How to Explore to Maximize Future Return

Csaba Szepesvári
University of Alberta
Department of Computing Science

Based on joint work with:
Yasin-Abbasi Yadkori and Dávid Pál
and the RLAI group
Some challenges
Some challenges

• Autonomous cars
Some challenges

- Autonomous cars
- Voice-user interface systems
Some challenges

- Autonomous cars
- Voice-user interface systems
- Dynamic treatment regimes
Some challenges

• Autonomous cars
• Voice-user interface systems
• Dynamic treatment regimes
• Intelligent tutoring
How?
Big Data
Machine Learning
What is missing from here?
What is missing from here?

Need to make decisions!
RL to the Rescue
RL to the Rescue

Goal: Maximize the total reward collected
What’s coming

• What are the unique challenges?

• How are we going to address them?

• Contents:
  
  • (No) Big Data in Decision Making

  • Learning Interactively: Opportunities, challenges, techniques

  • Scaling up Bandits
(No) Big Data in Decision Making
A Swimming Lesson

[Humble Pie] [Swimmer] [Cornucopia]

Key Insight from Theory of "Efficient RL"

[Strehl-Littman, 2008]

Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,’08]
A Swimming Lesson

Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,’08]
A Swimming Lesson

Slide graphics courtesy of Ben van Roy. Problem due to [Strehl-Littman,’08]
A Swimming Lesson

Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman, ‘08]
A Swimming Lesson

• **Reckless** data collection: Choose the actions *uniformly at random*!
A Swimming Lesson

- **Reckless** data collection: Choose the actions *uniformly at random*!
- **How much data** do we need to collect before we see the bounty for the first time, starting from the middle?

Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,’08]
A Swimming Lesson

- **Reckless** data collection: Choose the actions *uniformly at random!*
- **How much data** do we need to collect before we see the bounty for the first time, starting from the middle?
- How does this depend on the number of states?

Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,'08]
Time before bounty is found
Time before bounty is found

2 million steps for 19 states!
Time before bounty is found
Time before bounty is found

- Hitting time for random policy: \( \Theta(2^n) \)
Time before bounty is found

- Hitting time for random policy: $\Theta(2^n)$
- Hitting time for "swimming policy": $\Theta(n)$
Time before bounty is found

- Hitting time for random policy:
  \( \Theta(2^n) \)

- Hitting time for “swimming policy”:
  \( \Theta(n) \)

- Exponential gap on a very simple example! ..could be much worse on a real problem
Time before bounty is found

- Hitting time for random policy: \( \Theta(2^n) \)
- Hitting time for “swimming policy”: \( \Theta(n) \)

- Exponential gap on a very simple example! ..could be much worse on a real problem
- How “big” is big enough?
Time before bounty is found

- Hitting time for random policy: $\Theta(2^n)$
- Hitting time for “swimming policy”: $\Theta(n)$

- Exponential gap on a very simple example! ..could be much worse on a real problem
- How “big” is big enough?
- Will we ever have enough data? Can we do better?
Changing the game..
Changing the game..

- Allow data to be collected by a policy we select
Changing the game..

• Allow data to be collected by a policy we select

• Can we design more efficient data collection policies?
Standard RL Approach
Standard RL Approach

• Repeat:
Standard RL Approach

• Repeat:
  • Learn a “good” policy
Standard RL Approach

• Repeat:
  • Learn a “good” policy
  • Add randomness to induce exploration
Standard RL Approach

• Repeat:
  • Learn a “good” policy
  • Add randomness to induce exploration
  • Collect more data (multiple episodes)
Standard RL Approach

• Repeat:
  • Learn a “good” policy
  • Add randomness to induce exploration
  • Collect more data (multiple episodes)
  • “epsilon-greedy”, “Boltzmann exploration”
Standard RL Approach

- Repeat:
  - Learn a “good” policy
  - Add randomness to induce exploration
  - Collect more data (multiple episodes)
  - “epsilon-greedy”, “Boltzmann exploration”
  - “Dithering”
What happens with dithering in RiverSwim?
What happens with dithering in RiverSwim?

What is the policy learned initially?
How long do we need to wait until the bounty is first collected?
What happens with dithering in RiverSwim?

Dithering is NOT sufficient. Need smart exploration methods.

What is the policy learned initially?
How long do we need to wait until the bounty is first collected?
How do we evaluate a data collection strategy?
How do we evaluate a data collection strategy?

• How much data is needed to find a good policy?
How do we evaluate a data collection strategy?

- How much data is needed to find a good policy? ..reward collected/lost during data collection does not matter: “pure exploration” problem
How do we evaluate a data collection strategy?

- How much data is needed to find a good policy? The reward collected/lost during data collection does not matter: "pure exploration" problem

- How much reward is incurred during data collection? "exploitation" problem
How do we evaluate a data collection strategy?

- How much data is needed to find a good policy? ..reward collected/lost during data collection does not matter: **“pure exploration” problem**

- How much reward is incurred during data collection? **“exploitation” problem**
  
  Must optimize *while* learning. Explore or exploit? 
  
  Metric: **Regret**.
Intelligent Interactive Learning
Optimism in the Face of Uncertainty

Lai and Robbins (1985), Burnetas and Katehakis (1996),
Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
Optimism in the Face of Uncertainty

Repeat:

Lai and Robbins (1985), Burnetas and Katehakis (1996), Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
Optimism in the Face of Uncertainty

Repeat:
1. Find the set $S$ of likely “worlds” given the observations so far

Lai and Robbins (1985), Burnetas and Katehakis (1996), Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
Optimism in the Face of Uncertainty

Repeat:
1. Find the set $S$ of likely “worlds” given the observations so far

Lai and Robbins (1985), Burnetas and Katehakis (1996), Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
Optimism in the Face of Uncertainty

Repeat:
1. Find the set $S$ of likely “worlds” given the observations so far
2. Find the world $W$ in $S$ with the maximum payoff:

$$W = \arg\max_{W \in S} J(W)$$

Lai and Robbins (1985), Burnetas and Katehakis (1996), Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
Repeat:
1. Find the set \( S \) of likely “worlds” given the observations so far
2. Find the world \( W \) in \( S \) with the maximum payoff:
\[
W = \arg\max_{W \in S} J(W)
\]

Lai and Robbins (1985), Burnetas and Katehakis (1996), Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
Optimism in the Face of Uncertainty

Repeat:
1. Find the set $S$ of likely “worlds” given the observations so far
2. Find the world $W$ in $S$ with the maximum payoff:
   $$W = \arg\max_{W \in S} J(W)$$
3. Find the optimal policy for this world:
   $$\pi = \arg\max_{\xi} J(W, \xi)$$

Lai and Robbins (1985), Burnetas and Katehakis (1996), Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
Repeat:
1. Find the set $S$ of likely “worlds” given the observations so far
2. Find the world $W$ in $S$ with the maximum payoff:
   \[ W = \arg\max_{W \in S} J(W) \]
3. Find the optimal policy for this world:
   \[ \pi = \arg\max_{\xi} J(W, \xi) \]

Lai and Robbins (1985), Burnetas and Katehakis (1996), Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
Optimism in the Face of Uncertainty

Repeat:
1. Find the set \( S \) of likely “worlds” given the observations so far
2. Find the world \( W \) in \( S \) with the maximum payoff:
   \[
   W = \arg \max_{W \in S} J(W)
   \]
3. Find the optimal policy for this world:
   \[
   \pi = \arg \max_{\xi} J(W, \xi)
   \]

Lai and Robbins (1985), Burnetas and Katehakis (1996), Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
Optimism in the Face of Uncertainty

Repeat:
1. Find the set $S$ of likely “worlds” given the observations so far
2. Find the world $W$ in $S$ with the maximum payoff:
   \[ W = \arg\max_{W \in S} J(W) \]
3. Find the optimal policy for this world:
   \[ \pi = \arg\max_{\xi} J(W, \xi) \]

Lai and Robbins (1985), Burnetas and Katehakis (1996), Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
Optimism in the Face of Uncertainty

Repeat:
1. Find the set $S$ of likely “worlds” given the observations so far
2. Find the world $W$ in $S$ with the maximum payoff:
   $$W = \arg\max_{W \in S} J(W)$$
3. Find the optimal policy for this world:
   $$\pi = \arg\max_{\xi} J(W, \xi)$$
4. Use this policy until $S$ significantly shrinks

Lai and Robbins (1985), Burnetas and Katehakis (1996), Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others
How good is OFU?
How good is OFU?

$S$ states, $A$ actions, rewards in $[0,1]$. 

[Jaksch-Ortner-Auer,’10]
How good is OFU?

$S$ states, $A$ actions, rewards in $[0,1]$.

OFU for finite problems: UCRL2
How good is OFU?

$S$ states, $A$ actions, rewards in $[0,1]$. 

**Definition:** Diameter := maximum of best travel times between pairs of states. River swim: $D = S$

OFU for finite problems: UCRL2
How good is OFU?

$S$ states, $A$ actions, rewards in $[0,1]$.

**Definition:** Diameter := maximum of best travel times between pairs of states. River swim: $D = S$

- **Theorem:** The regret of an OFU learner satisfies

  $$R_T = \tilde{O}(DS\sqrt{AT})$$

**OFU for finite problems:** UCRL2
How good is OFU?

$S$ states, $A$ actions, rewards in $[0,1]$.

**Definition:** Diameter := maximum of best travel times between pairs of states. River swim: $D = S$

- **Theorem:** The regret of an OFU learner satisfies
  $$R_T = \tilde{O}(DS\sqrt{AT})$$

- **Theorem:** For any algorithm,
  $$R_T = \Omega(\sqrt{DSAT})$$

**OFU for finite problems: UCRL2**
Posterior Sampling
Reinforcement Learning

[Thompson, 1933(!), Strens ’00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:

[Thompson, 1933(!), Strens ’00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
• Prior over the worlds

[Thompson, 1933(!), Strens ’00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
• Prior over the worlds
• Likelihood model

[Thompson, 1933(!), Strens ’00]
A Bayesian start:
- Prior over the worlds
- Likelihood model
- Posterior: \( p(W|D) \propto p_W(W)p(D|W) \)

[Thompson, 1933(!), Strens ’00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
• Prior over the worlds
• Likelihood model
• Posterior: $p(W|D) \propto p_W(W)p(D|W)$

Repeat:

[Thompson, 1933(!), Strens ’00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

Repeat:
1. Sample a world $W$ from the posterior:
   $$W \sim P(W = \cdot | D)$$

[Thompson, 1933(!), Strens ’00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
• Prior over the worlds
• Likelihood model
• Posterior: \( p(W|D) \propto p_W(W)p(D|W) \)

Repeat:
1. Sample a world \( W \) from the posterior:

\[
W \sim P(W = \cdot | D)
\]

[Thompson, 1933(!), Strens ’00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
- Prior over the worlds
- Likelihood model
- Posterior: \( p(W|D) \propto p_W(W)p(D|W) \)

Repeat:
1. Sample a world \( W \) from the posterior:

\[
W \sim P(W = \cdot | D)
\]

[Thompson, 1933(!), Strens ’00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
• Prior over the worlds
• Likelihood model
• Posterior: \( p(W|D) \propto p_W(W)p(D|W) \)

Repeat:
1. Sample a world \( W \) from the posterior:
\[
W \sim P(W = \cdot|D)
\]

[Thompson, 1933(!), Strens '00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
• Prior over the worlds
• Likelihood model
• Posterior: \( p(W|D) \propto p_W(W)p(D|W) \)

Repeat:
1. Sample a world \( W \) from the posterior:

\[
W \sim P(W = \cdot|D)
\]

[Thompson, 1933(!), Strens ’00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
• Prior over the worlds
• Likelihood model
• Posterior: \( p(W|D) \propto p_W(W)p(D|W) \)

Repeat:
1. Sample a world \( W \) from the posterior:
   \[
   W \sim P(W = \cdot | D)
   \]
2. Find the optimal policy for this world:
   \[
   \pi = \arg\max_{\xi} J(W, \xi)
   \]

[Thompson, 1933(!), Strens '00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
- Prior over the worlds
- Likelihood model
- Posterior: \[ p(W|D) \propto p_W(W)p(D|W) \]

Repeat:
1. Sample a world \( W \) from the posterior:
   \[ W \sim P(W=\cdot|D) \]
2. Find the optimal policy for this world:
   \[ \pi = \arg\max_{\xi} J(W, \xi) \]

[Thompson, 1933(!), Strens '00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
• Prior over the worlds
• Likelihood model
• Posterior: \( p(W|D) \propto p_W(W)p(D|W) \)

Repeat:
1. Sample a world \( W \) from the posterior:
   \[ W \sim P(W = \cdot|D) \]
2. Find the optimal policy for this world:
   \[ \pi = \arg\max_{\xi} J(W, \xi) \]

[Thompson, 1933(!), Strens '00]
Posterior Sampling
Reinforcement Learning

A Bayesian start:
• Prior over the worlds
• Likelihood model
• Posterior: $p(W|D) \propto p_W(W)p(D|W)$

Repeat:
1. Sample a world $W$ from the posterior:
   $$W \sim P(W = \cdot | D)$$
2. Find the optimal policy for this world:
   $$\pi = \arg\max_{\xi} J(W, \xi)$$
3. Use this policy a “little while”

[Thompson, 1933(!), Strens ’00]
Beating a near-optimal algorithm

Table 1: Total regret in simulation. PSRL outperforms UCRL2 over different environments.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Random MDP</th>
<th>Random MDP</th>
<th>RiverSwim</th>
<th>RiverSwim</th>
</tr>
</thead>
<tbody>
<tr>
<td>·-episodes</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>·-horizon</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

Placement of tables and figures in the presentation is not intrinsic to the content.
Scaling up
Scaling up

- **Large** state-action spaces:
  need to **generalize** across states and actions
Scaling up

- **Large** state-action spaces: need to **generalize** across states and actions

- Model based approach:
Scaling up

- **Large** state-action spaces: need to **generalize** across states and actions

- Model based approach:

  \[ x_{t+1} = f(x_t, a_t, \theta_\star, z_{t+1}) \]
Scaling up

- **Large** state-action spaces: need to **generalize** across states and actions

- Model based approach:

\[ x_{t+1} = f(x_t, a_t, \theta_*, z_{t+1}) \]

next state
Scaling up

- **Large** state-action spaces: need to **generalize** across states and actions

- Model based approach:

\[ x_{t+1} = f(x_t, a_t, \theta_*, z_{t+1}) \]

- next state
- current state
Scaling up

• **Large** state-action spaces: need to **generalize** across states and actions

• Model based approach:

\[
x_{t+1} = f(x_t, a_t, \theta_*, z_{t+1})
\]

next state \hspace{1cm} current state \hspace{1cm} action
Scaling up

- **Large** state-action spaces: need to **generalize** across states and actions

- Model based approach:

  \[ x_{t+1} = f(x_t, a_t, \theta^*, z_{t+1}) \]

  \[ \text{next state} \quad \text{current state} \quad \text{unknown parameter} \quad \text{action} \]
Scaling up

- **Large** state-action spaces: need to **generalize** across states and actions

- Model based approach:

\[ x_{t+1} = f(x_t, a_t, \theta_*, z_{t+1}) \]

- next state
- current state
- unknown parameter
- action
- noise
Linear Quadratic Regulation
Linear Quadratic Regulation

\[ x_{t+1} = Ax_t + Ba_t + z_{t+1} \]

\[ c_{t+1} = x_t^\top Q x_t + a_t^\top R a_t \]
Linear Quadratic Regulation

\[ x_{t+1} = Ax_t + Ba_t + z_{t+1} \]
\[ c_{t+1} = x_t^\top Q x_t + a_t^\top R a_t \]

\( \theta_* = (A, B) \)
is unknown
Linear Quadratic Regulation

\[
x_{t+1} = Ax_t + Ba_t + z_{t+1}
\]
\[
c_{t+1} = x_t^\top Q x_t + a_t^\top Ra_t
\]

\[\theta_* = (A, B)\]
is unknown

- **Theorem [Abbasi-Sz 2011]**: For reachable and controllable systems, the regret of OFU satisfies
Linear Quadratic Regulation

\[ x_{t+1} = Ax_t + Ba_t + z_{t+1} \]
\[ c_{t+1} = x_t^\top Q x_t + a_t^\top Ra_t \]
\[ \theta_* = (A, B) \]
is unknown

- **Theorem [Abbasi-Sz 2011]**: For reachable and controllable systems, the regret of OFU satisfies

\[ R_T = \tilde{O}(\sqrt{T}) \]
Linear Quadratic Regulation

\[ x_{t+1} = Ax_t + Ba_t + z_{t+1} \]
\[ c_{t+1} = x_t^\top Q x_t + a_t^\top Ra_t \]
\[ \theta_\star = (A, B) \]
is unknown

- **Theorem [Abbasi-Sz 2011]**: For reachable and controllable systems, the regret of OFU satisfies

\[ R_T = \tilde{O}(\sqrt{T}) \]

- Key idea: Estimate the unknown parameter using l2 regularized least-squares, develop tight confidence sets
Nonlinear systems?
Nonlinear systems?

• Smoothness:

\[ y = f(x, a, \theta, z), y' = f(x, a, \theta', z) \]

\[ \Rightarrow \]

\[ \mathbb{E} [\|y - y'\|] \leq \|\theta - \theta'\|_{M(x,a)} \]
Nonlinear systems?

• Smoothness:

\[ y = f(x, a, \theta, z), y' = f(x, a, \theta', z) \]

\[ \Rightarrow \]

\[ \mathbb{E} [\|y - y'\|] \leq \|\theta - \theta'\| \cdot M(x, a) \]

• **Theorem [Abbasi-Sz]**: For smooth, “bounded” systems, if the posterior is “concentrating”, the Bayes regret of PSRL is bounded by

\[ R_T = \tilde{O}(\sqrt{T}) \]
Nonlinear systems?

• Smoothness:

\[ y = f(x, a, \theta, z), \ y' = f(x, a, \theta', z) \]

\[ \Rightarrow \]

\[ \mathbb{E} [\|y - y'\|] \leq \|\theta - \theta'\|_M(x, a) \]

• **Theorem [Abbasi-Sz]**: For smooth, “bounded” systems, if the posterior is “concentrating”, the Bayes regret of PSRL is bounded by

\[ R_T = \tilde{O}(\sqrt{T}) \]

• Key idea: Use \( M(x, a) \) to measure information.
Web Server Control

[Graph showing CPU load with blue and red lines]
Web Server Control

- Control variables:
Web Server Control

- Control variables:
  - How long to keep alive a connection without traffic on it
Web Server Control

- Control variables:
  - How long to keep alive a connection without traffic on it
  - Maximum number of clients that can be served
Web Server Control

- Control variables:
  - How long to keep alive a connection without traffic on it
  - Maximum number of clients that can be served
- State variables:
Web Server Control

• Control variables:
  • How long to keep alive a connection without traffic on it
  • Maximum number of clients that can be served

• State variables:
  • Processor load relative to ideal processor load
Web Server Control

- Control variables:
  - How long to keep alive a connection without traffic on it
  - Maximum number of clients that can be served

- State variables:
  - Processor load relative to ideal processor load
  - Memory usage relative to ideal memory usage
Results

Explore then exploit

Q-learning w. dithering

OFULQ

OFULQ prefetch
OFULQ vs. PSRL

The frequency of policy switches is controlled by a parameter, which ultimately controls the computation time.

OFULQ = OFU on LQR

Lazy PSRL = PSRL that switches to new policy based on $M(x, a)$
Higher noise

Figure 3: Regret vs time for a web server control problem. (Top-left) regret of the OFULQ algorithm when $\gamma = 0.1$. (Top-right): regret of the Lazy PSRL algorithm when $\gamma = 0.1$. (Bottom-left) regret of the OFULQ algorithm when $\gamma = 1.0$. (Bottom-right): regret of the Lazy PSRL algorithm when $\gamma = 1.0$.

Figure 4: Regret of the Lazy PSRL algorithm with different priors. The prior is a zero mean Gaussian distribution with covariance matrix $\sigma^2 I$. The horizontal axis is $\gamma$.

8 Conclusions

We studied the problem of efficient computation of a nearly Bayes-optimal policy in average cost problems with smoothly parameterized, possibly nonlinear dynamics. In particular, we showed that lazy PSRL, when the same policy is used until the uncertainty in the posterior is sufficiently reduced leads to an algorithm whose computational cost depends mainly on the cost of solving the underlying classical (non-Bayesian) optimal control problem, and also on the cost of sampling from the posterior. Our analysis guarantees that the resulting method is indeed near Bayes optimal for a large class of systems. We also studied the effect of possibly exploding state and proposed a specific way to deal with this issue. As opposed to previous analysis of PSRL by Osband et al. [6], our analysis does not rely on a “UCB type” argument, but it hinges upon the concentration of the posterior, which we showed in two specific cases.

OFULQ = OFU on LQR
Lazy PSRL = PSRL that switches to new policy based on $M(x, a)$
Scaling up Bandits
Bandit Problems

Goal: maximize the total reward incurred
Linear Bandits
Linear Bandits
Linear Bandits

• Actions are elements of a vector space:
  \[ a \in \mathcal{A} \subseteq \mathbb{R}^d \]

• Reward: \[ R_t = \langle A_t, \theta_* \rangle + Z_t \]
Linear Bandits

• Actions are elements of a vector space:
  \[ a \in \mathcal{A} \subseteq \mathbb{R}^d \]

• Reward:
  \[ R_t = \langle A_t, \theta^* \rangle + Z_t \]

• L2 problem:
  \[ \|\theta\|_2 \leq 1, \|a\|_2 \leq 1 \]
Linear Bandits

- Actions are elements of a vector space:
  \[ a \in \mathcal{A} \subset \mathbb{R}^d \]

- Reward:
  \[ R_t = \langle A_t, \theta_\star \rangle + Z_t \]

- L2 problem:
  \[ \|\theta\|_2 \leq 1, \|a\|_2 \leq 1 \]

- **Theorem [Dani et al ’08]:** For subgaussian noise, OFU’s regret for the L2 problem is
  \[ R_T = \tilde{O}(d\sqrt{T}) \]
Confidence sets matter

Empirical Results: The Influence of Confidence Sets

- OFUL using the confidence set of \[AYPS11\] – “New bound”
- OFUL using the confidence set of \[DHK08\] – “Old bound”

- “New bound”: Abbasi-Pal-Sz’11
- “Old bound”: Dani-Hayes-Kakade ‘08
The challenge
The challenge

• Linear estimation problem
The challenge

• Linear estimation problem

The observations are \( R_1, A_1, \ldots, R_t, A_t \), where

\[ \ldots, R_t = \langle A_t, \theta_* \rangle + Z_t, \ldots \]
The challenge

• Linear estimation problem

  The observations are $R_1, A_1, \ldots, R_t, A_t$, where

  $$\ldots, R_t = \langle A_t, \theta_* \rangle + Z_t, \ldots$$

• Given $0 \leq \delta \leq 1$, find a set

  $$C_t = C_t(\delta, R_1, A_1, \ldots, R_t, A_t) \subset \mathbb{R}^d$$

  such that $\mathbb{P}(\theta_* \in C_t) \geq 1 - \delta$
The challenge

• Linear estimation problem
  The observations are $R_1, A_1, \ldots, R_t, A_t$, where
  $$\ldots, R_t = \langle A_t, \theta_* \rangle + Z_t, \ldots$$

• Given $0 \leq \delta \leq 1$, find a set
  $$C_t = C_t(\delta, R_1, A_1, \ldots, R_t, A_t) \subset \mathbb{R}^d$$
  such that $\mathbb{P}(\theta_* \in C_t) \geq 1 - \delta$

• The covariates, $A_t \in \mathbb{R}^d$, are chosen by a bandit algorithm, they are far from independent!

• We need a honest confidence set!
The challenge

• Linear estimation problem
  The observations are $R_1, A_1, \ldots, R_t, A_t$, where
  \[ \ldots, R_t = \langle A_t, \theta_* \rangle + Z_t, \ldots \]

• Given $0 \leq \delta \leq 1$, find a set
  \[ C_t = C_t(\delta, R_1, A_1, \ldots, R_t, A_t) \subset \mathbb{R}^d \]
  such that $\mathbb{P}(\theta_* \in C_t) \geq 1 - \delta$

• The covariates, $A_t \in \mathbb{R}^d$, are chosen by a bandit algorithm, they are far from independent!

• We need a honest confidence set!

• How to exploit sparsity of $\theta_*$?
A general solution
A general solution

• If we have a good predictor for an adversarial linear regression problem with small regret, the predictions $\hat{R}_1, \ldots, \hat{R}_t$ and the regret bound $B_t$ should give us a honest, tight confidence set.
A general solution

• If we have a good predictor for an adversarial linear regression problem with small regret, the predictions $\hat{R}_1, \ldots, \hat{R}_t$ and the regret bound $B_t$ should give us a honest, tight confidence set.

• **Theorem [Abbasi-Pal-Sz ’12]:** With probability $1 - \delta$, $\theta_\ast \in C_n$ holds for all $n \geq 1$, where

$$C_n = \left\{ \theta \in \mathbb{R}^d : \sum_{t=1}^{n} (\hat{R}_t - \langle A_t, \theta \rangle)^2 \leq 1 + 2B_n + 32\gamma^2 \ln \left( \frac{\gamma\sqrt{8} + \sqrt{1 + B_n}}{\delta} \right) \right\}$$
Sparse Linear Bandits
Sparse Linear Bandits

• **Theorem [YPSz ’12]:** The regret of OFUL enjoys

\[ R_T = \tilde{O}(\sqrt{dTB_T}) \]
Sparse Linear Bandits

• **Theorem [YPSz ’12]:** The regret of OFUL enjoys
  \[ R_T = \tilde{O}(\sqrt{dT B_T}) \]

• **Theorem [Gerchinowitz ’11]:** There exist an algorithm that achieves
  \[ B_T = O(p \log(dT)) \]

for linear regression with \( p \)-sparse parameter vectors belonging to the hypercube.
Sparse Linear Bandits

- **Theorem [YPSz ’12]:** The regret of OFUL enjoys
  \[ R_T = \tilde{O}(\sqrt{dTB_T}) \]

- **Theorem [Gerchinowitz ’11]:** There exist an algorithm that achieves
  \[ B_T = O(p \log(dT)) \]
  for linear regression with \( p \)-sparse parameter vectors belonging to the hypercube.

- **Corollary [YPSz ’12]:** For such problems,
  \[ R_T = \tilde{O}(\sqrt{dpT}) \]
Sparse Linear Bandits

- **Theorem [YPSz ’12]:** The regret of OFUL enjoys
  \[ R_T = \tilde{O}(\sqrt{dT B_T}) \]

- **Theorem [Gerchinowitz ’11]:** There exist an algorithm that achieves
  \[ B_T = O(p \log(dT)) \]
  for linear regression with \( p \)-sparse parameter vectors belonging to the hypercube.

- **Corollary [YPSz ’12]:** For such problems,
  \[ R_T = \tilde{O}(\sqrt{dpT}) \]

- **Theorem [YPSz’12]:** For all algorithms,
  \[ R_T = \Omega(\sqrt{dT}) \]
Why?
Why?

- Prediction problems (Candes, Tao 2006 and Bickel, Ritov, Tsybakov 2009), under RIP for LASSO:

\[ \left\| \hat{\theta}_n - \theta_\star \right\|_2 \sim \frac{\sqrt{p \log(d)}}{n} \]
Why?

• Prediction problems (Candes, Tao 2006 and Bickel, Ritov, Tsybakov 2009), under RIP for LASSO:

\[ \| \hat{\theta}_n - \theta_* \|_2 \sim \sqrt{p \log(d)/n} \]

• What is the difference?
Why?

• Prediction problems (Candes, Tao 2006 and Bickel, Ritov, Tsybakov 2009), under RIP for LASSO:
  \[ \| \hat{\theta}_n - \theta_* \|_2 \sim \sqrt{p \log(d) / n} \]

• What is the difference?
  
  • Good algorithms select good actions frequently
    \[ \implies \text{No RIP} \]
Why?

• Prediction problems (Candes, Tao 2006 and Bickel, Ritov, Tsybakov 2009), under RIP for LASSO:
  \[
  \left\| \hat{\theta}_n - \theta_\star \right\|_2 \sim \sqrt{p \log(d)/n}
  \]

• What is the difference?
  • Good algorithms select good actions frequently
    \[ \implies \text{No RIP} \]
  • Covariates are highly correlated
Still.. does it work?

Figure 4.18: Comparing the OFUL-EG and the OFUL-LS algorithms on synthetic data. The action set is $k = 200$ randomly generated vectors in $\{1, +1\}^{200}$. The parameter vector $\theta^*$ has only 10 non-zero elements, each being equal to $0.1$. The algorithm observes $\theta^*, a_t$ corrupted by a Gaussian noise drawn from $N(0, 0.1^2)$. The time horizon is $T = 1000$. We set the least-squares regularizer to $\lambda = 1$, and the EG learning rate to $\alpha = 1$. (a) The OFUL-LS algorithm outperforms the OFUL-EG algorithm (b) The OFUL-EG algorithm with the improved confidence width (4.20) outperforms the OFUL-LS algorithm (c) Improving the regret of the OFUL-EG algorithm with confidence width (4.21) (d) Experimenting with a problem with a smaller dimensionality and action set, $k = 100, d = 100$. 

$d = 100, p = 10$
Still.. does it work?

Yes, it does!

Figure 4.18: Comparing the OFUL-EG and the OFUL-LS algorithms on synthetic data. The action set is \( k = 200 \) randomly generated vectors in \( \{1, +1\}^{200} \). The parameter vector \( \theta^* \) has only 10 non-zero elements, each being equal to 0.1. The algorithm observes \( h, a_t \) corrupted by a Gaussian noise drawn from \( N(0, 0.1^2) \). The time horizon is \( T = 1000 \). We set the least-squares regularizer to \( \gamma = 1 \), and the EG learning rate to \( \gamma = 1 \). (a) The OFUL-LS algorithm outperforms the OFUL-EG algorithm (b) The OFUL-EG algorithm with the improved confidence width (4.20) outperforms the OFUL-LS algorithm (c) Improving the regret of the OFUL-EG algorithm with confidence width (4.21) (d) Experimenting with a problem with a smaller dimensionality and action set, \( k = 100, d = 100 \).
Summary
Summary

• To make impact, we need to solve decision problems
Summary

• To make impact, we need to solve **decision problems**
• This makes a **BIG** difference
Summary

• To make impact, we need to solve decision problems
• This makes a BIG difference
  • Passive data collection can be extremely ineffective: small “big data” !?
Summary

• To make impact, we need to solve decision problems
• This makes a BIG difference
  • Passive data collection can be extremely ineffective: small “big data” !?
• Need smart algorithms for learning and control
Summary

• To make impact, we need to solve **decision problems**
• This makes a **BIG** difference
  • Passive data collection can be extremely ineffective: **small “big data”** !?
• Need **smart algorithms for learning and control**
  • Planning to learn is critical
Summary

• To make impact, we need to solve decision problems
• This makes a BIG difference
  • Passive data collection can be extremely ineffective: small “big data”!?
• Need smart algorithms for learning and control
  • Planning to learn is critical
  • OFU or PSRL: Competing designs
Summary

• To make impact, we need to solve decision problems
• This makes a **BIG** difference
  • Passive data collection can be extremely ineffective: small “big data” !?
• Need **smart algorithms for learning and control**
  • Planning to learn is critical
  • OFU or PSRL: Competing designs
• Current research: **Scaling up!**
Summary

• To make impact, we need to solve **decision problems**
• This makes a **BIG** difference
  • Passive data collection can be extremely ineffective: small “big data” !?
• Need **smart algorithms for learning and control**
  • Planning to learn is critical
  • OFU or PSRL: Competing designs
• Current research: **Scaling up!**
  • **Sparsity..?**
Summary

• To make impact, we need to solve **decision problems**
• This makes a **BIG** difference
  • Passive data collection can be extremely ineffective: **small “big data”** !?
• Need **smart algorithms for learning and control**
  • Planning to learn is critical
  • OFU or PSRL: Competing designs
• Current research: **Scaling up!**
  • **Sparseness**..?

Significant computational, algorithmic and statistical challenges remain. Much to be done!!
Thanks for being here!
Questions?