Multi-Dimensional Online Tracking

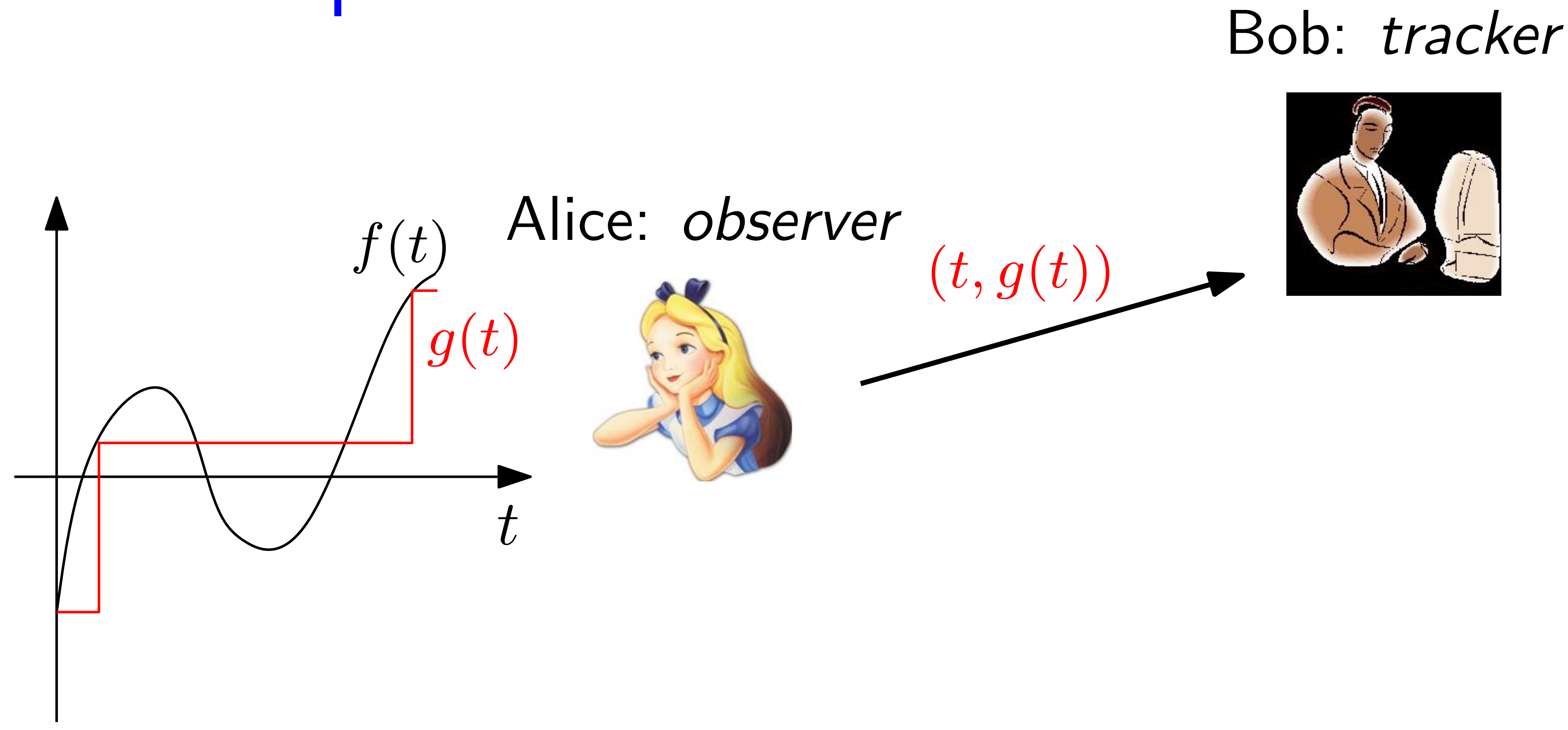
Ke Yi and Qin Zhang

Hong Kong University of Science & Technology

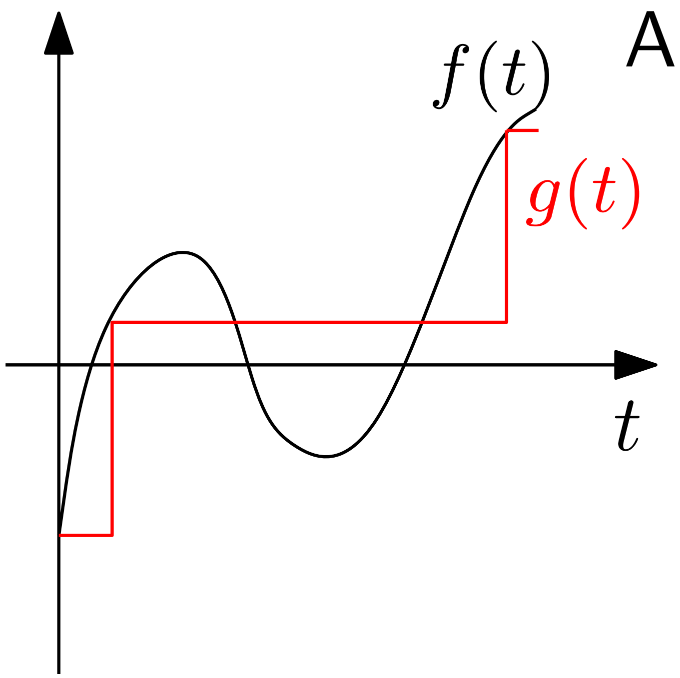
SODA 2009

January 4-6, 2009

A natural problem



A natural problem



Alice: *observer*

$(t, g(t))$

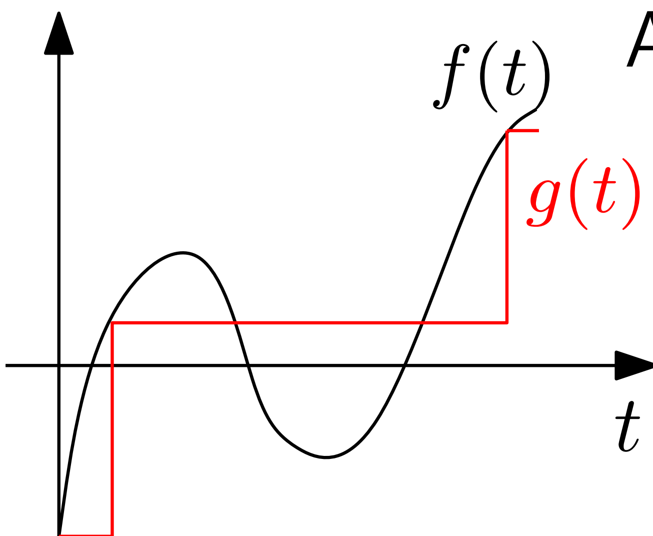
Bob: *tracker*

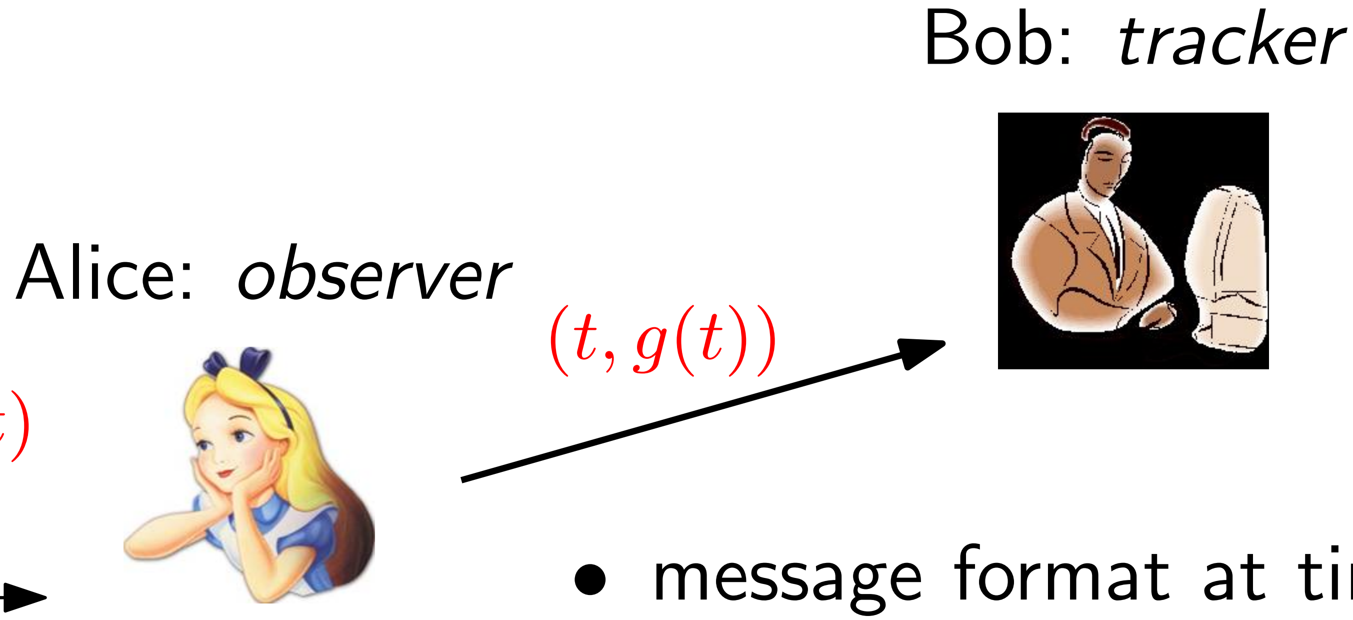
- message format at time t_{now}: $(t_{now}, g(t_{now}))$.
- communicate to guarantee that at $\forall t_{now}$

$$\|f(t_{now}) - g(t_{last})\| \leq \Delta$$

t_{last}: the last time Bob got informed.

A natural problem





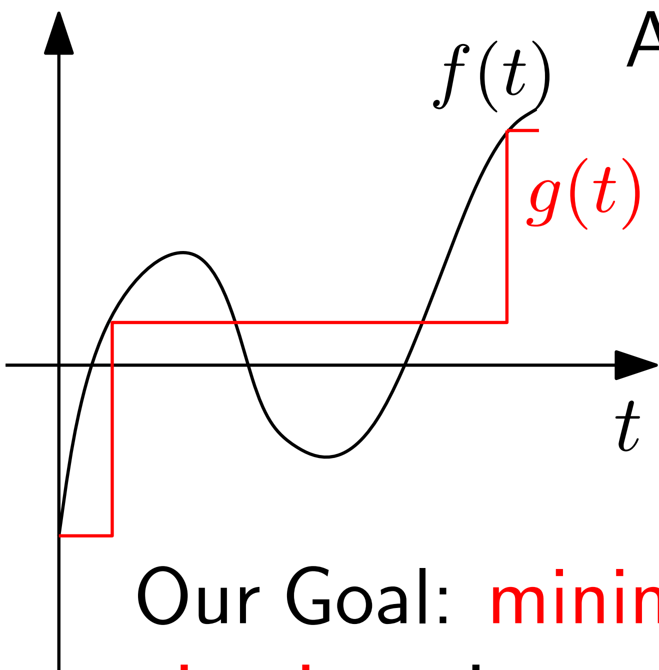
Our Goal: minimize communication, in terms of *competitive ratios.* For

- message format at time t_{now}: $(t_{now}, g(t_{now}))$.
- communicate to guarantee that at $\forall t_{now}$

$$\|f(t_{now}) - g(t_{last})\| \leq \Delta$$

t_{last}: the last time Bob got informed.

A natural problem



Alice: *observer*

$(t, g(t))$

Bob: *tracker*

Our Goal: minimize communication, in terms of *competitive ratios.* For

1. $f : Z^+ \rightarrow Z$
2. $f : Z^+ \rightarrow Z^d$
3. with prediction

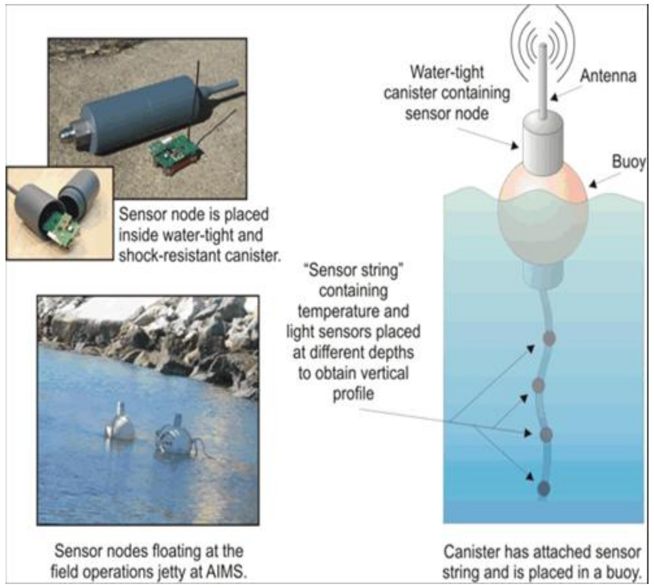
- message format at time t_{now}: $(t_{now}, g(t_{now}))$.
- communicate to guarantee that at $\forall t_{now}$

$$\|f(t_{now}) - g(t_{last})\| \leq \Delta$$

t_{last}: the last time Bob got informed.

Motivation

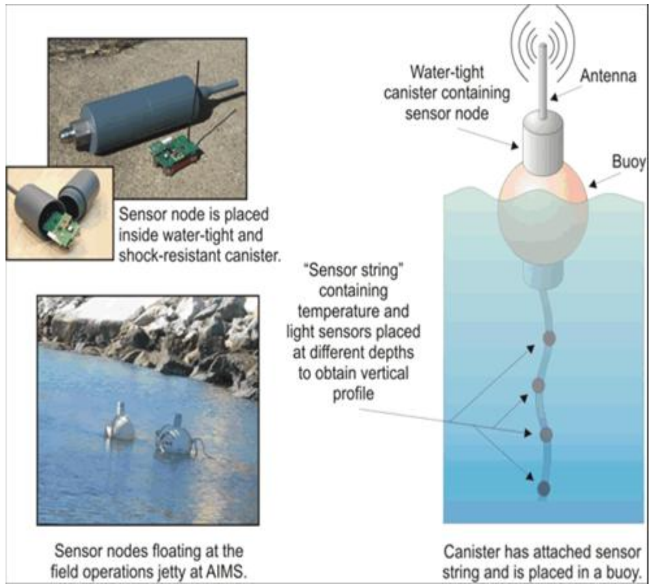
- Wireless sensors



Monitoring the temperature $\longrightarrow$ 1D case

Motivation

- Wireless sensors

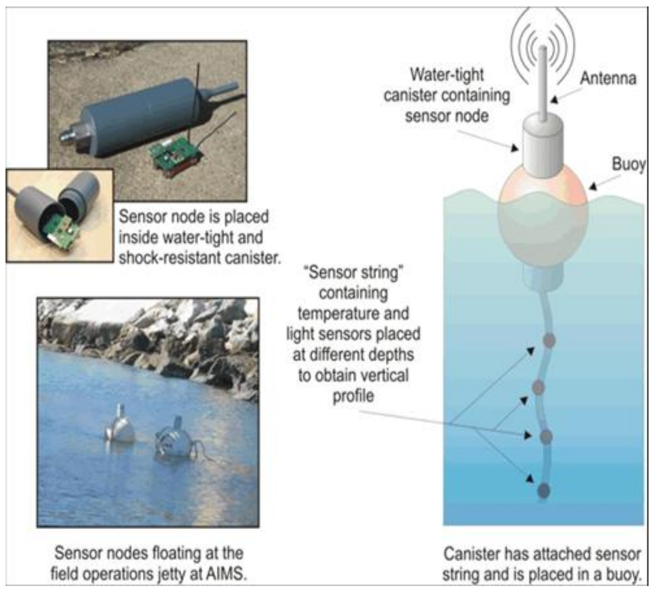


Transmission of Data is the biggest source of battery drain!

Monitoring the temperature $\longrightarrow$ 1D case

Motivation

- Wireless sensors



Monitoring the temperature → 1D case

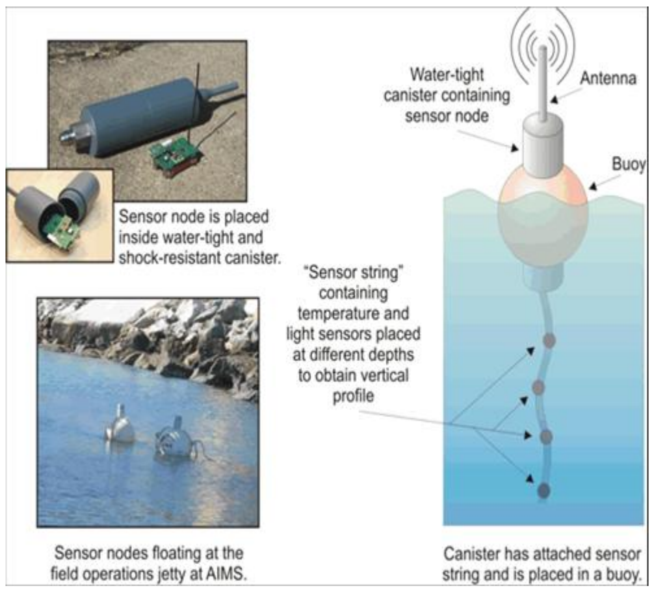
Transmission of Data is the biggest source of battery drain!

- Location-based services

Keep track of the user's location → 2D case

Motivation

- Wireless sensors



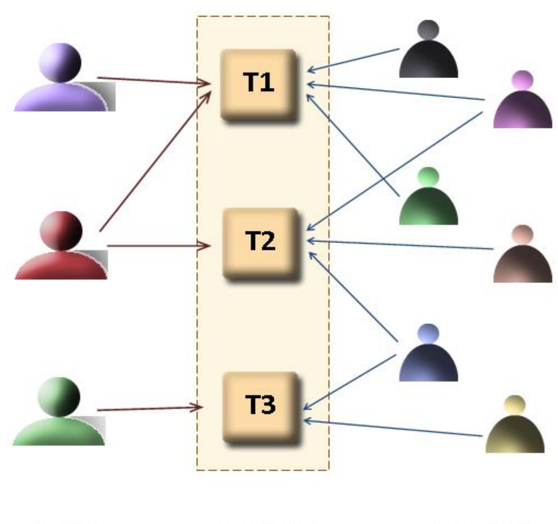
Monitoring the temperature → 1D case

Transmission of Data is the biggest source of battery drain!

- Location-based services

Keep track of the user's location → 2D case

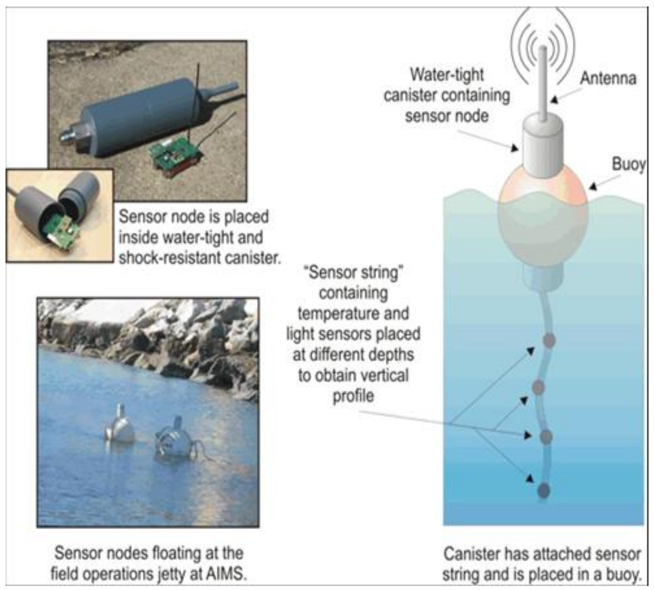
- Publish/subscribe system



Subscribers register (potentially the same) queries at the publisher; results (a set of items) change over time → high-D case

Motivation

- Wireless sensors



Monitoring the temperature → 1D case

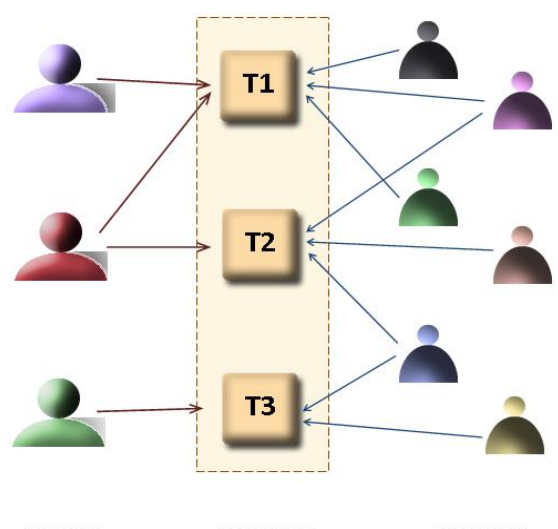
Transmission of Data is the biggest source of battery drain!

- Location-based services

Keep track of the user's location → 2D case

- Publish/subscribe system

Bandwidth consumption is the main concern!



Subscribers register (potentially the same) queries at the publisher; results (a set of items) change over time → high-D case

Naive solution fails

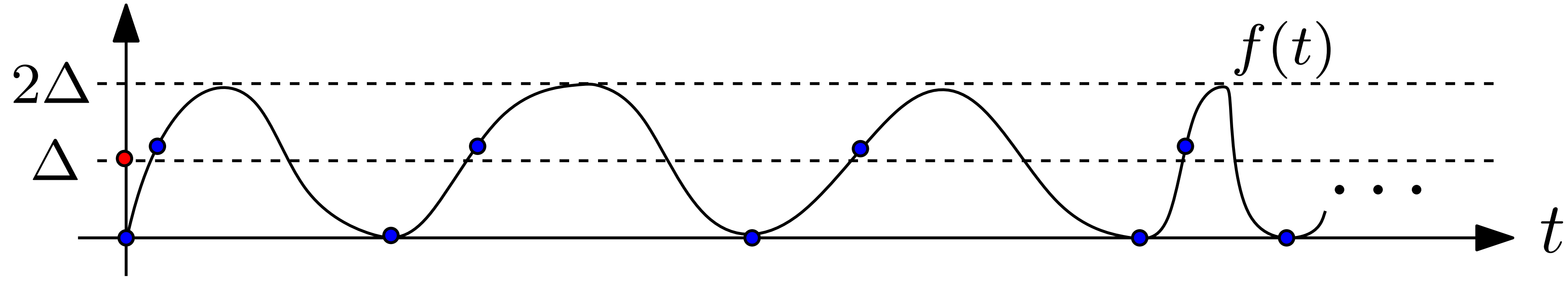
- Consider tracking the function $f : Z^+ \rightarrow Z$, and require an absolute error of at most Δ.
  - The natural solution is to
    1. first communicate $f(0)$ to Bob.
    2. every time $f(t)$ has changed by more than Δ since the last communication, Alice updates Bob with the current $f(t)$.

Naive solution fails

- Consider tracking the function $f : Z^+ \rightarrow Z$, and require an absolute error of at most Δ.
  - The natural solution is to
    1. first communicate $f(0)$ to Bob.
    2. every time $f(t)$ has changed by more than Δ since the last communication, Alice updates Bob with the current $f(t)$.
  - Unbounded competitive ratio!

Naive solution fails

- Consider tracking the function $f : Z^+ \to Z$, and require an absolute error of at most Δ.
  - The natural solution is to
    1. first communicate $f(0)$ to Bob.
    2. every time $f(t)$ has changed by more than Δ since the last communication, Alice updates Bob with the current $f(t)$.
  - Unbounded competitive ratio!



SOL $= \infty$, OPT $= 1$!

Our Results

problem	comp. ratio	running time
1-dim	$O(\log \Delta)$	$O(1)$
d-dim	$O(d^2 \log(d\Delta))$	$\text{poly}(d, \log \Delta)$
1-dim + prediction	$O(\log(\Delta T))$	$\text{poly}(\Delta, T)$

Results for online tracking. T: length of the tracking period.

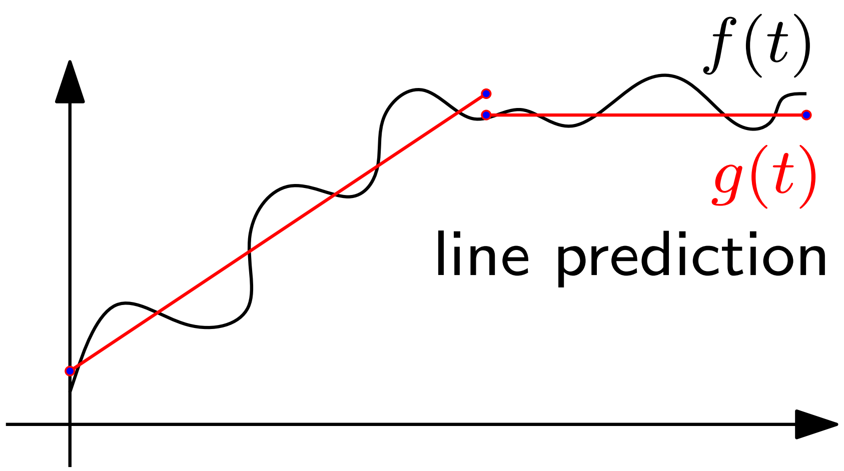
Our Results

problem	comp. ratio	running time
1-dim	$O(\log \Delta)$	$O(1)$
d-dim	$O(d^2 \log(d\Delta))$	$\mathrm{poly}(d, \log \Delta)$
1-dim + prediction	$O(\log(\Delta T))$	$\mathrm{poly}(\Delta, T)$

Results for online tracking. T: length of the tracking period.

- Prediction.

  Allow to send prediction functions (e.g. linear functions) instead of only a single value every time. OPT also uses the same family of functions.



Related research domains

- Communication complexity

Alice (has x) $\overset{\text{compute } f(x,y)}{\Longleftrightarrow}$ Bob (has y), x, y are given offline.

Our case.

1. Alice: observer, Bob: tracker.
2. Inputs arrive online, only seen by Alice.

Related research domains

- Communication complexity

  Alice (has x) $\overset{\text{compute } f(x,y)}{\Longleftrightarrow}$ Bob (has y), x, y are given offline.

  Our case.

  1. Alice: observer, Bob: tracker.
  2. Inputs arrive online, only seen by Alice.

- Data streams

  Small space.

  Our case: communication cost.

One dimension

- General idea to track $f : Z^+ \to Z$.

Divide the whole tracking period into rounds, and show that $\mathcal{A}_{\texttt{OPT}}$ must communicate once in each round, while our algorithm communicates at most, say, k times

$\to$ competitive ratio k.

One dimension (Cont.)

- The Algorithm to track $f : Z^+ \rightarrow Z$

Algorithm 1: One round of 1D tracking

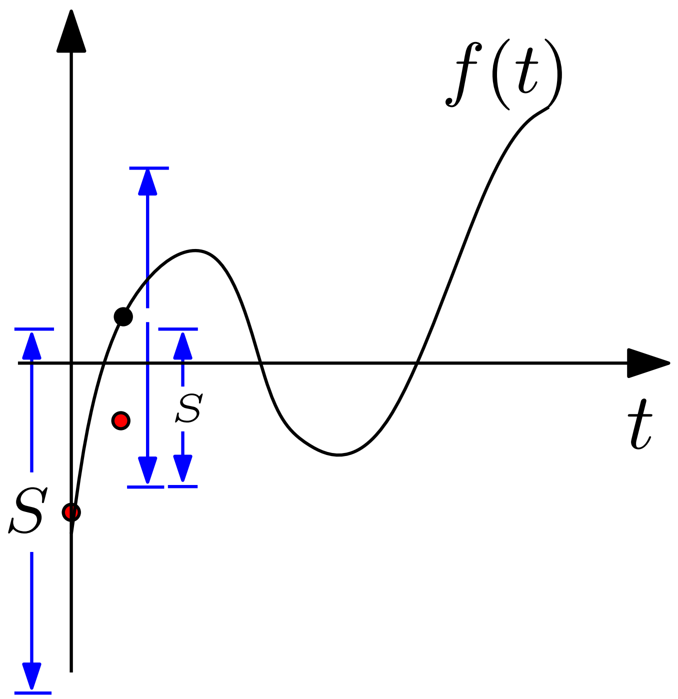
let $S = [f(t_{now}) - \Delta, f(t_{now}) + \Delta] \cap \mathbb{Z}$;
while $S \neq \emptyset$ **do**
    let $g(t_{now})$ be the median of S;
    send $g(t_{now})$ to Bob;
    wait until $\|f(t_{now}) - g(t_{last})\| > \Delta$;
    $S \leftarrow S \cap [f(t_{now}) - \Delta, f(t_{now}) + \Delta]$;

One dimension (Cont.)

- The Algorithm to track $f : Z^+ \rightarrow Z$

Algorithm 1: One round of 1D tracking

let $S = [f(t_{now}) - \Delta, f(t_{now}) + \Delta] \cap \mathbb{Z}$;
while $S \neq \emptyset$ **do**
   let $g(t_{now})$ be the median of S;
   send $g(t_{now})$ to Bob;
   wait until $\|f(t_{now}) - g(t_{last})\| > \Delta$;
   $S \leftarrow S \cap [f(t_{now}) - \Delta, f(t_{now}) + \Delta]$;

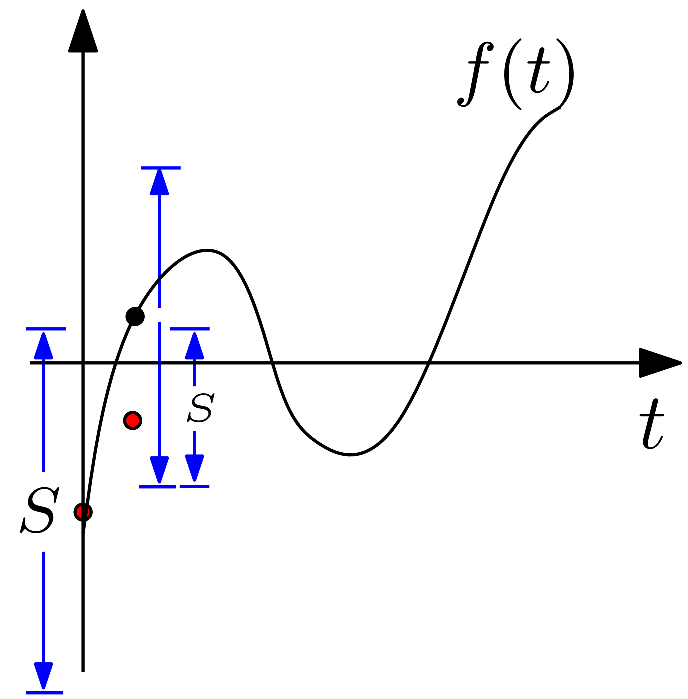


One dimension (Cont.)

- The Algorithm to track $f : Z^+ \to Z$

Algorithm 1: One round of 1D tracking

```
let S = [f(t_now) − Δ, f(t_now) + Δ] ∩ ℤ;
while S ≠ ∅ do
    let g(t_now) be the median of S;
    send g(t_now) to Bob;
    wait until ||f(t_now) − g(t_last)|| > Δ;
    S ← S ∩ [f(t_now) − Δ, f(t_now) + Δ];
```



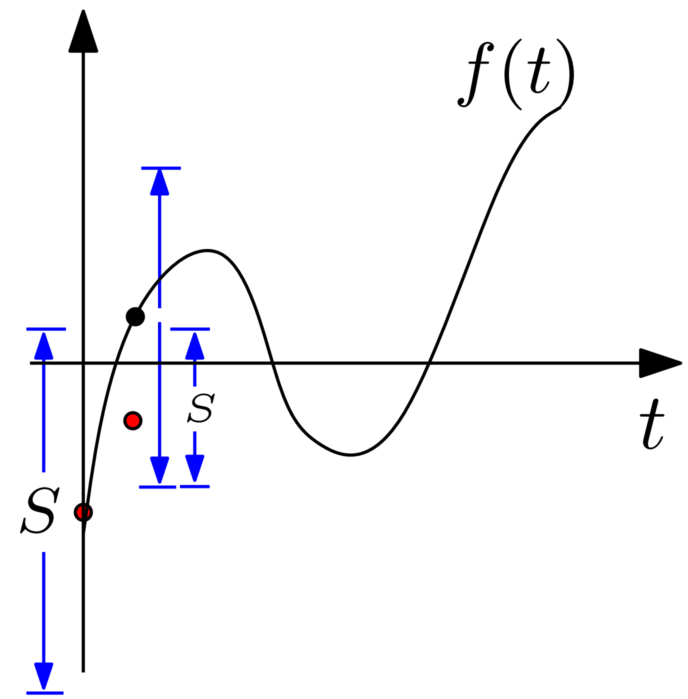
- The Analysis
  - If $\mathcal{A}_{\texttt{OPT}}$ hasn't sent a message in the current round, then its last message must be included in S.
  - The cardinality of S decreases by half whenever Algorithm 1 sends a message.

One dimension (Cont.)

- The Algorithm to track $f : Z^+ \rightarrow Z$

Algorithm 1: One round of 1D tracking

let $S = [f(t_{now}) - \Delta, f(t_{now}) + \Delta] \cap \mathbb{Z}$;
while $S \neq \emptyset$ **do**
  let $g(t_{now})$ be the median of S;
  send $g(t_{now})$ to Bob;
  wait until $\|f(t_{now}) - g(t_{last})\| > \Delta$;
  $S \leftarrow S \cap [f(t_{now}) - \Delta, f(t_{now}) + \Delta]$;



- The Analysis
  - If $\mathcal{A}_{\texttt{OPT}}$ hasn't sent a message in the current round, then its last message must be included in S.
  - The cardinality of S decreases by half whenever Algorithm 1 sends a message.

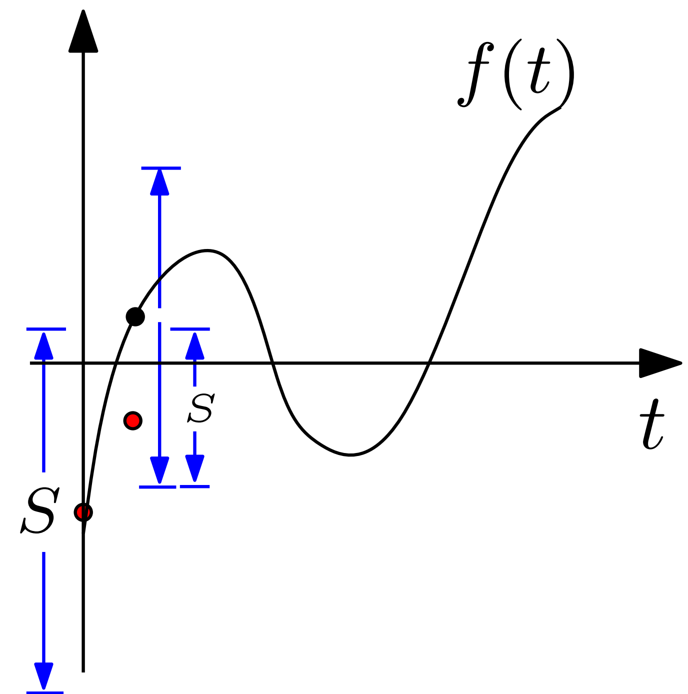
$\Rightarrow O(\log \Delta)$ -competitive

One dimension (Cont.)

- The Algorithm to track $f : Z^+ \rightarrow Z$

Algorithm 1: One round of 1D tracking

let $S = [f(t_{now}) - \Delta, f(t_{now}) + \Delta] \cap \mathbb{Z}$;
while $S \neq \emptyset$ **do**
  let $g(t_{now})$ be the median of S;
  send $g(t_{now})$ to Bob;
  wait until $\|f(t_{now}) - g(t_{last})\| > \Delta$;
  $S \leftarrow S \cap [f(t_{now}) - \Delta, f(t_{now}) + \Delta]$;



- The Analysis
  - If $\mathcal{A}_{\texttt{OPT}}$ hasn't sent a message in the current round, then its last message must be included in S.
  - The cardinality of S decreases by half whenever Algorithm 1 sends a message.

$\Rightarrow O(\log \Delta)$ -competitive

Also tight!

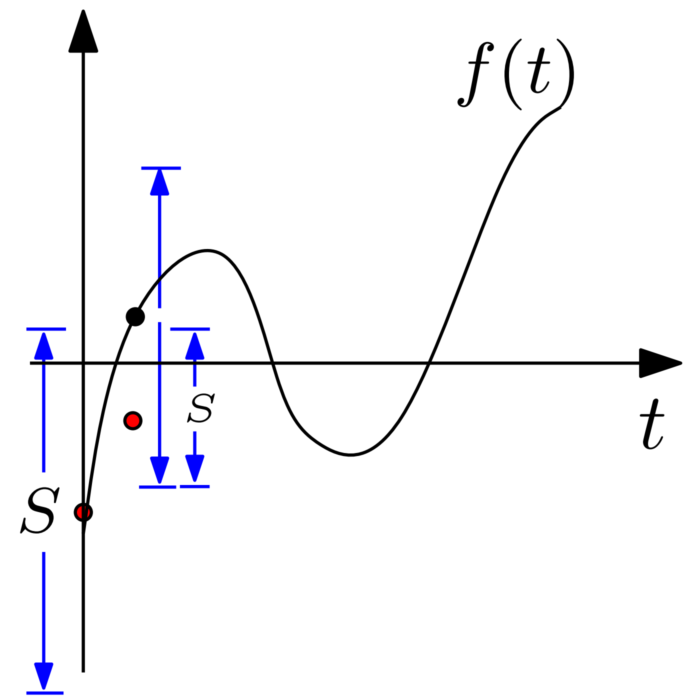
One dimension (Cont.)

- The Algorithm to track $f : Z^+ \to Z$

Real range unbounded!

Algorithm 1: One round of 1D tracking

> let $S = [f(t_{now}) - \Delta, f(t_{now}) + \Delta] \cap \mathbb{Z}$;
> **while** $S \neq \emptyset$ **do**
>   let $g(t_{now})$ be the median of S;
>   send $g(t_{now})$ to Bob;
>   wait until $\|f(t_{now}) - g(t_{last})\| > \Delta$;
> $S \leftarrow S \cap [f(t_{now}) - \Delta, f(t_{now}) + \Delta]$;



- The Analysis
  - If $\mathcal{A}_{\texttt{OPT}}$ hasn't sent a message in the current round, then its last message must be included in S.
  - The cardinality of S decreases by half whenever Algorithm 1 sends a message.

$\Rightarrow O(\log \Delta)$ -competitive

Also tight!

High dimensions

- The general idea follows from 1D

  Divide the whole tracking period into rounds, and show that the competitive ratio in each round is k.

High dimensions

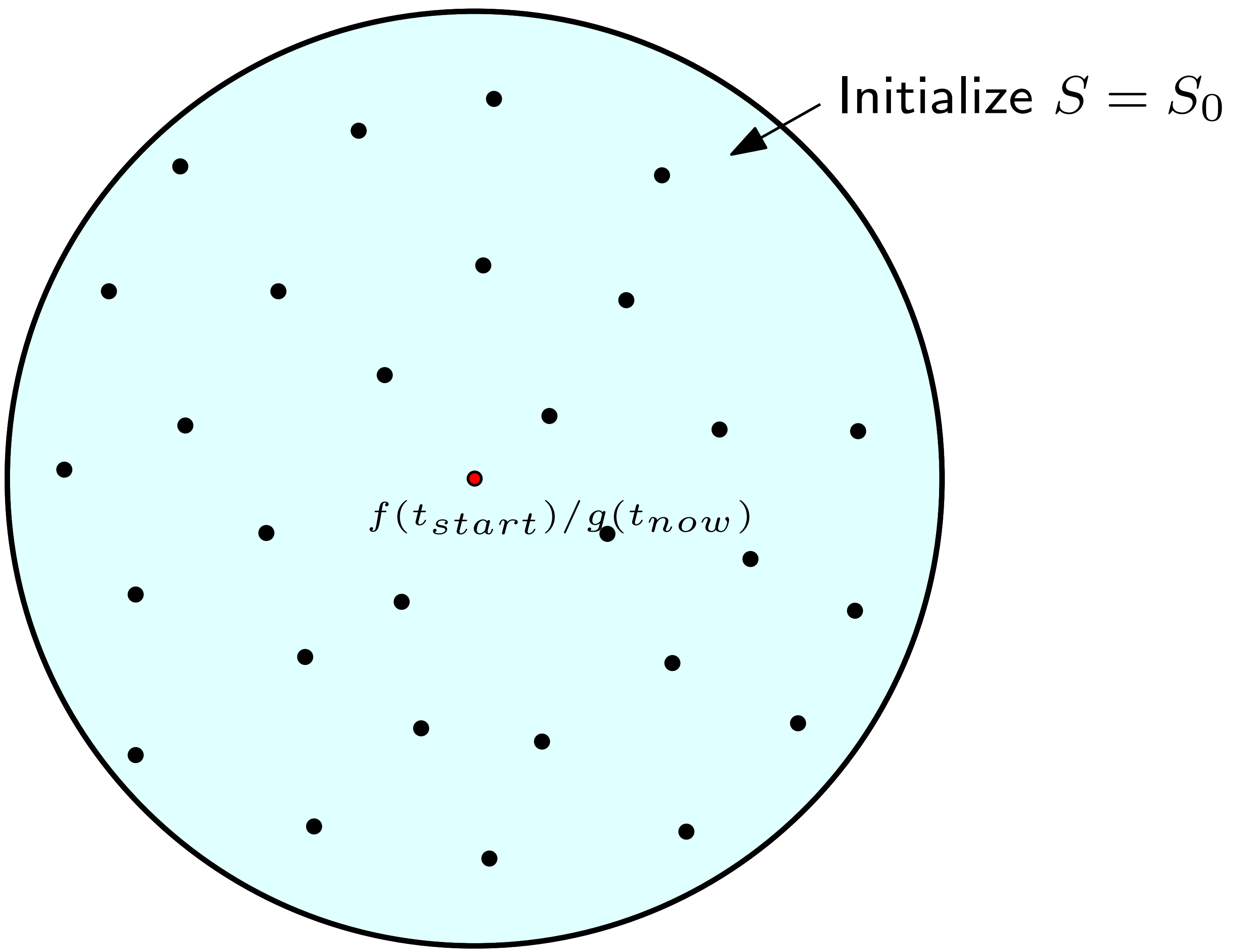
- The general idea follows from 1D

  Divide the whole tracking period into rounds, and show that the competitive ratio in each round is k.

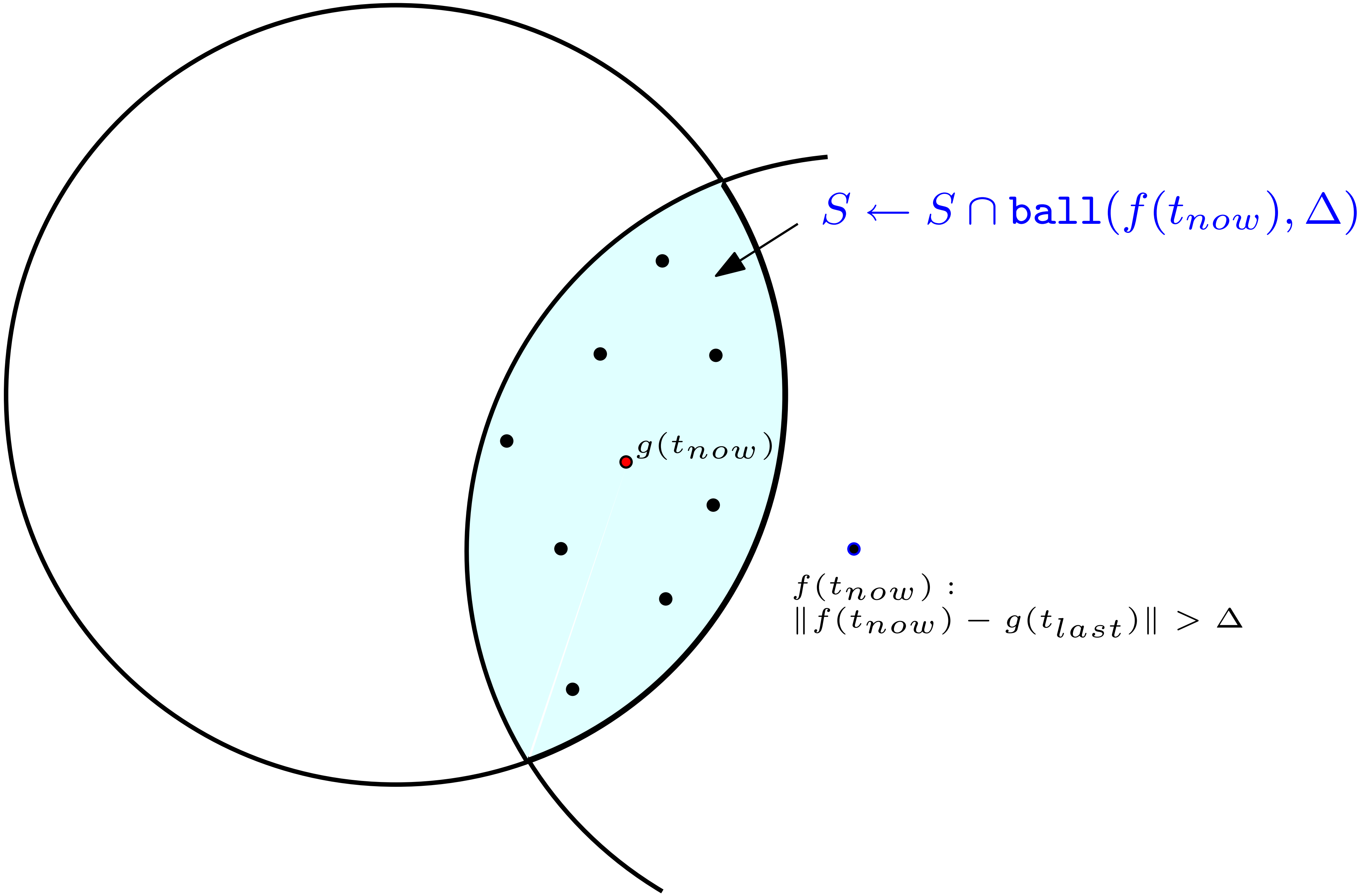
- The framework of one round

  1. At time $t = t_{start}$, initialize a set $S = S_0$. (Many choices of S_0)
  2. In each iteration in the while loop, we first pick a "median" from S as $g(t_{now})$ and send it to Bob.
  3. When f deviates from $g(t_{last})$ by more than Δ, we cut S as $S \leftarrow S \cap \texttt{ball}(f(t_{now}), \Delta)$.
  4. When S becomes empty, we can terminate the round.

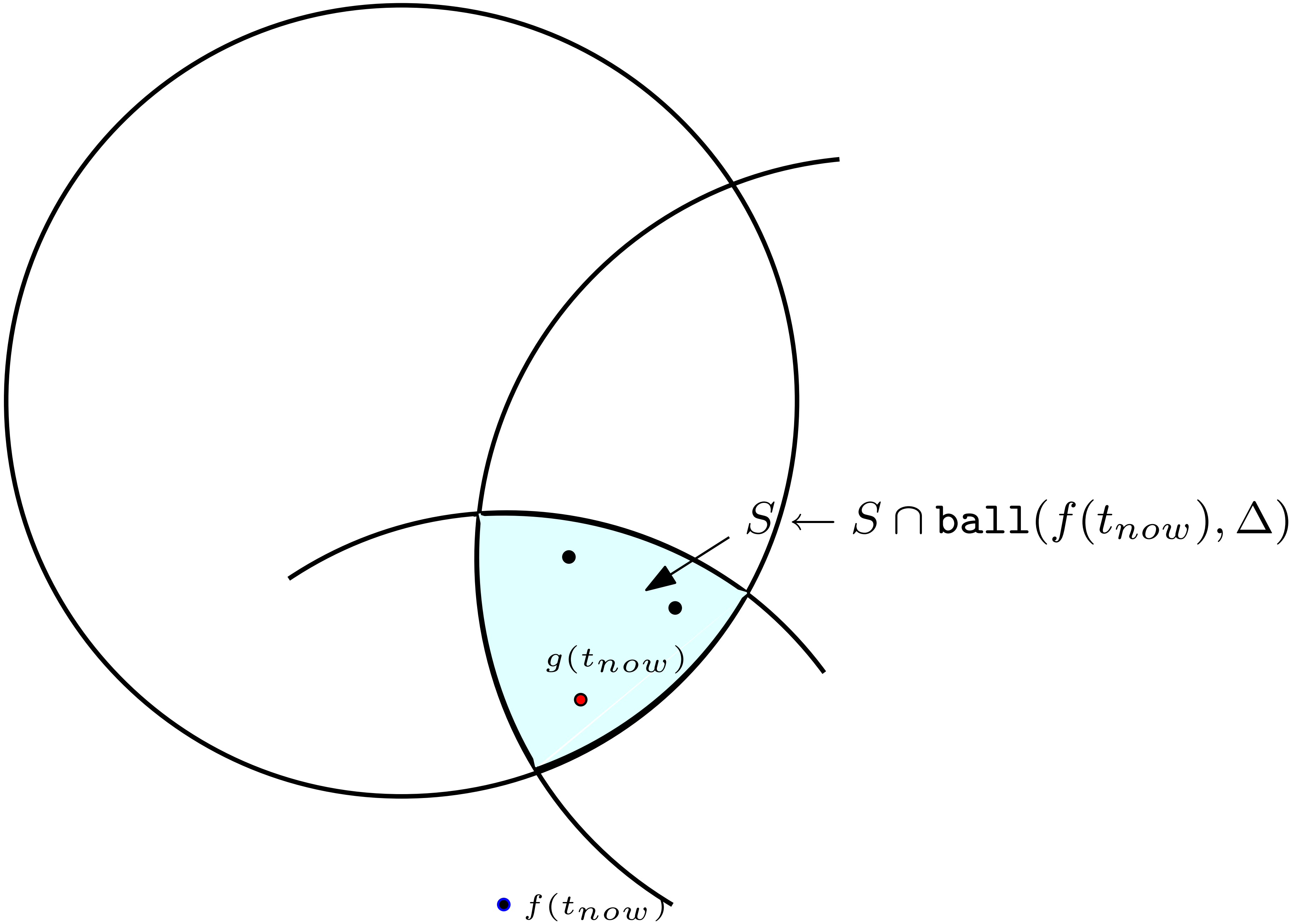
High dimensions (cont.)



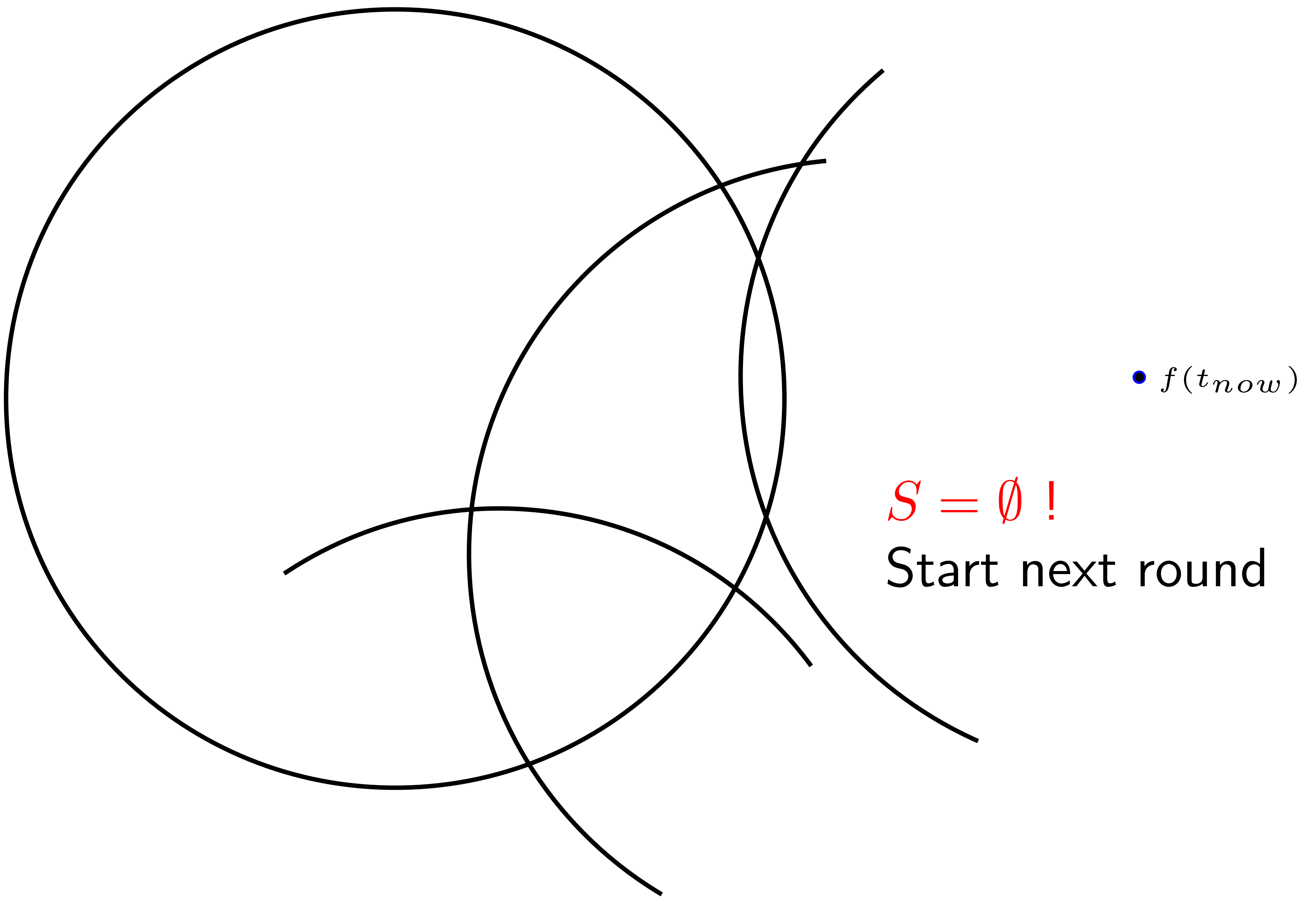
High dimensions (cont.)



High dimensions (cont.)



High dimensions (cont.)



$S = \emptyset$!

Start next round

High dimensions (cont.)

- The key property of S.

  If S becomes empty at some time step, then $\mathcal{A}_{\mathtt{OPT}}$ must have communicated once in the current round.

High dimensions (cont.)

- The key property of S.

  If S becomes empty at some time step, then $\mathcal{A}_{\mathtt{OPT}}$ must have communicated once in the current round.

- Two main issues left ...

  1. How to choose the initial set S_0 so that above property is met?

  2. How to pick the median so that we can have small competitive ratios?

Take one, Tukey medians

- Choices for two issues
  - Set $S_0 = C_2 \cup C_3 \ldots \cup C_{d+1}$

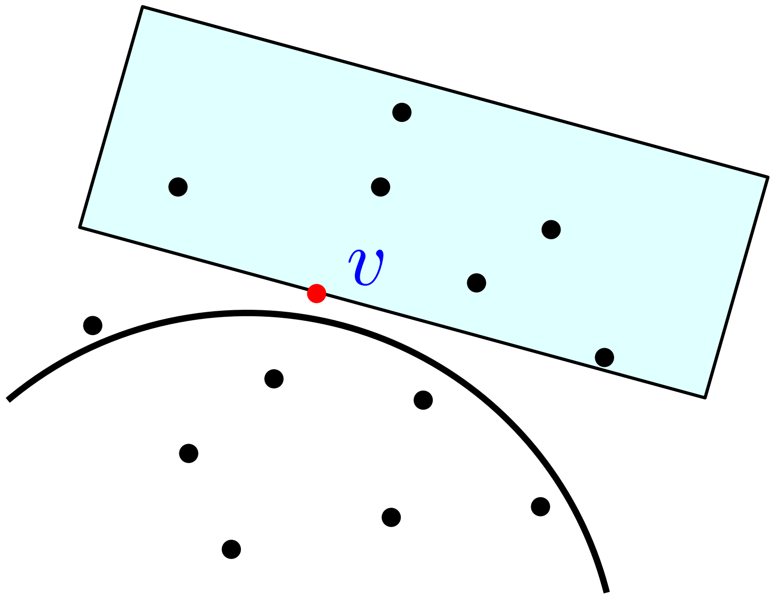
    C_l: be the collection of centers of the smallest enclosing balls of every l points in $\texttt{Ball}(f(t_{start}), 2\Delta) \cap Z^d$.
  - Send the Tukey median of S at every triggering.

Take one, Tukey medians

- Choices for two issues
  - Set $S_0 = C_2 \cup C_3 \ldots \cup C_{d+1}$

    C_l: be the collection of centers of the smallest enclosing balls of every l points in $\texttt{Ball}(f(t_{start}), 2\Delta) \cap Z^d$.
  - Send the Tukey median of S at every triggering.
- Definition of the Tukey medians

Any halfspace containing Tukey median v also contains at least $\frac{1}{d+1}n$ points where n is the cardinality of the point set.



Tukey medians (cont.)

- Analysis of competitive ratio (ρ)
  - $$S_0 = O\left(d\left(\frac{e(\lfloor 4\Delta \rfloor+1)^d}{d+1}\right)^{d+1}\right)$$
  - $|S|$ deceases by a factor of at least $1/(d+1)$ at every triggering of communication $\Rightarrow$ $\rho = \log_{1+\frac{1}{d}} |S_0| = O(d^3 \log \Delta)$

Tukey medians (cont.)

- Analysis of competitive ratio (ρ)
  - $S_0 = O\left(d\left(\frac{e(\lfloor 4\Delta \rfloor + 1)^d}{d+1}\right)^{d+1}\right)$
  - $|S|$ deceases by a factor of at least $1/(d+1)$ at every triggering of communication $\Rightarrow$ $\rho = \log_{1+\frac{1}{d}} |S_0| = O(d^3 \log \Delta)$
- However, the running time is exponential in d :(
  1. S_0 is too large.
  2. Computing Tukey medians in high-D (even approximately) is hard.

Volume cutting

$\Rightarrow O(d^2 \log d\Delta)$-competitive;
running time polynomial in d and $\log \Delta$.

Volume cutting

$\Rightarrow O(d^2 \log d\Delta)$-competitive;
running time polynomial in d and $\log \Delta$.

- Send the (approximate) centroids of a convex set containing S.

Volume cutting

$\Rightarrow O(d^2 \log d\Delta)$-competitive;
running time polynomial in d and $\log \Delta$.

- Send the (approximate) centroids of a convex set containing S.
- Use some geometry to bound the number of cuts performed until $|S| \leq 1$.

Volume cutting

$\Rightarrow O(d^2 \log d\Delta)$-competitive;
running time polynomial in d and $\log \Delta$.

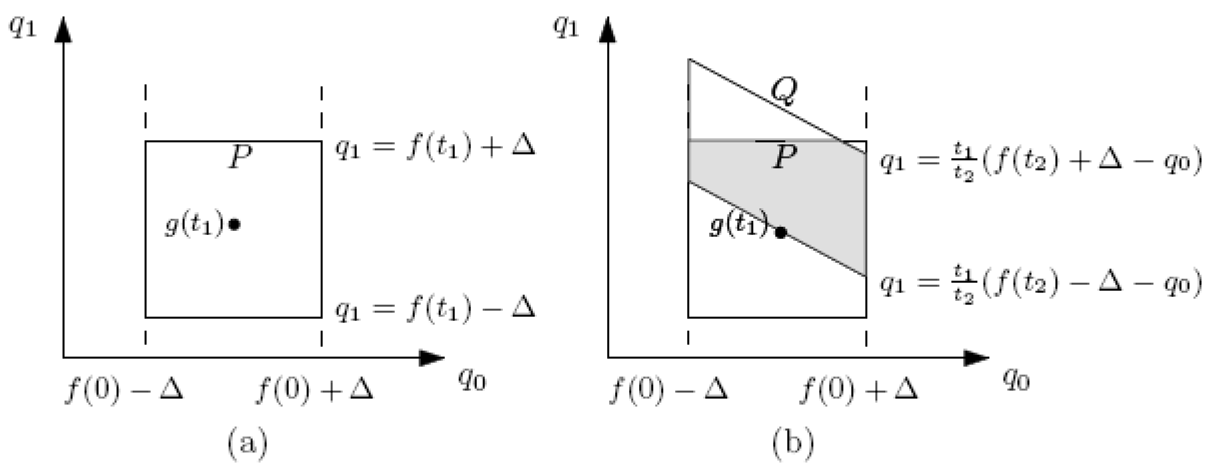
- Send the (approximate) centroids of a convex set containing S.
- Use some geometry to bound the number of cuts performed until $|S| \leq 1$.
- Techniques similar to those used in convex programming to find the last point of S if exists.

With prediction

- We consider the case that algorithms are allowed to send a linear function to predict the future trend of f

With prediction

- We consider the case that algorithms are allowed to send a linear function to predict the future trend of f
- Ideas.
  1. Still follow the general framewok.
  2. Cutting in the parametric space.



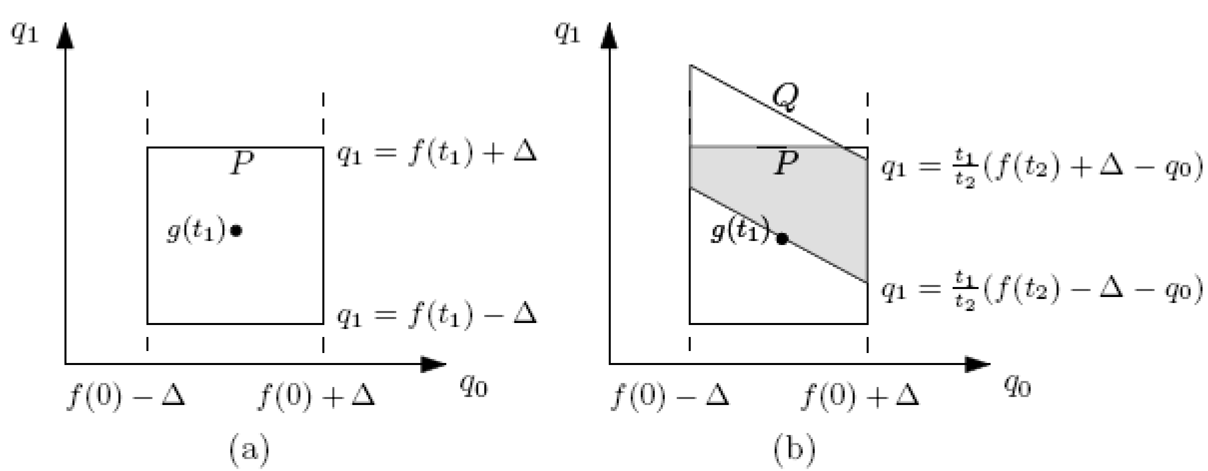
A line l passing $(0, q_0), (t_1, q_1)$ $\longrightarrow$ a point (q_0, q_1) in 2D.

With prediction

- We consider the case that algorithms are allowed to send a linear function to predict the future trend of f

- Ideas.

    1. Still follow the general framewok.
    2. Cutting in the parametric space.



(a) (b)

A line l passing $(0, q_0), (t_1, q_1)$ $\longrightarrow$ a point (q_0, q_1) in 2D.

- Competitive ratio: $O(\log(\Delta T))$. T : length of the tracking period,

Open problems and future directions

- Generalize our techniques to multiple observers.

- What if the length of the messages is considered (in the high-D) case?

- Lower bounds for high dimensional tracking.

- Online tracking in other metric spaces.

Open problems and future directions

- Generalize our techniques to multiple observers.
  - Need a stronger model.
- What if the length of the messages is considered (in the high-D) case?
- Lower bounds for high dimensional tracking.
- Online tracking in other metric spaces.

The End

THANK YOU

Q and A