Leveraging Structure in Nonstochastic Bandit Problems: Some Examples

Claudio Gentile

INRIA and Google

cla.gentile@gmail.com

April 24th, 2018

New York University

Joint with:

N. Alon, N. Cesa-Bianchi, P. Gaillard, S. Gerchinovitz, Y. Mansour, A. Minora, S. Mannor, O. Shamir

Goal of this presentation

Recent activity in the analysis of structured bandit problems
in nonstochastic settings

Requirements:

- Structure has to be meaningful and
- Regret analyses have to capture this structure in appropriate ways

Outline

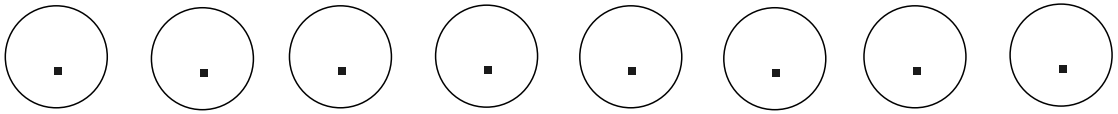
- Nonstochastic bandit game (vanilla + some extensions)
- (Delayed) cooperation among nonstochastic bandit agents in distributed environments
- Learning nonparametric policies in a nonstochastic setting

Outline

- Nonstochastic bandit game (vanilla + some extensions)
- (Delayed) cooperation among nonstochastic bandit agents in distributed environments
- Learning nonparametric policies in a nonstochastic setting

Nonstochastic bandit game/1

N actions for Player



For $t = 1, 2, \ldots$:

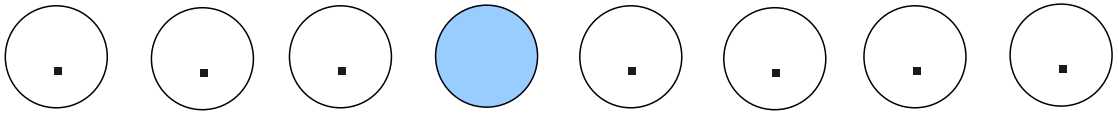
1. Losses $\ell_t(i) \in [0, 1]$ are assigned (deterministically) by opponent to every action $i = 1 \ldots N$ (hidded to player)

2. Player picks action I_t (possibly using randomization) and incurs loss $\ell_t(I_t)$

3. Player gets feedback information: $\ell_t(I_t)$

Nonstochastic bandit game/1

N actions for Player



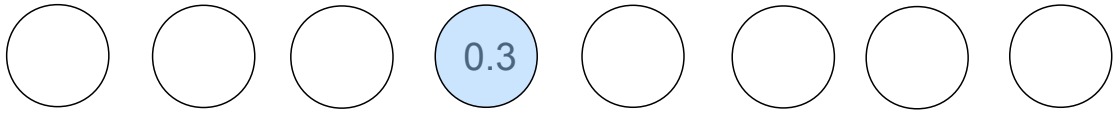
For $t = 1, 2, \ldots$:

1. Losses $\ell_t(i) \in [0, 1]$ are assigned (deterministically by opponent to every action $i = 1 \ldots N$ (hidded to player)

2. Player picks action I_t (possibly using randomization) and incurs loss $\ell_t(I_t)$

3. Player gets feedback information: $\ell_t(I_t)$

Nonstochastic bandit game/1



For $t = 1, 2, \ldots$:

1. Losses $\ell_t(i) \in [0, 1]$ are assigned (deterministically by opponent to every action $i = 1 \ldots N$ (hidden to player)

2. Player picks action I_t (possibly using randomization) and incurs loss $\ell_t(I_t)$

3. Player gets feedback information: $\ell_t(I_t)$

Nonstochastic bandit game/2

Goal [external regret]: Given T rounds, Player's total loss

$$\sum_{t=1}^{T} \ell_t(I_t)$$

must be close to that of single best action in hindsight for Player

Regret of Player for T rounds:

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(I_t)\right] - \min_{i=1\ldots N} \sum_{t=1}^{T} \ell_t(i)$$

Want : $R_T = o(T)$ as T grows large ("no regret")

Lower bound:

$$\Omega(\sqrt{TN})$$

Nonstochastic bandit game/3: Exp3 Alg. [Auer et al. 02]

At round t pick action $I_t = i$ with probability proportional to

$$\exp\left(-\eta \sum_{s=1}^{t-1} \widehat{\ell}_s(i)\right), \qquad i = 1 \ldots N$$

$$\widehat{\ell}_s(i) = \begin{cases} \dfrac{\ell_s(i)}{\Pr_s(\ell_s(i) \text{ is observed in round } s)} & \text{if } \ell_s(i) \text{ is observed} \\ 0 & \text{otherwise} \end{cases}$$

- Only one nonzero component in $\widehat{\ell}_t$
- Exponentially-weighted alg with (importance sampling) loss estimates

$$\widehat{\ell}_t(i) \approx \ell_t(i)$$

- Upper bound on regret:

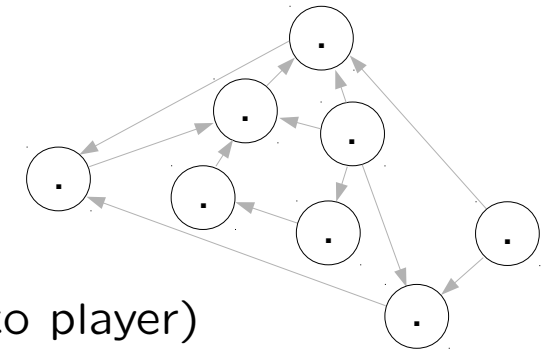
$$R_T \leq \sqrt{TN \ln N}$$

- Improved upper bound: $O(\sqrt{TN})$ (the INF alg.) [AB09]

Side-info over actions/1

[MS11,A+13,K+15]

N actions for Player



Before game starts, sequence of
feedback graphs $G_t = (V, E_t)$
$V = \{1, \ldots, N\}$
generated by exogenous source (hidden to player)
All self-loops included

For $t = 1, 2, \ldots$:

1. Losses $\ell_t(i) \in [0, 1]$ are assigned deterministically by opponent to every action $i = 1 \ldots N$ (hidded to player)

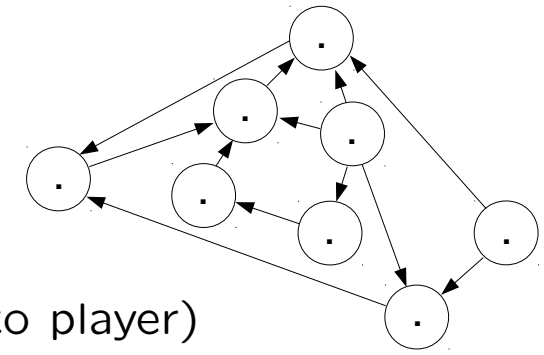
2. Player picks action I_t (possibly using randomization) and incurs loss $\ell_t(I_t)$

3. Player gets feedback information: $\{\ell_t(j) : (I_t, j) \in E_t\}$

Side-info over actions/1

[MS11,A+13,K+15]

N actions for Player



Before game starts, sequence of
feedback graphs $G_t = (V, E_t)$
$V = \{1, \ldots, N\}$
generated by exogenous source (hidden to player)
All self-loops included

For $t = 1, 2, \ldots$:

1. Losses $\ell_t(i) \in [0, 1]$ are assigned deterministically by opponent to every action $i = 1 \ldots N$ (hidded to player)

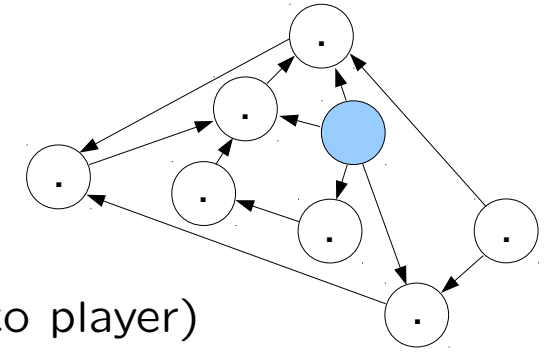
2. Player picks action I_t (possibly using randomization) and incurs loss $\ell_t(I_t)$

3. Player gets feedback information: $\{\ell_t(j) : (I_t, j) \in E_t\}$

Side-info over actions/1

[MS11,A+13,K+15]

N actions for Player



Before game starts, sequence of
feedback graphs $G_t = (V, E_t)$
$V = \{1, \ldots, N\}$
generated by exogenous source (hidden to player)
All self-loops included

For $t = 1, 2, \ldots$:

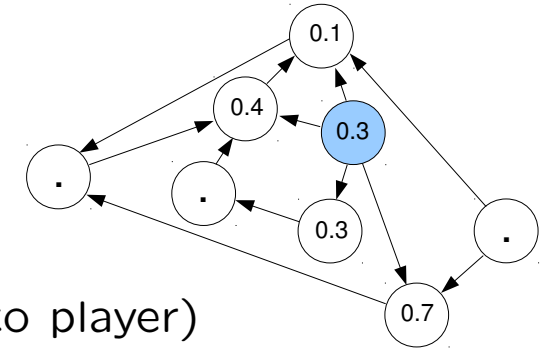
1. Losses $\ell_t(i) \in [0, 1]$ are assigned deterministically by opponent to every action $i = 1 \ldots N$ (hidded to player)

2. Player picks action I_t (possibly using randomization) and incurs loss $\ell_t(I_t)$

3. Player gets feedback information: $\{\ell_t(j) : (I_t, j) \in E_t\}$

Side-info over actions/1

[MS11,A+13,K+15]



N actions for Player

Before game starts, sequence of
feedback graphs $G_t = (V, E_t)$
$V = \{1, \ldots, N\}$
generated by exogenous source (hidden to player)
All self-loops included

For $t = 1, 2, \ldots$:

1. Losses $\ell_t(i) \in [0, 1]$ are assigned deterministically by opponent to every action $i = 1 \ldots N$ (hidded to player)

2. Player picks action I_t (possibly using randomization) and incurs loss $\ell_t(I_t)$

3. Player gets feedback information: $\{\ell_t(j) : (I_t, j) \in E_t\}$

Side-info over actions/2: Exp3-IX Alg. [K+15]

At round t pick action $I_t = i$ with probability proportional to

$$\exp\left(-\eta \sum_{s=1}^{t-1} \widehat{\ell}_s(i)\right), \qquad i = 1 \dots N$$

$$\widehat{\ell}_s(i) = \begin{cases} \dfrac{\ell_s(i)}{\gamma_t + \Pr_s(\ell_s(i) \text{ is observed in round } s)} & \text{if } \ell_s(i) \text{ is observed} \\ 0 & \text{otherwise} \end{cases}$$

- Note: prob. of observing loss of action $\neq$ prob. of playing action
- Exponentially-weighted alg with γ_t-biased (importance sampling) loss estimates

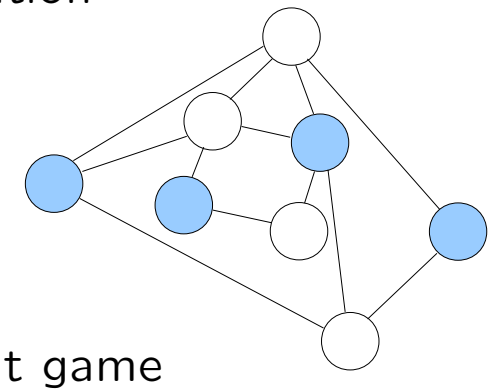
$$\widehat{\ell}_t(i) \approx \ell_t(i)$$

- Bias is controlled by $\gamma_t = 1/\sqrt{t}$

Side-info over actions/3

[A+13,K+15]

Independence number $\alpha(G_t)$: disregard edge orientation



clique: $\underbrace{1}$ expert game $\leq \quad \alpha(G_t) \quad \leq$ edgeless: $\underbrace{N}$ bandit game

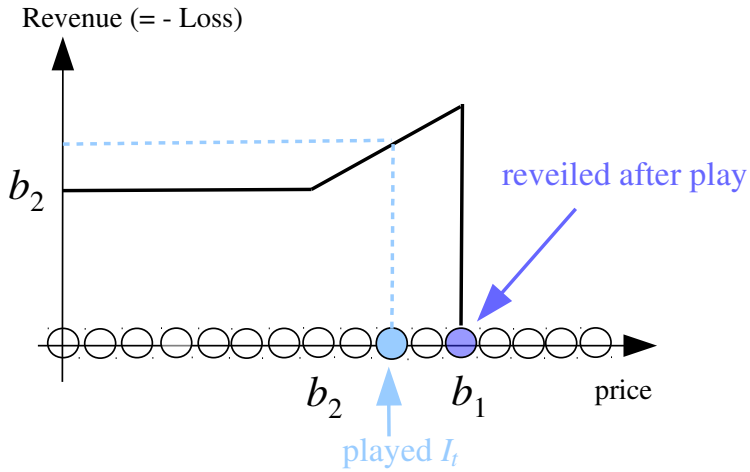
Regret analysis:

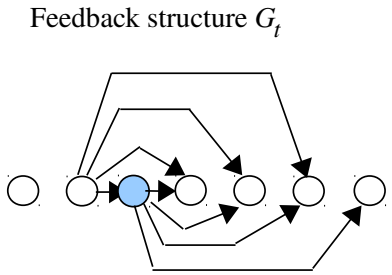
$$R_T = O\left(\ln(TN)\sqrt{\sum_{t=1}^{T}\alpha(G_t)}\right)$$

If $G_t = G \ \forall t$:

$$R_T = \tilde{O}\left(\sqrt{T\alpha(G)}\right)$$

Side-info over actions/4: Simple example



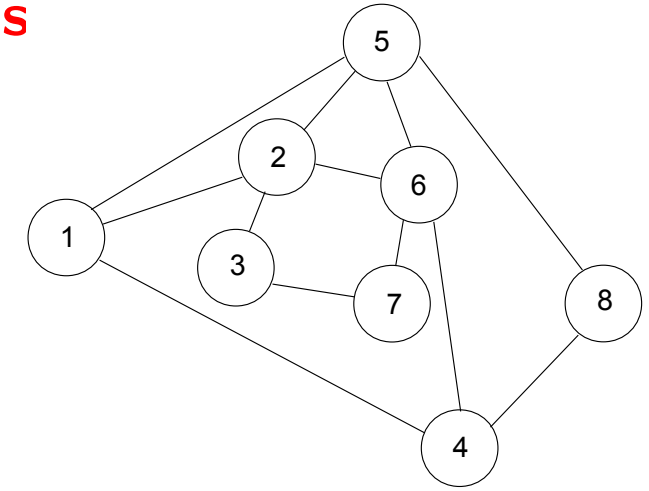


- Second-price auction with reserve (seller side)
  highest bid revealed to seller (e.g. AppNexus)
- Auctioneer is third party
- After seller plays reserve price I_t, both seller's revenue and highest bid revealed to him/her
- Seller/Player in a position to observe all revenues for prices $j \geq I_t$
- $\alpha(G) = 1$: $R_T = O\left(\ln(TN)\sqrt{T}\right)$ (expert game up to logs)

Outline

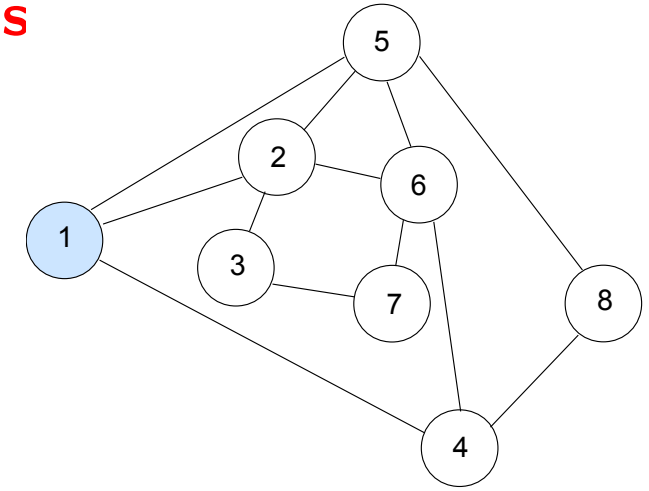
- Nonstochastic bandit game (vanilla + some extensions)
- (Delayed) cooperation among nonstochastic bandit agents in distributed environments
- Learning nonparametric policies in a nonstochastic setting

Cooperation among bandit agents



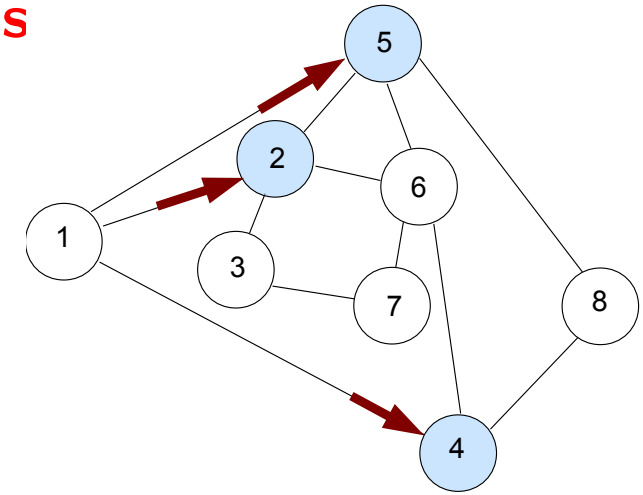
- K agents on the nodes of given communication network $G = (V, E)$
- Agents cooperate to solve same problem (same set of N actions)
- Each agent only knows its neighborhood (neither K nor G are known)
- But each agent runs bandit alg. (e.g. same instance of Exp3)
- At the end of each round each agent observes its own loss and sends it to its neighbors
- Each agent sends its own messages, but also forwards to its neighbors received messages which are not too old
- G is then synced multihop and broadcast communication network

Cooperation among bandit agents



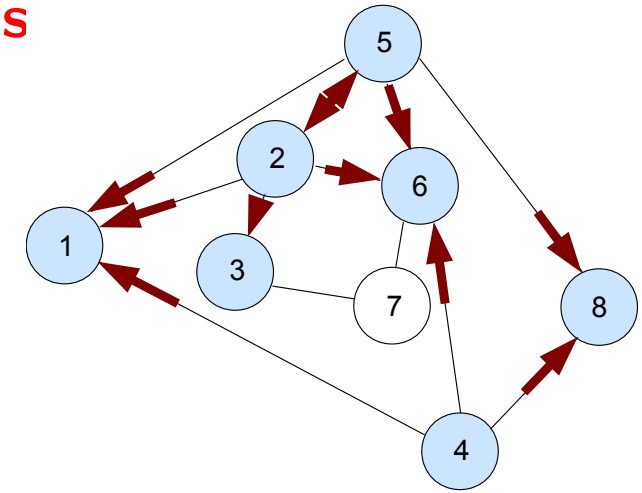
- K agents on the nodes of given communication network $G = (V, E)$
- Agents cooperate to solve same problem (same set of N actions)
- Each agent only knows its neighborhood (neither K nor G are known)
- But each agent runs bandit alg. (e.g. same instance of Exp3)
- At the end of each round each agent observes its own loss and sends it to its neighbors
- Each agent sends its own messages, but also forwards to its neighbors received messages which are not too old
- G is then synced multihop and broadcast communication network

Cooperation among bandit agents



- K agents on the nodes of given communication network $G = (V, E)$
- Agents cooperate to solve same problem (same set of N actions)
- Each agent only knows its neighborhood (neither K nor G are known)
- But each agent runs bandit alg. (e.g. same instance of Exp3)
- At the end of each round each agent observes its own loss and sends it to its neighbors
- Each agent sends its own messages, but also forwards to its neighbors received messages which are not too old
- G is then synced multihop and broadcast communication network

Cooperation among bandit agents



- K agents on the nodes of given communication network $G = (V, E)$
- Agents cooperate to solve same problem (same set of N actions)
- Each agent only knows its neighborhood (neither K nor G are known)
- But each agent runs bandit alg. (e.g. same instance of Exp3)
- At the end of each round each agent observes its own loss and sends it to its neighbors
- Each agent sends its own messages, but also forwards to its neighbors received messages which are not too old
- G is then synced multihop and broadcast communication network

Some related work

- Cooperative nonstochastic bandits without delays [Awerbuch and Kleinberg, '08]
- Cooperative stochastic bandits [Szorenyi et al., '13, '16, Kumar et al. '16]
- Stochastic bandits that compete for shared resources (cognitive radio networks) [...]
- Distributed gradient descent [Zinkevich et al. '09, Agarwal and Duchi '11, McMahan and Streeter '14, Duchi et al. '15 . . .]

Cooperative agents in Distributed Environment/1: Learning Protocol/1

For $t = 1, 2 \ldots$ and each agent v :

1. v picks action $I_t(v) \sim \boldsymbol{p}_t(v)$ and incurs (and observes) loss $\ell_t(I_t(v))$ [same loss vector for all agents]
2. v sends to its neighbors message
$$m_t(v) = \langle t, v, I_t(v), \ell_t(I_t(v), \boldsymbol{p}_t(v)) \rangle$$
3. v receives from its neighbors a variable number of messages $m_{t-s}(v')$ and forwards only those such that $s < d$ (i.e. not older than d)

Delay (and communication control) :

- Each agent receives message from other agents with delay equal to shortest-path distance between the two
- A message sent by some agent v at time t will be received by all agents whose shortest-path distance from v is at most d
- Communication control mechanism (time-to-live d) is exogenous parameter of the learning problem

Cooperative agents in Distributed Environment/1: Learning Protocol/2

Average welfare regret :

$$R_T^{\text{coop}} = \frac{1}{K} \sum_{v \in V} \mathbb{E} \left[\sum_{t=1}^{T} \ell_t(I_t(v)) \right] - \min_{i \in A} \sum_{t=1}^{T} \ell_t(i)$$

Remarks :

- In case of no cooperation ($G = (V, \emptyset)$) then

$$R_T^{\text{coop}} \leq \sqrt{T N \ln N}$$

- But relying on other agents' plays may improve quality of estimators $\widehat{\ell}_t(i)$
- Delay parameter d trades off quality and quantity of shared info (beyond controlling message complexity)

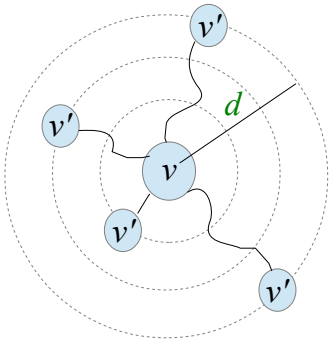
Cooperative agents in Distributed Environment/2

Cooperative (delayed) importance sampling estimate :

$$\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_{t-d}(i)}{\Pr_{t-d}(\ell_{t-d}(i) \text{ is observed by } v \text{ at time } t)} & \text{if } \ell_{t-d}(i) \text{ is observed by } v \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

- Agent v :

  As d grows s/he observes more and more losses but from further and further rounds



- Induced communication graph G_d (d-th power of G): connect any two nodes whose shortest-path dist. in G is $\leq d$

Cooperative agents in Distributed Environment/3: Bounds/1

Average welfare regret bounds :

$$R_T^{\text{coop}} \leq \sqrt{\left(d + 1 + \frac{N}{K}\alpha(G_d)\right) T \ln N}$$

Independence number of d-th power of G

Few examples/1:

- If $K \approx N$ and $d = \text{diam}(G)$

$$R_T^{\text{coop}} \leq \sqrt{(d+1)\, T \ln N}$$

  - Each agent sees losses of N random actions with delay d
  - reminiscent of d-delayed full info minimax rate of $\sqrt{(d+1)T \ln N}$
  - convenient only if $\text{diam}(G)$ small

Cooperative agents in Distributed Environment/3: Bounds/2

Average welfare regret bounds :

$$R_T^{\text{coop}} \leq \sqrt{\left(d + 1 + \frac{N}{K}\alpha(G_d)\right) T \ln N}$$

Independence number of d-th power of G

Few examples/2:

- If G is arbitrary connected graph then $\alpha(G_d) \leq \frac{2K}{d}$ and $d \approx \sqrt{N}$ gives

$$R_T^{\text{coop}} \leq N^{1/4}\sqrt{T \ln N}$$

  - better than single agent minimax $\sqrt{TN}$

Cooperative agents in Distributed Environment/4: Individual delays and ttl/1

In practice agents:

- May use personalized param's if they know network topology:
  - $d(v)$ = willingness of v to receive and $ttl(v)$ = willingness of v to send
  - v has small individual delay $d(v)$ if located in dense area and small individual $ttl(v)$ if central rather than peripheral (so as not to waste msgs)
- Need not know param's of other agents

Exogenous communication structure:

- Individual param's $\mathcal{P} = \{d(v), ttl(v)\}_{v \in V}$
- We do not assume agents know param's of others
- v uses msgs from v' if their shortest-path dist $\leq \min\{d(v), ttl(v')\}$
- Induced communication graph: $G_{\mathcal{P}} = G + \mathcal{P}$ is now directed

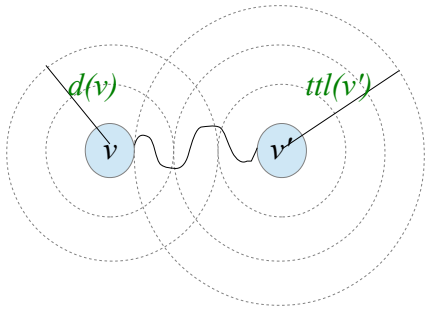
Cooperative agents in Distributed Environment/4: Individual delays and ttl/2

Cooperative (delayed) importance sampling estimate :

$$\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_{t-d(v)}(i)}{\Pr_{t-d(v)}(\ell_{t-d(v)}(i) \text{ is observed by } v \text{ at time } t)} & \text{if } \ell_{t-d(v)}(i) \text{ is observed by } v \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

v receives losses incurred by v' iff

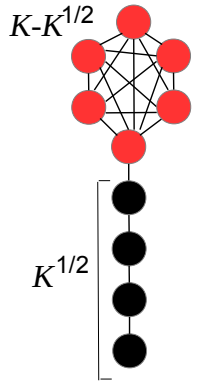
$$\text{shortest-path-dist}_G(v, v') \leq \min\{d(v), ttl(v')\}$$



Cooperative agents in Distributed Environment/5: Bounds

Average welfare regret bounds :

$$R_T^{\text{coop}} \leq \sqrt{\left(\underbrace{\bar{d} + 1 + \frac{N}{K}\alpha(G_{\mathcal{P}}) \ln(TKN)}_{\text{Independence number of induced commun. graph}} \right) T \ln N}$$



Compare main terms in regret bound when $K \approx N$:

- common delay param. (earlier):

  $d + \alpha(G_d)$

  $d = K^{1/4} \implies \widetilde{O}(K^{1/4})$

- individual param's:

  $\bar{d} + \alpha(G_{\mathcal{P}})$

  $\bar{d}$ = avg. delay over agents
  $d(v) = 1$ for red, $d(v) = \sqrt{K}$ for black $\implies \widetilde{O}(1)$

Outline

- Nonstochastic bandit game (vanilla + some extensions)
- (Delayed) cooperation among nonstochastic bandit agents in distributed environments
- Learning nonparametric policies in a nonstochastic setting

Learning against Lipschitz policies/1

Ingredients:

- Context (metric) space $\mathcal{X}$ (e.g., $\mathcal{X} = \mathbf{R}^d$)
- Action (metric) space $\mathcal{Y}$ (e.g., $\mathcal{Y} = [0, 1]$)
- Class of Lipschitz (and bounded) policies $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$
- Lipschitz loss functions $\ell_t : \mathcal{Y} \to [0, 1]$ (our "structure")

Learning protocol(s):

- Opponent picks context $x_t \in \mathcal{X}$
- Player observes x_t and picks action $\widehat{y}_t \in \mathcal{Y}$,
- Player pays loss $\ell_t(\widehat{y}_t)$
- Player observes:
  - $\ell_t(\widehat{y}_t)$ only [bandit info: contextual bandit]
  - $\ell_t(y)$ $\forall y \geq y_t$ [one-sided full info: contextual one-sided expert]

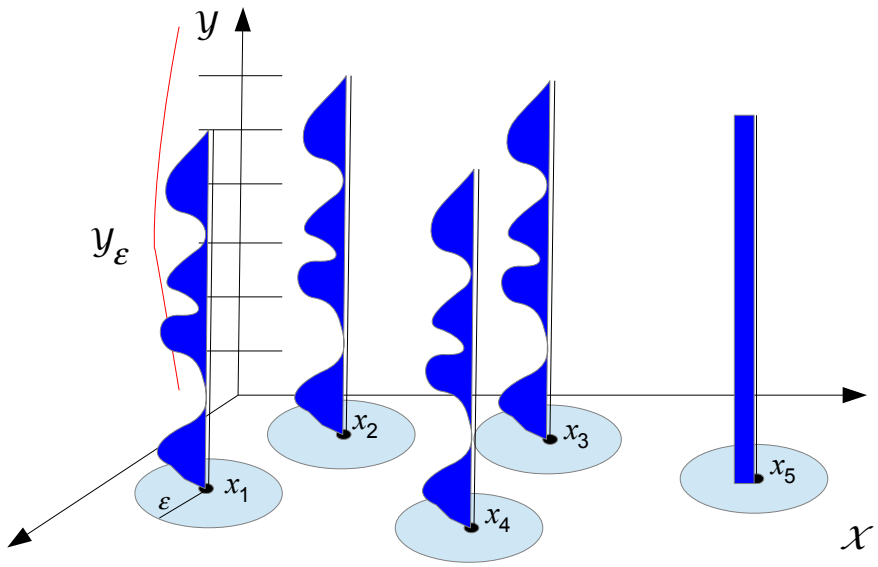
Learning against Lipschitz policies/2

Regret of Player for T rounds w.r.t. $\mathcal{F}$:

$$R_T(\mathcal{F}) = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\widehat{y}_t)\right] - \min_{f\in\mathcal{F}} \sum_{t=1}^{T} \ell_t(f(x_t))$$

Want : $R_T = o(T)$ as T grows large ("no regret")

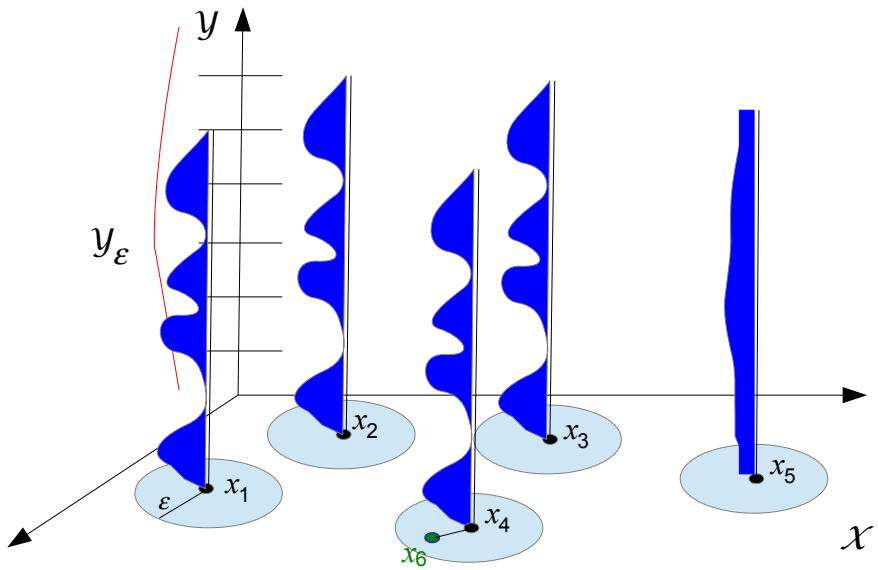
Contextual bandit game: a folk algorithm [K04,S14,...]



Each newly created ball centered in x_t hosts instance of EXP3 over discretized action space $\mathcal{Y}_\epsilon$

- If x_t outside any ball so far, create new ball centered on x_t
- Determine active EXP3 instance by past center x_s closest to x_t
- Draw action $\widehat{y}_t$ according to active EXP3 and update its weights only

Contextual bandit game: a folk algorithm [K04,S14,...]



Each newly created ball centered in x_t hosts instance of EXP3 over discretized action space $\mathcal{Y}_\epsilon$

- If x_t outside any ball so far, create new ball centered on x_t
- Determine active EXP3 instance by past center x_s closest to x_t
- Draw action $\widehat{y}_t$ according to active EXP3 and update its weights only

Contextual bandit game: regret bounds [K04,S14,...]

- $d =$ metric dimension of $\mathcal{X}$
- $1 =$ metric dimension of $\mathcal{Y}$

Then:

- Lipschitz: $\tilde{O}(T^{\frac{d+2}{d+3}})$ [folk alg]
- Convex: $\tilde{O}(T^{\frac{d+1}{d+2}})$ [folk alg + BEL16]
- Lower bound for $\underbrace{d = 0}_{no\ \ context}$: $\Omega(T^{\frac{2}{3}})$ [B+11]

In all cases:

- Exploit finite coverability of $\mathcal{X}$ and $\mathcal{Y}$
- Set radius ϵ appropriately

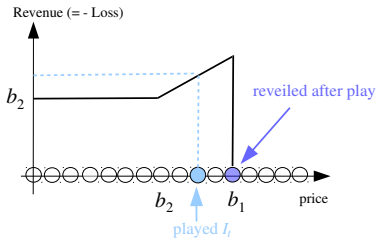
Contextual one-sided expert game/1 [CB+17]

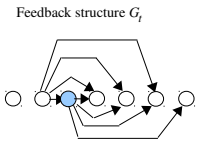
General but suboptimal approach:
Discretize set $\mathcal{F}$ of Lipschitz functions
Sup norm approximation at precision ϵ with $\approx 2^{(1/\epsilon)^d}$ functions, and run independent algs. at each ball

Why suboptimal?
Algs at each ball treated as uncorrelated: structure not exploited for large $\mathcal{F}$

Remark: In one-sided expert game, using Exp3-IX-like combined with folk alg on ϵ-balls over $\mathcal{X}$ yields regret

$$R_T(\mathcal{F}) \lesssim \sqrt{T \ln N_\epsilon} + T\epsilon \quad \text{if the } \ell_t \text{ are (semi-)Lipschitz}$$
$$\lesssim T^{\frac{d+1}{d+2}} \quad \text{when optimizing on } \epsilon$$

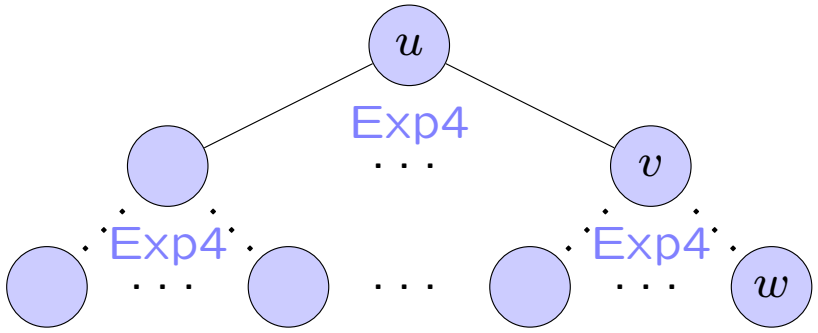




Contextual one-sided expert game/2: Chaining/1 [CB+17]

Ideas of the algorithm:
Hierarchical covering of $\mathcal{F}$ = tree whose nodes are functions in $\mathcal{F}$

- The nodes at each depth m define a (2^{-m})-covering of $\mathcal{F}$
- Any function $f^* \in \mathcal{F}$ is represented by unique path/chain in the tree
- Run an instance of Exp4 (adapted to one-sided expert feedback) on each node of tree
- Instance A_f at node f uses the predictions of child instances as expert advice



Level m $\rightsquigarrow 2^{-m}$ covering of $\mathcal{F}$

Level $m+1$ $\rightsquigarrow 2^{-(m+1)}$ covering

Level M (leaves) $\rightsquigarrow 2^{-M}$ covering

Contextual one-sided expert game/2: Chaining/2 [CB+17]

Key issues (Lipschitz losses):

- Small local ranges: losses associated with neighboring nodes are close
- Local version of Exp4 scaling with loss range: possible because of richer feedback
- Regret:

$$R_T(\mathcal{F}) \lesssim \gamma T + \int_\gamma^1 \sqrt{\frac{T}{\gamma} \ln N(\mathcal{F}, \epsilon)} d\epsilon \qquad \forall \gamma > 0$$

$$\lesssim T^{\frac{d}{d+1}} \quad \text{(when } \mathcal{F} \text{ are Lipschitz on } [0,1]^d\text{)}$$

- Improvements when $\mathcal{F}$ = Lipschitz functions on $[0,1]^d$, time efficient algorithm (wavelet-based approx.):
  - Improved regret rate $T^{\frac{d-1/3}{d+2/3}}$
  - Running time per round: $\approx T^\alpha$, $\alpha < 2$

Learning against Lipschitz policies

Bounds abound !

Exponents of T:

- Contextual bandits:
  - General Lipschitz: $\frac{d+2}{d+3}$
  - Convex: $\frac{d+1}{d+2}$
- Contextual one-sided:
  - General Lipschitz: $\frac{d}{d+1}$
  - Semi-Lipschitz: $\frac{d+1}{d+2}$
  - Rectangular context space and general Lipschitz ($d \geq 1$): $\frac{d-1/3}{d+2/3}$
- Contextual experts ($d \geq 2$): $\frac{d-1}{d}$ (tight) [RST15]

Conclusions and open questions

Recent activity in nonstochastic bandits with structure (over action space)

Many open questions related to the various problems presented, e.g.,

In cooperative agent case:

- Simultaneous regret bounds that hold for each agent individually
- Slightly different problems across agents (tradeoff cooperation-diversity)
- Bounded/unreliable communication
- Privacy-preserving
- Cooperative-competitive

In learning with Lipschitz policies

- Tighter upper bounds
- Lower bounds
- Efficient algorithms