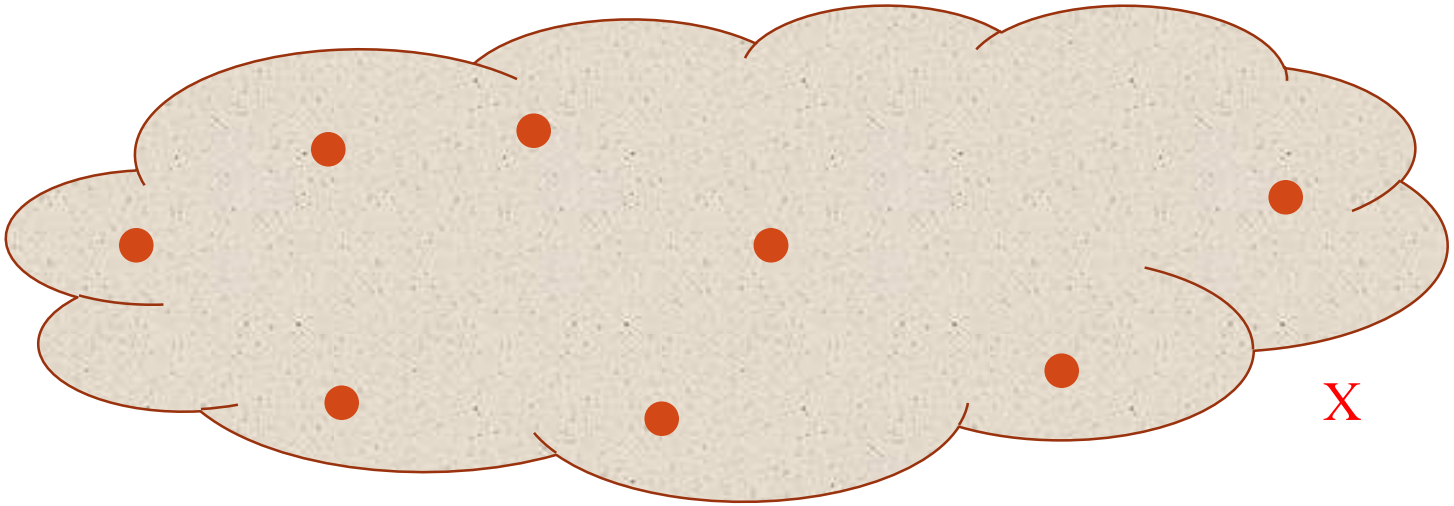
c-*ZoomingDim*



# Zooming algorithm

- maintain a finite set of *active arms*
  - start with no active arms, *activate* one by one.
    in each round, play one of the active arms.
- ACTIVATION RULE: add a new active arm? which one?
  SELECTION RULE: choose which active arm to play next



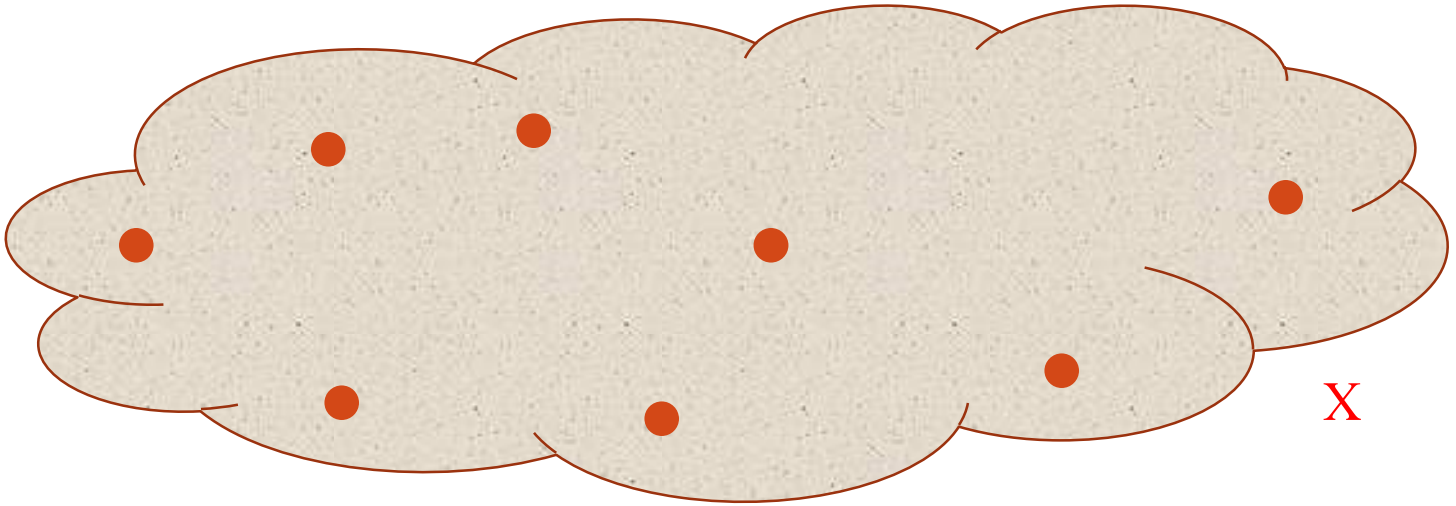
# Activation rule

- $r_t(x)$ = confidence radius of arm $x$ at time $t$

  $|\ \text{SAMPLEAVERAGE}_t(x) - \mu(x)\ | \leq r_t(x)$ **w.h.p.**

- by Chernoff Bounds

$$r_t(x) \leq \sqrt{\frac{8 \log t}{\#\text{ samples from } x}}$$

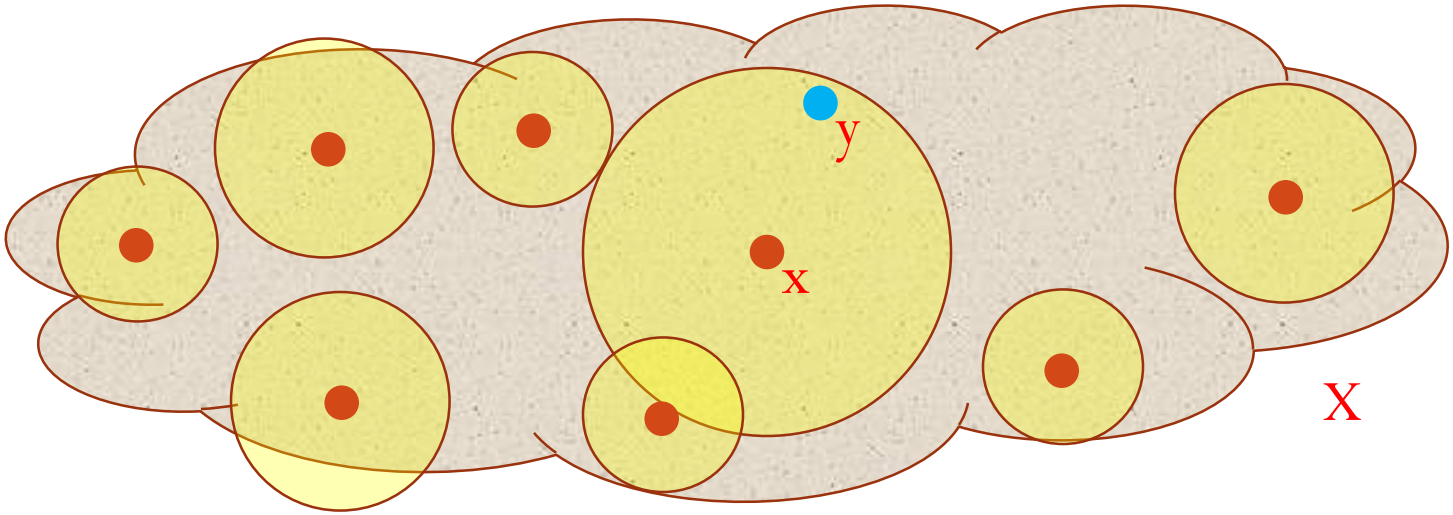


# Activation rule

- $r_t(x)$ = confidence radius of arm $x$ at time $t$

  $|\ \text{SAMPLEAVERAGE}_t(x) - \mu(x)\ | \le r_t(x)$ **w.h.p.**

- *confidence ball* $\mathbf{B}_t(x) = \mathbf{B}(x, r_t(x))$
- **intuition**: should we activate $y$?

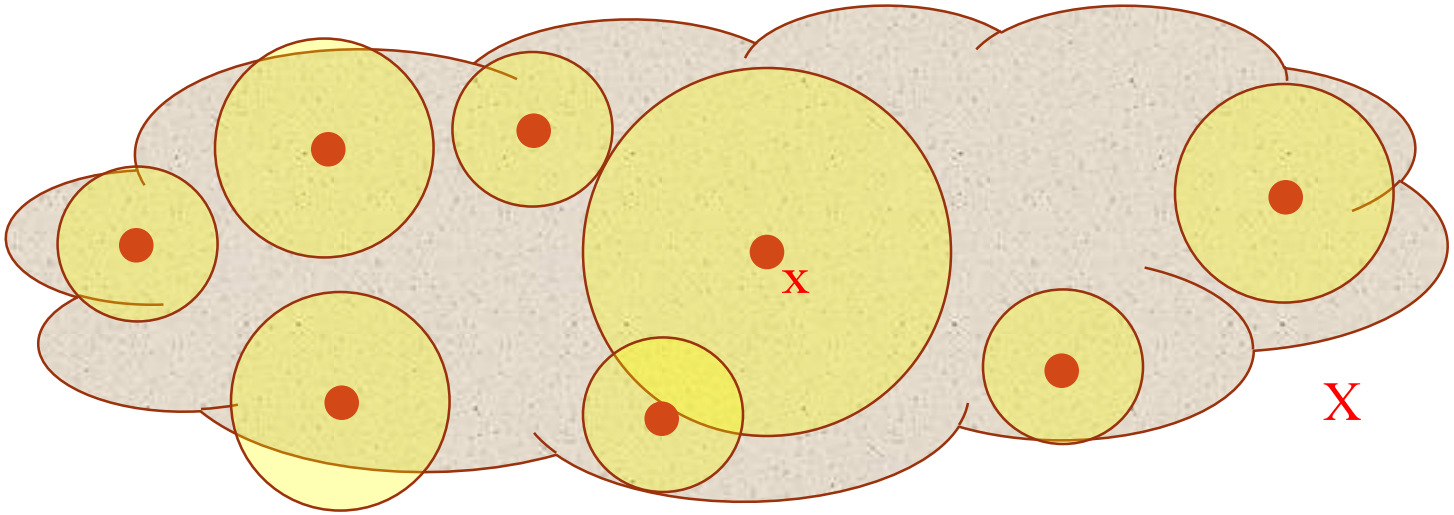


# Activation rule

- $r_t(x)$ = confidence radius of arm $x$ at time $t$

  $|\ \text{SAMPLEAVERAGE}_t(x) - \mu(x)\ | \le r_t(x)$ **w.h.p.**

- *confidence ball* $\mathbf{B}_t(x) = \mathbf{B}(x, r_t(x))$
- **intuition**: no point to activate arm which is ***covered***
- **maintain invariant: all arms are covered**

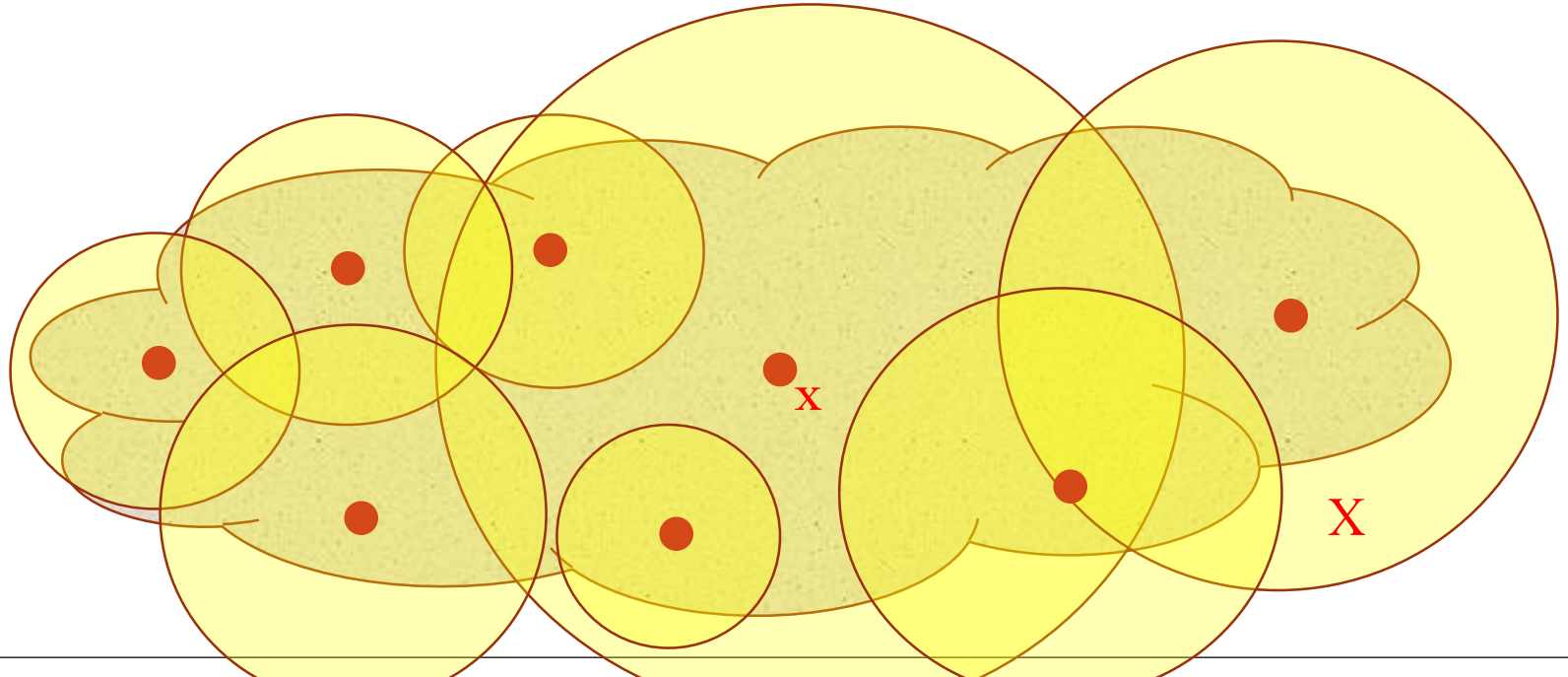


# Activation rule

- $r_t(x)$ = confidence radius of arm $x$ at time $t$

  $|\ \text{SAMPLEAVERAGE}_t(x) - \mu(x)\ | \le r_t(x)$ **w.h.p.**

- *confidence ball* $\mathbf{B}_t(x) = \mathbf{B}(x, r_t(x))$
- **intuition**: no point to activate arm which is ***covered***
- **maintain invariant: all arms are covered**

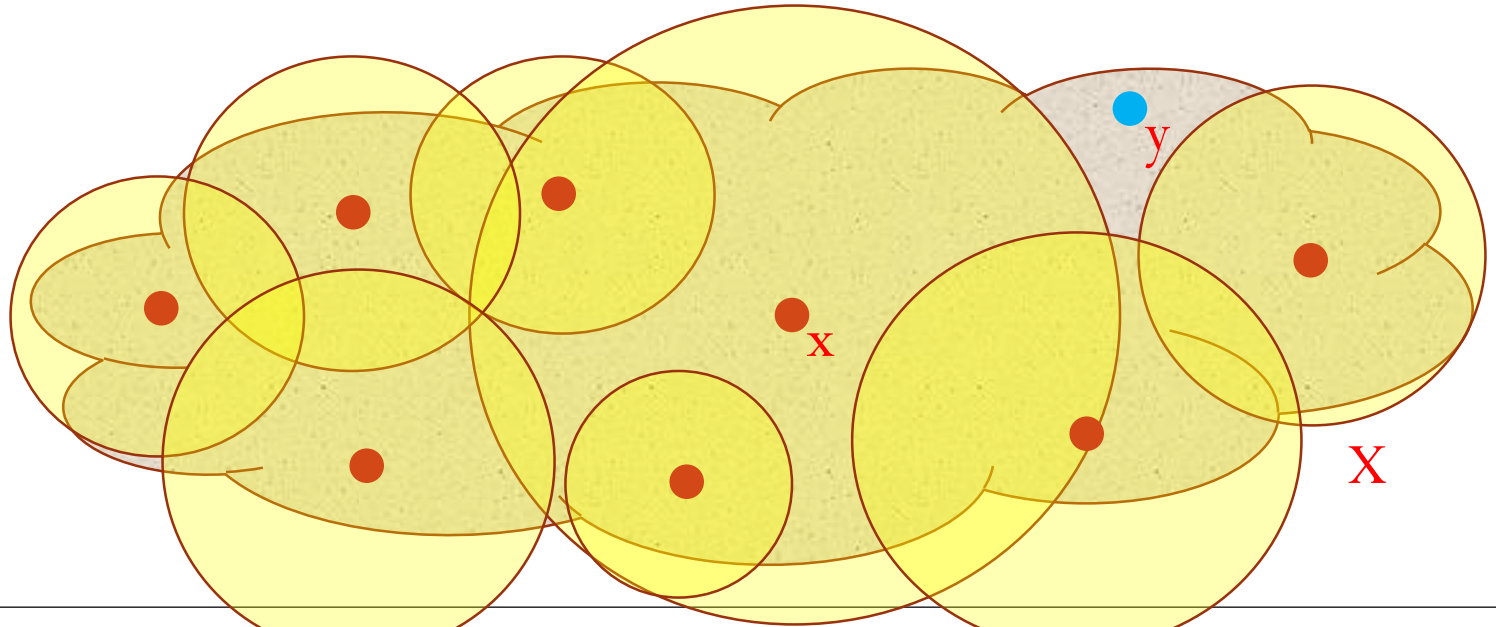


# Activation rule

- $r_t(x)$ = confidence radius of arm $x$ at time $t$

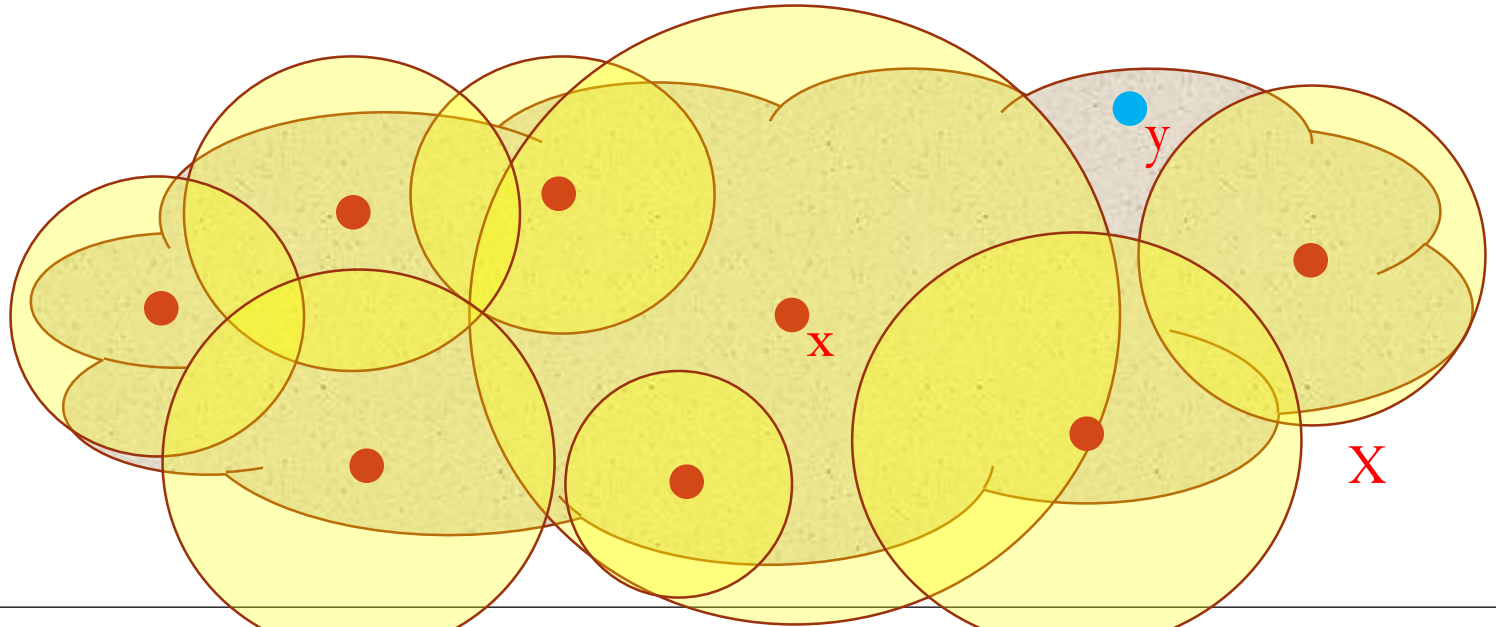
  $|\ \text{SAMPLEAVERAGE}_t(x) - \mu(x)\ | \le r_t(x)$ **w.h.p.**

- *confidence ball* $\mathbf{B}_t(x) = \mathbf{B}(x, r_t(x))$
- **intuition**: no point to activate arm which is ***covered***
- **maintain invariant: all arms are covered**



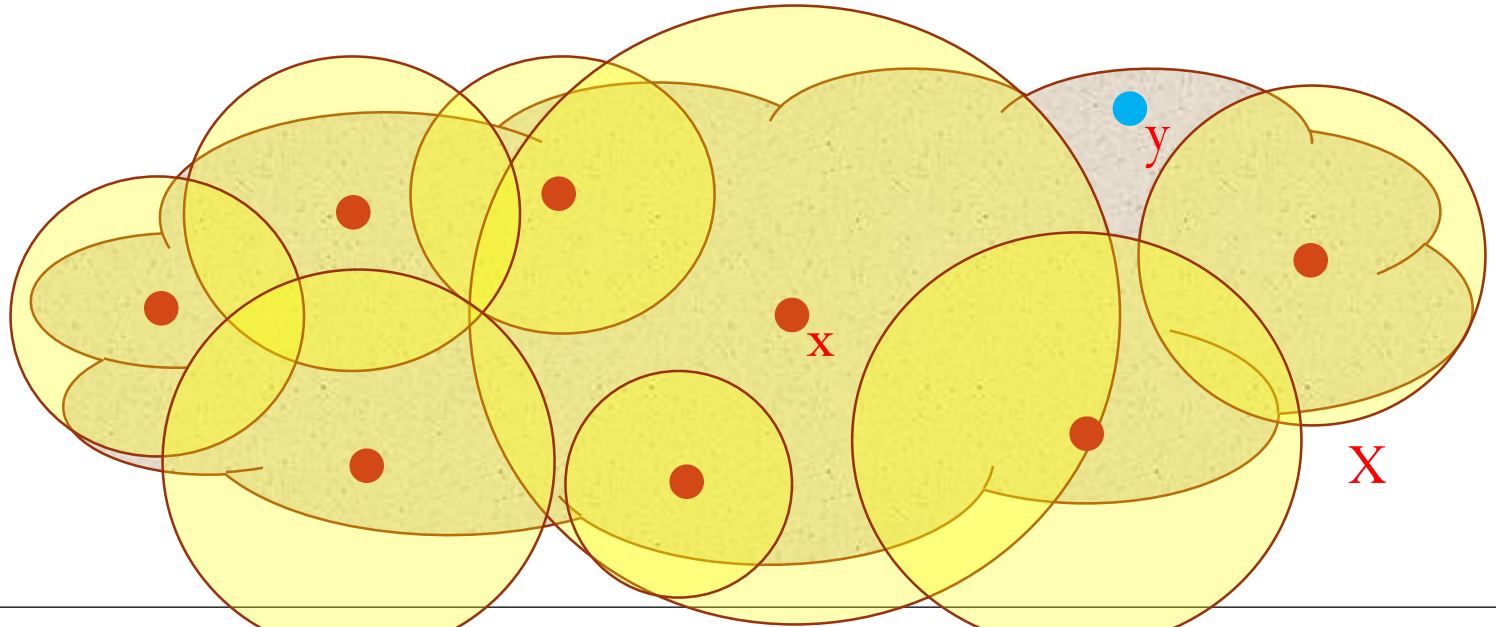
# Activation rule

- **maintain invariant: all arms are covered**
  - what  if some arm becomes uncovered?



# Activation rule

- **maintain invariant: all arms are covered**
- ACTIVATION RULE: if arm y becomes uncovered, **activate it**
  - initially confidence radius $r_t(y)$ is very large, so confidence ball $\mathbf{B}(y, r_t(y))$ covers the entire metric
  - self-adjusting: "zoom in on region R" $\Leftrightarrow$ activate many arms in R $\Leftrightarrow$ arms in R are played often $\Leftrightarrow$ arms in R are good



# Selection rule

- Define $\text{INDEX}_t(x) = \text{SAMPLEAVERAGE}_t(x) + 2\, r_t(x)$
  - Recall: $|\ \text{SAMPLEAVERAGE}_t(x) - \mu(x)\ | \le r_t(x)$ w.h.p.
- SELECTION RULE: play active arm **with max index**
  - why does it make sense? If index is large then:
    *either* sample average is large ($\Rightarrow$ good arm) ,
    *or* confidence radius is large ($\Rightarrow$ need to explore it more)

# Sketch of analysis

- Key fact: if $x$ is played at time $t$ then $\text{INDEX}_t(x) \geq \mu^*$
- "badness" $\Delta(x) \equiv \mu^* - \mu(x)$

Consider active arms $x$ such that $r/2 \leq \Delta(x) \leq r$

- To bound regret, we show that:
  - we don't activate too many "bad" arms:
    sparsity: $L(x, y) \geq \Omega(r)$
  - each "bad" arm is not played too often :
    $\#\text{samples}(x) \leq O(1/r^2)$

# Extensions

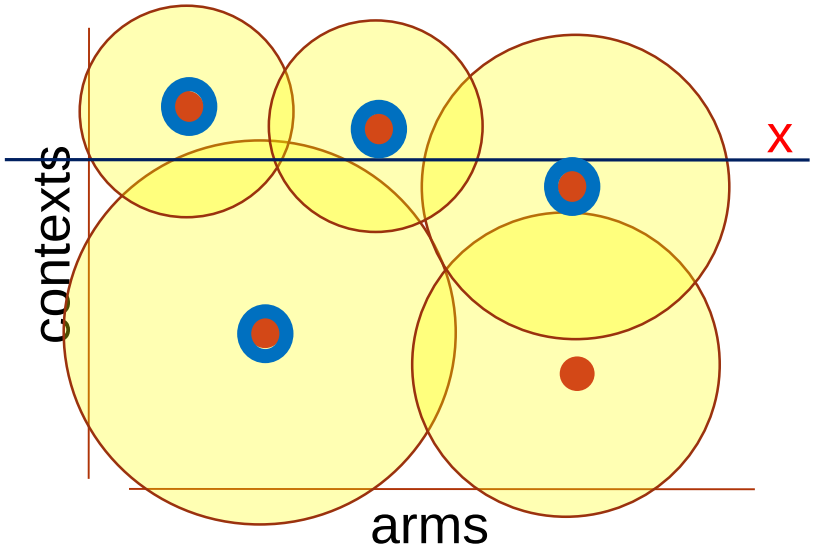
- Relaxed assumptions
  - no need for triangle inequality
  - "weak Lipschitz condition": $\mu(x^*) - \mu(y) \leq \mathbf{L}(x^*, y)$
- Special cases
  - (much) more efficient sampling if $\max_x \mu(x) = 1$
  - if $\mu(x) \equiv f(\mathbf{L}(x, S))$ *distance to target set S*
    then *ZoomingDim* = *CovDim*(S)

# Extension: contextual bandits

- *Contextual bandits*: in each round, an adversary chooses context x, an algorithm chooses arm y, and the expected payoff is $\mu(x,y)$.
  - if arms are ads, contexts are page/user profiles
- *Similarity info*: given a metric space on (x,y) pairs s.t.

$$|\mu(x,y) - \mu(x',y')| \leq L((x,y),(x',y'))$$

- *Contextual zooming algorithm* ( *Slivkins (2009)* )




active points

confidence balls:
radius reflects uncertainty

look at *relevant* active points
pick one with largest index