The Hidden World of Bandits

Alessandro LAZARIC (*INRIA-Lille*)

Workshop on Sequential Learning and Applications, Toulouse

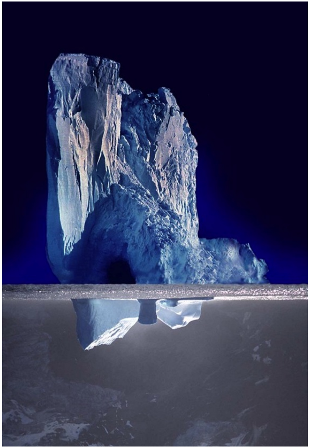
SequeL – INRIA Lille

November 9, 2015

Joint work with

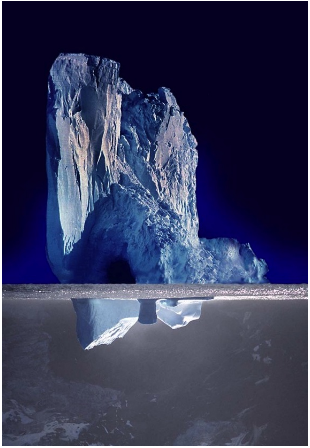
- Mohammad Azar (Northwestern University)
- Emma Brunskill (CMU)
- Anima Anandkumar (UCI)
- Kamyar Azizzade (UCI)

The Hidden World of Bandits



Many
bandit problems /
contexts / observations

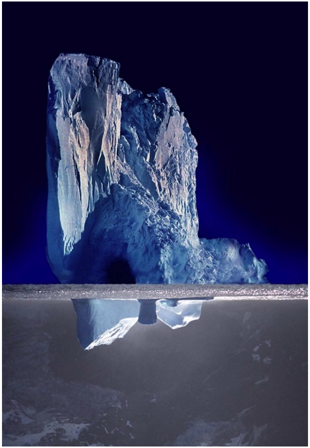
The Hidden World of Bandits



Many
bandit problems /
contexts / observations

Few hidden
structures

The Hidden World of Bandits



Many

bandit problems /
contexts / observations

Few hidden

structures

$\Rightarrow$ How do we learn *structures and solutions* at the same time?

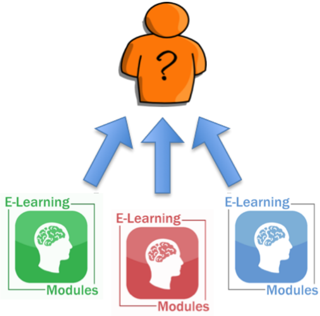
Outline

Sequential Transfer in MAB with Finite Set of Models

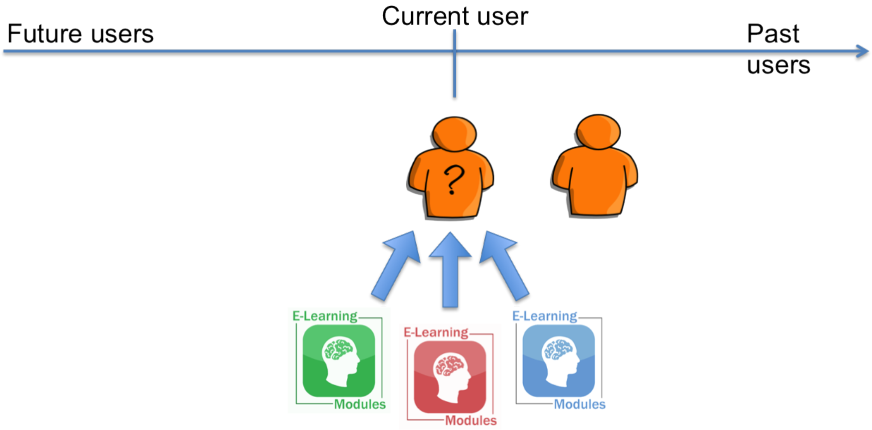
Learning in Partially Observable MDPs

Conclusions

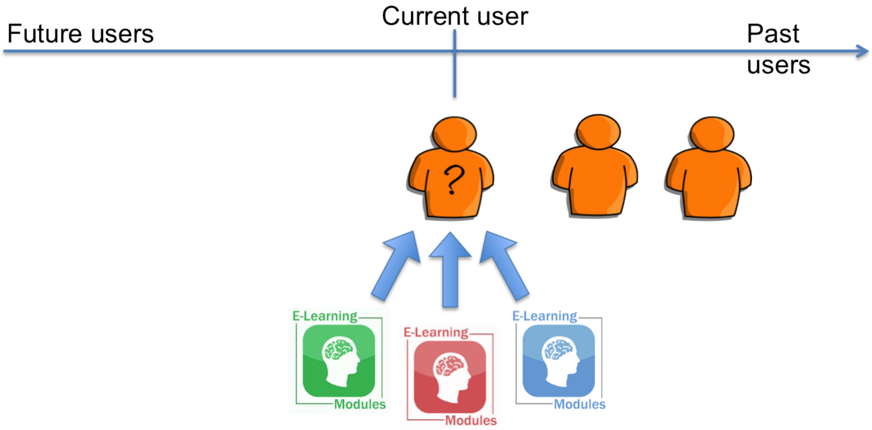
Multi-armed Bandit with *Hidden* Type



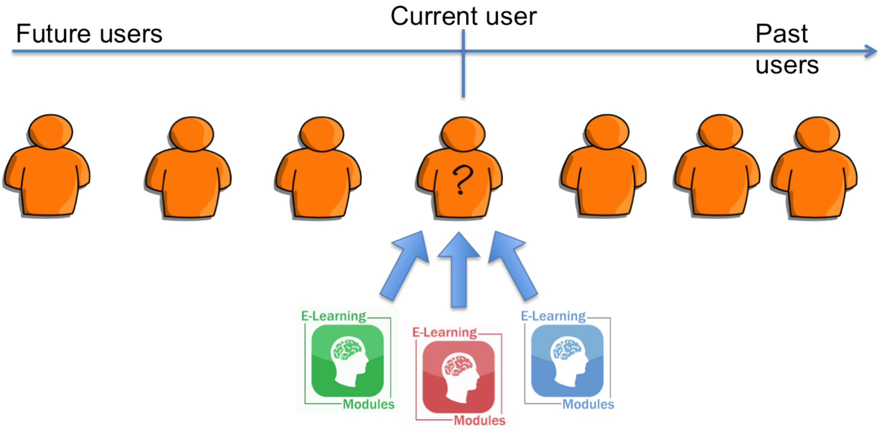
Multi-armed Bandit with *Hidden* Type



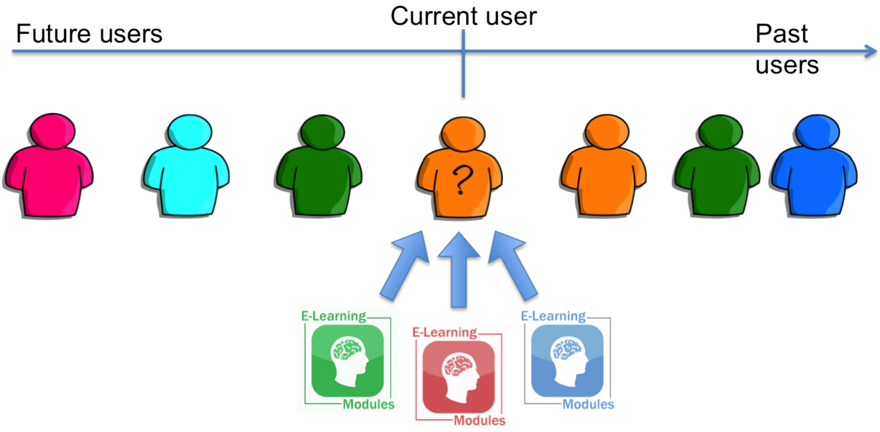
Multi-armed Bandit with *Hidden* Type



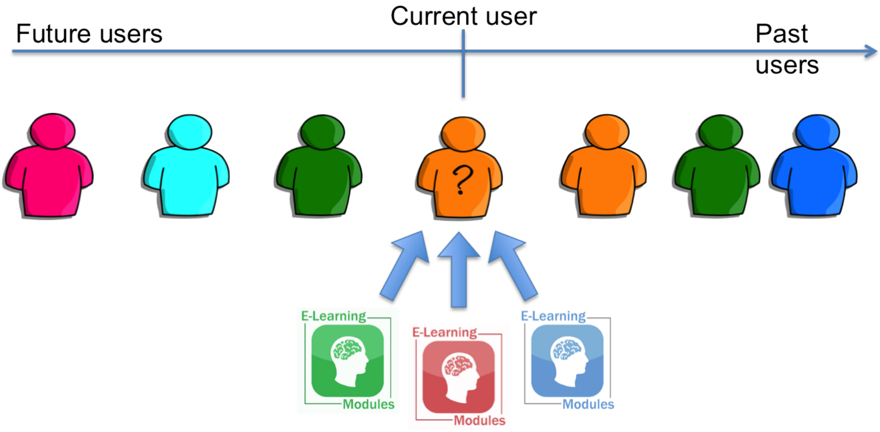
Multi-armed Bandit with *Hidden* Type



Multi-armed Bandit with *Hidden* Type

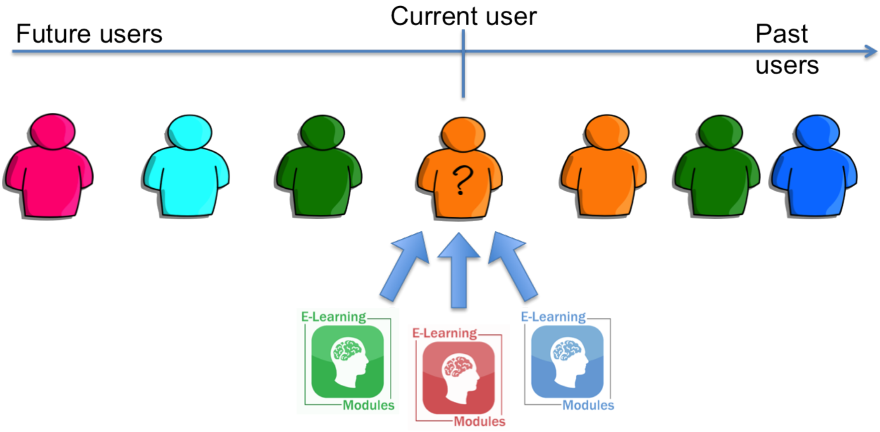


Multi-armed Bandit with *Hidden* Type



Learning the ***hidden type*** of bandit significantly reduces the regret

Multi-armed Bandit with *Hidden* Type



Learning the ***hidden type*** of bandit significantly reduces the regret

[Agrawal et al., IEEE TAC'89] [Azar et al., NIPS'13], [Maillard et al., ICML'14] [Lattimore and Munos, NIPS'14]

The Setting

- Set of arms $\mathcal{A} = \{1, \ldots, K\}$
- Set of types $\Theta = \{\theta_1, \ldots, \theta_m\}$
- Distribution over types ρ
- Arm mean $\mu_i(\theta)$, best arm $i_*(\theta)$, best value $\mu_*(\theta)$
- Arm gap $\Delta_i(\theta) = \mu_*(\theta) - \mu_i(\theta)$
- Model gap $\Gamma_i(\theta, \theta') = |\mu_i(\theta) - \mu_i(\theta')|$

The Protocol

- **for** $j = 1, \ldots, J$ (episodes)
  - Draw *task* $\bar{\theta}^j$ from ρ
  - **for** $t = 1, \ldots, n$ (steps)
    - Learner selects arm I_t^j
    - Learner observes $X_t^j \sim \nu_i(\bar{\theta}^j)$
    - Learner updates estimates
  - **endfor**
- **endfor**

The Protocol

- **for** $j = 1, \ldots, J$ (episodes)
  - Draw *task* $\bar{\theta}^j$ from ρ
  - **for** $t = 1, \ldots, n$ (steps)
    - Learner selects arm I_t^j
    - Learner observes $X_t^j \sim \nu_i(\bar{\theta}^j)$
    - Learner updates estimates
  - **endfor**
- **endfor**

- Task regret $R_n^j = \sum_{i \neq i_*(\bar{\theta}^j)} T_{i,n}^j \Delta_i(\bar{\theta}^j)$
- Global regret $R_J = \sum_{j=1}^J R_n^j$

The Protocol

- **for** $j = 1, \ldots, J$ (episodes)
  - Draw *task* $\bar{\theta}^j$ from ρ
  - **for** $t = 1, \ldots, n$ (steps)
    - Learner selects arm I_t^j
    - Learner observes $X_t^j \sim \nu_i(\bar{\theta}^j)$
    - Learner updates estimates
  - **endfor**
- **endfor**

- Task regret $R_n^j = \sum_{i \neq i_*(\bar{\theta}^j)} T_{i,n}^j \Delta_i(\bar{\theta}^j)$
- Global regret $R_J = \sum_{j=1}^J R_n^j$

$\Rightarrow$ Usually n is *small* and J is *large*

The Advantage of Knowing Θ

Assumption: $\{\mu_i(\theta)\}_{i,\theta}$ are known

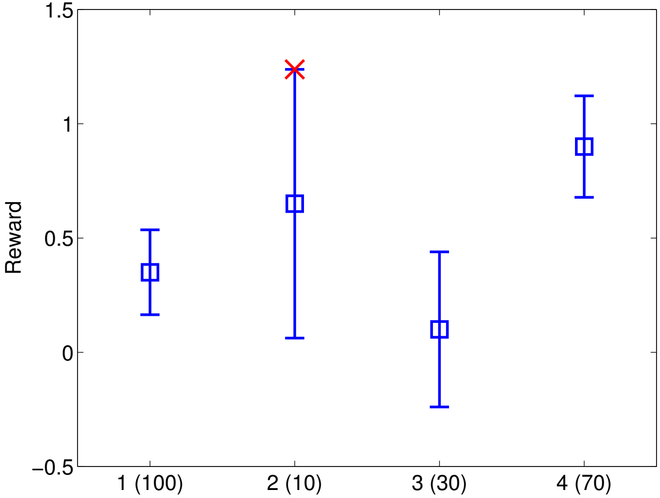
The Advantage of Knowing Θ

Assumption: $\{\mu_i(\theta)\}_{i,\theta}$ are known

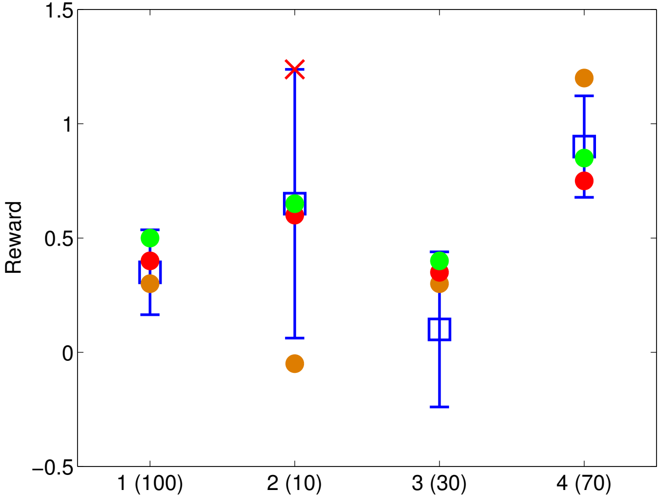
mUCB$(\{\mu_i(\theta)\}_{i,\theta})$

- **for** $t = 1, \ldots, n$ (steps)
  - Let $\epsilon_{i,t} = c\sqrt{\log(t)/T_{i,t}}$
  - Build set of active types $\Theta_t = \{\theta : \forall i, |\mu_i(\theta) - \widehat{\mu}_{i,t}| \leq \epsilon_{i,t}\}$
  - Select $\theta_t = \arg\max_{\theta \in \Theta_t} \mu_*(\theta)$
  - Pull arm $I_t = i_*(\theta_t)$
  - Learner observes reward and update estimates
- **endfor**

The Advantage of Knowing Θ



The Advantage of Knowing Θ



The Advantage of Knowing Θ

Theorem

Given $\{\mu_i(\theta)\}_{i,\theta}$, the mUCB achieves a per-task regret with type $\bar{\theta}$

$$\mathbb{E}[R_n(\bar{\theta})] \leq \sum_{i \in \mathcal{A}_+} \frac{2\log(K_* n^3)}{\min_{\theta \in \Theta_{+,i}} \Gamma_i(\theta, \bar{\theta})}$$

The Advantage of Knowing Θ

Theorem

Given $\{\mu_i(\theta)\}_{i,\theta}$, the mUCB achieves a per-task regret with type $\bar{\theta}$

$$\mathbb{E}[R_n(\bar{\theta})] \leq \sum_{i \in \mathcal{A}_+} \frac{2\log(K_* n^3)}{\min_{\theta \in \Theta_{+,i}} \Gamma_i(\theta, \bar{\theta})}$$

- Optimistic types $\Theta_+(\bar{\theta}) = \{\theta : \mu_*(\theta) > \mu_*(\bar{\theta})\}$

The Advantage of Knowing Θ

Theorem

Given $\{\mu_i(\theta)\}_{i,\theta}$, the mUCB achieves a per-task regret with type $\bar{\theta}$

$$\mathbb{E}[R_n(\bar{\theta})] \leq \sum_{i \in \mathcal{A}_+} \frac{2\log(K_* n^3)}{\min_{\theta \in \Theta_{+,i}} \Gamma_i(\theta, \bar{\theta})}$$

- Optimistic types $\Theta_+(\bar{\theta}) = \{\theta : \mu_*(\theta) > \mu_*(\bar{\theta})\}$
- Optimistic types with optimal arm i, $\Theta_{+,i}(\bar{\theta}) = \{\theta \in \Theta_+ : i_*(\theta) = i\}$

The Advantage of Knowing Θ

Theorem

Given $\{\mu_i(\theta)\}_{i,\theta}$, the mUCB achieves a per-task regret with type $\bar{\theta}$

$$\mathbb{E}[R_n(\bar{\theta})] \leq \sum_{i \in \mathcal{A}_+} \frac{2\log(K_* n^3)}{\min_{\theta \in \Theta_{+,i}} \Gamma_i(\theta, \bar{\theta})}$$

- Optimistic types $\Theta_+(\bar{\theta}) = \{\theta : \mu_*(\theta) > \mu_*(\bar{\theta})\}$
- Optimistic types with optimal arm i, $\Theta_{+,i}(\bar{\theta}) = \{\theta \in \Theta_+ : i_*(\theta) = i\}$
- Possible optimal arms $\mathcal{A}_*(\Theta') = \{i \in \mathcal{A} : \exists \theta \in \Theta' : i = i_*(\theta)\}$

The Advantage of Knowing Θ

Theorem

Given $\{\mu_i(\theta)\}_{i,\theta}$, the mUCB achieves a per-task regret with type $\bar{\theta}$

$$\mathbb{E}[R_n(\bar{\theta})] \leq \sum_{i \in \mathcal{A}_+} \frac{2\log(K_* n^3)}{\min_{\theta \in \Theta_{+,i}} \Gamma_i(\theta, \bar{\theta})}$$

- Optimistic types $\Theta_+(\bar{\theta}) = \{\theta : \mu_*(\theta) > \mu_*(\bar{\theta})\}$
- Optimistic types with optimal arm i, $\Theta_{+,i}(\bar{\theta}) = \{\theta \in \Theta_+ : i_*(\theta) = i\}$
- Possible optimal arms $\mathcal{A}_*(\Theta') = \{i \in \mathcal{A} : \exists \theta \in \Theta' : i = i_*(\theta)\}$
- Possible optimal arms of optimistic types $\mathcal{A}_+ = \mathcal{A}_*(\Theta_+(\bar{\theta}))$

Learning the Hidden Types

Consider the random vector $\boldsymbol{Z} \in \mathbb{R}^K$, such that $[\boldsymbol{Z}]_i$ is obtained by sampling θ from ρ and and then sampling a reward from $\nu_i(\theta)$

- First moment $\mathbb{E}[\boldsymbol{Z}|\theta] = \boldsymbol{\mu}(\theta) \in \mathbb{R}^K$
- Second moment $M_2 = \mathbb{E}[\boldsymbol{Z}_1 \otimes \boldsymbol{Z}_2]$
- Third moment $M_3 = \mathbb{E}[\boldsymbol{Z}_1 \otimes \boldsymbol{Z}_2 \otimes \boldsymbol{Z}_3]$

Learning the Hidden Types

Consider the random vector $\boldsymbol{Z} \in \mathbb{R}^K$, such that $[\boldsymbol{Z}]_i$ is obtained by sampling θ from ρ and and then sampling a reward from $\nu_i(\theta)$

- First moment $\mathbb{E}[\boldsymbol{Z}|\theta] = \boldsymbol{\mu}(\theta) \in \mathbb{R}^K$
- Second moment $M_2 = \mathbb{E}[\boldsymbol{Z}_1 \otimes \boldsymbol{Z}_2]$
- Third moment $M_3 = \mathbb{E}[\boldsymbol{Z}_1 \otimes \boldsymbol{Z}_2 \otimes \boldsymbol{Z}_3]$

$$M_2 \overset{iid}{=} \sum_{\theta \in \Theta} \rho(\theta) \mathbb{E}[\boldsymbol{Z}_1|\theta] \otimes \mathbb{E}[\boldsymbol{Z}_2|\theta] = \sum_{\theta \in \Theta} \rho(\theta) \boldsymbol{\mu}(\theta) \otimes \boldsymbol{\mu}(\theta)$$

$$M_3 \overset{iid}{=} \sum_{\theta \in \Theta} \rho(\theta) \mathbb{E}[\boldsymbol{Z}_1|\theta] \otimes \mathbb{E}[\boldsymbol{Z}_2|\theta] \otimes \mathbb{E}[\boldsymbol{Z}_3|\theta] = \sum_{\theta \in \Theta} \rho(\theta) \boldsymbol{\mu}(\theta) \otimes \boldsymbol{\mu}(\theta) \otimes \boldsymbol{\mu}(\theta)$$

Learning the Hidden Types

Consider the random vector $\boldsymbol{Z} \in \mathbb{R}^K$, such that $[\boldsymbol{Z}]_i$ is obtained by sampling θ from ρ and and then sampling a reward from $\nu_i(\theta)$

- First moment $\mathbb{E}[\boldsymbol{Z}|\theta] = \boldsymbol{\mu}(\theta) \in \mathbb{R}^K$
- Second moment $M_2 = \mathbb{E}[\boldsymbol{Z}_1 \otimes \boldsymbol{Z}_2]$
- Third moment $M_3 = \mathbb{E}[\boldsymbol{Z}_1 \otimes \boldsymbol{Z}_2 \otimes \boldsymbol{Z}_3]$

$$M_2 \overset{iid}{=} \sum_{\theta\in\Theta} \rho(\theta)\mathbb{E}[\boldsymbol{Z}_1|\theta] \otimes \mathbb{E}[\boldsymbol{Z}_2|\theta] = \sum_{\theta\in\Theta} \rho(\theta)\boldsymbol{\mu}(\theta) \otimes \boldsymbol{\mu}(\theta)$$

$$M_3 \overset{iid}{=} \sum_{\theta\in\Theta} \rho(\theta)\mathbb{E}[\boldsymbol{Z}_1|\theta] \otimes \mathbb{E}[\boldsymbol{Z}_2|\theta] \otimes \mathbb{E}[\boldsymbol{Z}_3|\theta] = \sum_{\theta\in\Theta} \rho(\theta)\boldsymbol{\mu}(\theta) \otimes \boldsymbol{\mu}(\theta) \otimes \boldsymbol{\mu}(\theta)$$

$\Rightarrow$ $\rho(\theta)$ and $\boldsymbol{\mu}(\theta)$ are the result of tensor decomposition of M_3 (after orthogonalization using M_2)

Learning the Hidden Types

Assumption: $T^j_{i,n} \geq 3$ (can be forced by the algorithm)

Learning the Hidden Types

Assumption: $T^j_{i,n} \geq 3$ (can be forced by the algorithm)

At each episode l split the samples in three (independent) batches

$$[\widetilde{\mu}^l_1]_i = \frac{3}{T^l_{i,n}} \sum_{t=1}^{T^l_{i,n}/3} Y^l_{i,t}, \quad [\widetilde{\mu}^l_2]_i = \frac{3}{T^l_{i,n}} \sum_{t=T^l_{i,n}/3+1}^{2T^l_{i,n}/3} Y^l_{i,t}, \quad [\widetilde{\mu}^l_3]_i = \frac{3}{T^l_{i,n}} \sum_{t=2T^l_{i,n}/3+1}^{T^l_{i,n}} Y^l_{i,t},$$

Learning the Hidden Types

Assumption: $T^j_{i,n} \geq 3$ (can be forced by the algorithm)

At each episode l split the samples in three (independent) batches

$$[\widetilde{\boldsymbol{\mu}}^l_1]_i = \frac{3}{T^l_{i,n}} \sum_{t=1}^{T^l_{i,n}/3} Y^l_{i,t}, \quad [\widetilde{\boldsymbol{\mu}}^l_2]_i = \frac{3}{T^l_{i,n}} \sum_{t=T^l_{i,n}/3+1}^{2T^l_{i,n}/3} Y^l_{i,t}, \quad [\widetilde{\boldsymbol{\mu}}^l_3]_i = \frac{3}{T^l_{i,n}} \sum_{t=2T^l_{i,n}/3+1}^{T^l_{i,n}} Y^l_{i,t},$$

Compute estimates

$$\widehat{M}_2 = \frac{1}{j}\sum_{l=1}^{j} \widetilde{\boldsymbol{\mu}}^l_1 \otimes \widetilde{\boldsymbol{\mu}}^l_2, \qquad \text{and} \qquad \widehat{M}_3 = \frac{1}{j}\sum_{l=1}^{j} \widetilde{\boldsymbol{\mu}}^l_1 \otimes \widetilde{\boldsymbol{\mu}}^l_2 \otimes \widetilde{\boldsymbol{\mu}}^l_3.$$

Learning the Hidden Types

Lemma

*$\widehat{M}_2$ and $\widehat{M}_3$ are unbiased estimators of M_2 and M_3 and**

$$||M_3 - \widehat{M}_3|| \leq K^{3/2}\sqrt{\frac{\log(K/\delta)}{j}}; \qquad ||M_2 - \widehat{M}_2|| \leq K\sqrt{\frac{\log(K/\delta)}{j}}$$

with high probability w.r.t. tasks and samples randomness.

*Up to constants

Learning the Hidden Types

Assumptions

- $\{\boldsymbol{\mu}(\theta)\}_\theta$ are linearly independent (i.e., $m < K$)
- $\rho(\theta) > 0$ for all $\theta \in \Theta$

Learning the Hidden Types

Assumptions

- $\{\boldsymbol{\mu}(\theta)\}_\theta$ are linearly independent (i.e., $m < K$)
- $\rho(\theta) > 0$ for all $\theta \in \Theta$

Theorem

There exists J_0 such that for any $j \geq J_0$ (up to permutation π)

$$\|\boldsymbol{\mu}(\theta) - \widehat{\boldsymbol{\mu}}^j(\pi(\theta))\| \leq \epsilon^j := C(\Theta)K^{2.5}m^2\sqrt{\frac{\log(K/\delta)}{j}}$$

with

$$C(\Theta) := C\lambda_{\max}\sqrt{\sigma_{\max}/\sigma_{\min}^3}\,(\sigma_{\max}/\Gamma_\sigma + 1/\sigma_{\min} + 1/\sigma_{\max})$$

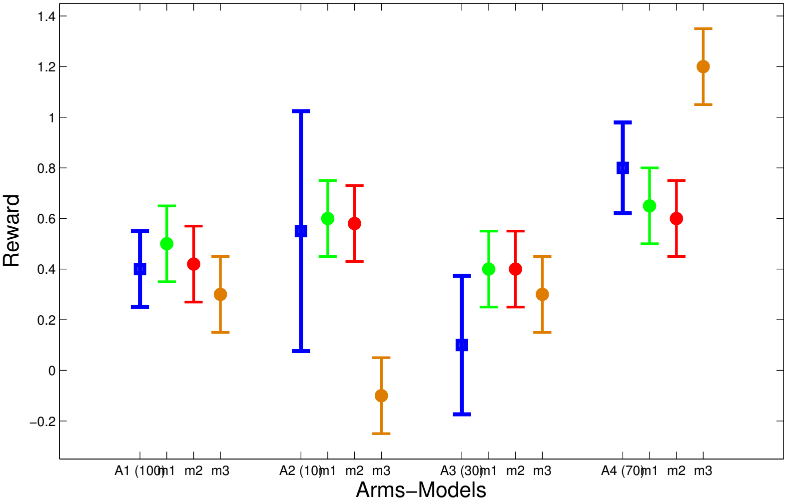
with high probability and independently from the bandit strategy (as soon as $T^l_{i,n} \geq 3$).

The Advantage of Learning Θ

tUCB

- **for** $j = 1, \ldots, J$ (tasks)
  - Let $\epsilon^j = C\sqrt{\log(K/\delta)/j}$
  - **for** $t = 1, \ldots, n$ (steps)
    - Let $\epsilon_{i,t} = c\sqrt{\log(t)/T_{i,t}}$
    - Build set of active types $\widehat{\Theta}_t = \{\theta : \forall i, |\widehat{\mu}_i(\theta) - \widehat{\mu}_{i,t}| \leq \epsilon_{i,t} + \epsilon^j\}$
    - Compute $B_t(i;\theta) = \min\left\{(\widehat{\mu}_i(\theta) + \epsilon^j), (\widehat{\mu}_{i,t} + \epsilon_{i,t})\right\}$
    - Select $\theta_t = \arg\max_{\theta \in \widehat{\Theta}_t} \max_i B_t(i;\theta)$
    - Pull arm $I_t = \arg\max_i B_t(i;\theta_t)$
    - Learner observes reward and update estimates
  - **endfor**
- **endfor**

The Advantage of Learning Θ



The Advantage of Learning Θ

Theorem

If tUCB is run over J episodes then

$$R_J \leq \sum_{j=1}^{J} \Bigg(\sum_{i \in \mathcal{A}_1^j} \min \Bigg\{ \frac{2 \log \left(Kn^2/\delta\right)}{\Delta_i(\bar{\theta}^j)^2}, \frac{\log \left(Kn^2/\delta\right)}{2 \min\limits_{\theta \in \Theta^j_{i,+}(\bar{\theta}^j)} \widehat{\Gamma}^j_i(\theta; \bar{\theta}^j)^2} \Bigg\} \Delta_i(\bar{\theta}^j)$$

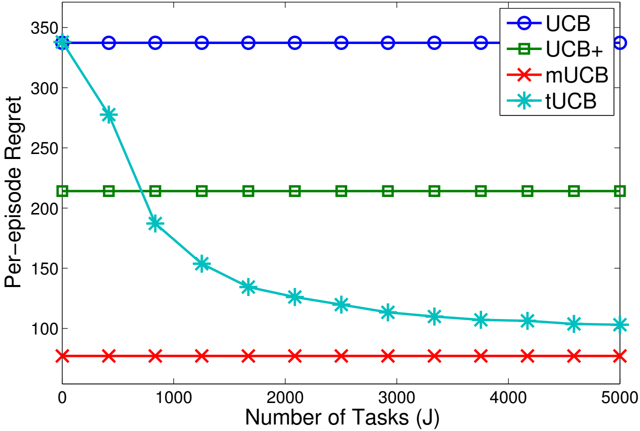
$$+ \sum_{i \in \mathcal{A}_2^j} \frac{2 \log \left(Kn^2/\delta\right)}{\Delta_i(\bar{\theta}^j)} \Bigg),$$

where (because of ϵ^j)

- $\mathcal{A}_1^j$ *arms optimal for models that can be discarded*
- $\mathcal{A}_2^j$ *arms optimal for models that cannot be discarded*

The Advantage of Learning Θ

$K = 7$, $m = 5$ with small model gaps



Summary

Pros

- Smooth integration of LVM with MAB
- Performance is never worse than UCB and it gets better at each task

Summary

Pros

- Smooth integration of LVM with MAB
- Performance is never worse than UCB and it gets better at each task

Cons

- Constants in ϵ^j are mostly unknown
- Residual exploration of all arms

Summary

Pros

- Smooth integration of LVM with MAB
- Performance is never worse than UCB and it gets better at each task

Cons

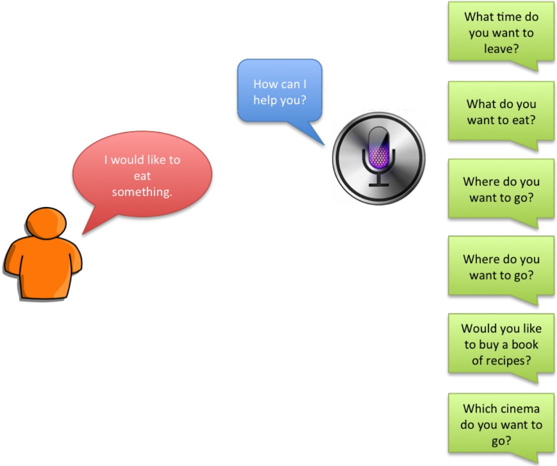
- Constants in ϵ^j are mostly unknown
- Residual exploration of all arms

Questions

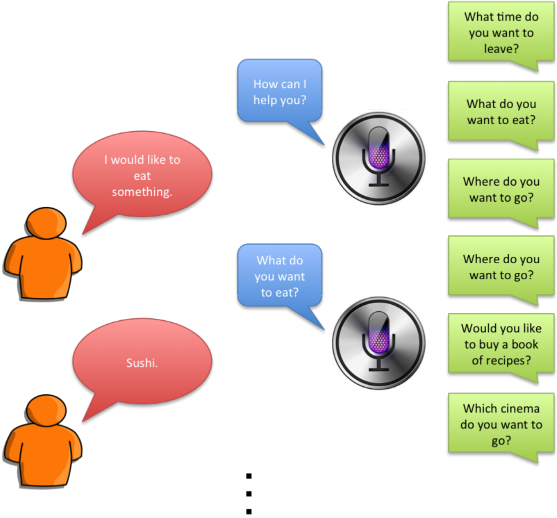
- Is it possible to "accelerate" the model learning by exploring more at the beginning?
- How do we estimate m?

Outline

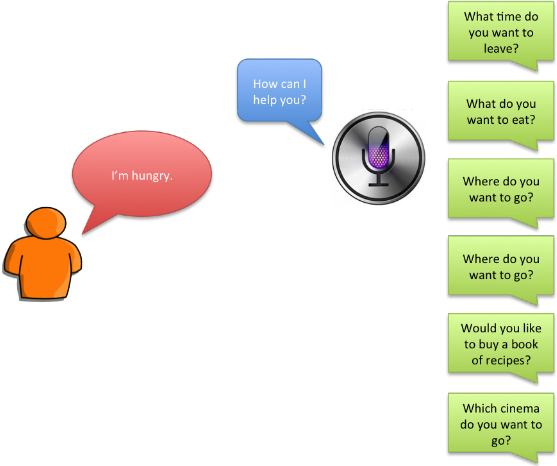
Partially Observable Markov Decision Process



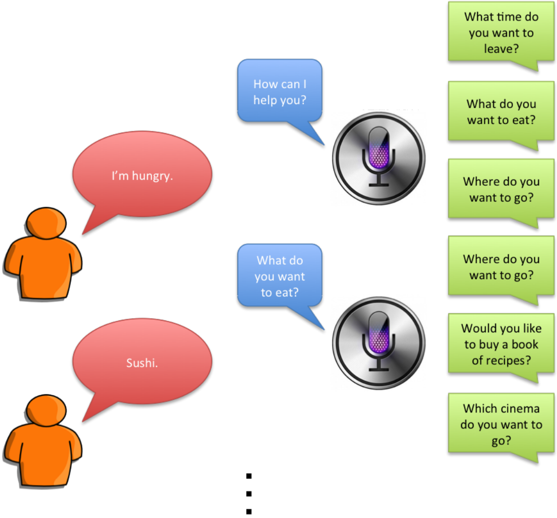
Partially Observable Markov Decision Process



Partially Observable Markov Decision Process



Partially Observable Markov Decision Process



Partially Observable Markov Decision Process

Partially Observable Markov Decision Process

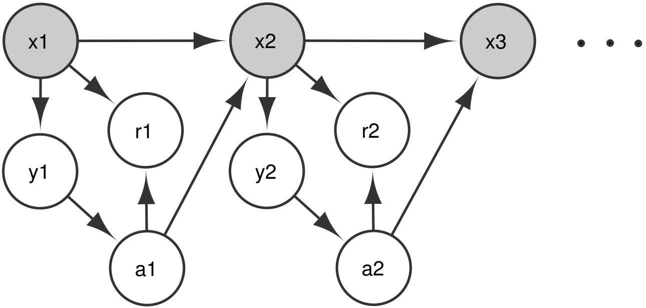
Learning the ***observation model*** allows learning better policies

The Setting

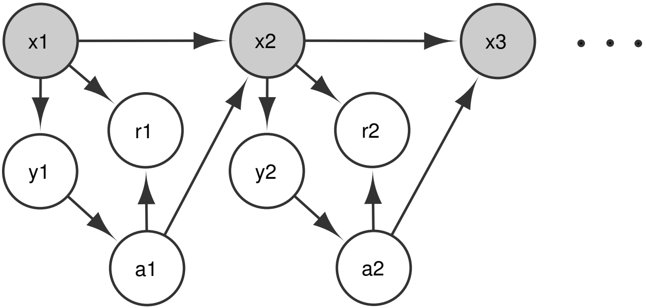
A finite POMDP M is a tuple $\langle \mathcal{X}, \mathcal{A}, \mathcal{Y}, \mathcal{R}, f_T, f_R, f_O \rangle$

- $\mathcal{X}$ is a finite state space with $|\mathcal{X}| = X$
- $\mathcal{A}$ is a finite action space with $|\mathcal{A}| = A$
- $\mathcal{Y}$ is a finite observation space with $|\mathcal{Y}| = Y$
- $\mathcal{R}$ is a finite reward space with $|\mathcal{R}| = R$ bounded by $r_{\max}$
- f_T is the transition density $f_T(x'|x, a)$
- f_R is the reward density $f_R(r|x, a)$
- f_O is the observation density $f_O(y|x)$

The Setting



The Setting



$\Rightarrow$ unlike the bandit model, here observations, actions, and hidden variables are very much ***dependent***

The Setting

Policies

- Deterministic memory-less: ***bad***

The Setting

Policies

- Deterministic memory-less: ***bad***
- Stochastic memory-less: ***ok*** [Barto et al., IEEE-SMC'83], [Loch, Singh, ICML'98], [Williams, Singh, NIPS'98], [Li et al., EJ of Op. Research'2011]

The Setting

Policies

- Deterministic memory-less: ***bad***
- Stochastic memory-less: ***ok*** [Barto et al., IEEE-SMC'83], [Loch, Singh, ICML'98], [Williams, Singh, NIPS'98], [Li et al., EJ of Op. Research'2011]
- Deterministic history-based: ***optimal*** (requires belief state)

The Setting

A (stochastic memory-less) policy π

- is defined by the density $f_\pi(a|y)$
- induces a stationary distribution $\omega_\pi(x)$
- has an average reward $\eta_\pi = \sum_{x \in \mathcal{X}} \omega(x) \overline{r}_\pi(x)$

$\Rightarrow$ Optimal policy $\pi^* = \arg\max_\pi \eta_\pi$

$\Rightarrow$ Regret $R_T = T\eta^* - \sum_{t=1}^{T} r_t$

The Setting

Assumptions

1. Set of policies $\mathcal{P} = \{\pi : \min_y \min_a f_\pi(a|y) > \pi_{\min}\}$
2. For any policy $\pi \in \mathcal{P}$, the Markov chain $f_{T,\pi}(x'|x)$ is ergodic
3. The observation model is not aliased (no two states with same observations)
4. The transition model is not aliased (no two states with same transitions)

The Setting

Assumptions

1. Set of policies $\mathcal{P} = \{\pi : \min_y \min_a f_\pi(a|y) > \pi_{\min}\}$
2. For any policy $\pi \in \mathcal{P}$, the Markov chain $f_{T,\pi}(x'|x)$ is ergodic
3. The observation model is not aliased (no two states with same observations)
4. The transition model is not aliased (no two states with same transitions)

Good news: 3 and 4 can be relaxed
Bad news: 1 and 2 cannot be removed (maybe...)

The Multi-View Model

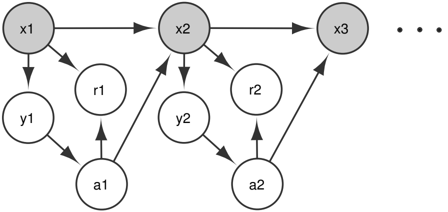
- Fix policy $\pi \in \mathcal{P}$
- For each action l, if $a_t = l$, construct views:

$$\vec{v}_{1,t}^{(l)} = (a_{t-1}, y_{t-1}, r_{t-1}); \;\; \vec{v}_{2,t}^{(l)} = (y_t, r_{t-1}); \;\; \vec{v}_{1,t}^{(l)} = (a_{t+1}, y_{t+1}, r_{t+1})$$

The Multi-View Model

- Fix policy $\pi \in \mathcal{P}$
- For each action l, if $a_t = l$, construct views:

$$\vec{v}_{1,t}^{(l)} = (a_{t-1}, y_{t-1}, r_{t-1}); \;\; \vec{v}_{2,t}^{(l)} = (y_t, r_{t-1}); \;\; \vec{v}_{1,t}^{(l)} = (a_{t+1}, y_{t+1}, r_{t+1})$$



$\Rightarrow$ $\vec{v}_{1,t}^{(l)}, \vec{v}_{2,t}^{(l)}, \vec{v}_{3,t}^{(l)}$ are three ***independent*** views of x_t (ie, conditioned on x_t they are independent random variables)

The Multi-View Model

Construct matrices

$$M_2^{(l)} = \mathbb{E}\left[\vec{v}_1^{(l)} \otimes \vec{v}_2^{(l)}\right]$$
$$M_3^{(l)} = \mathbb{E}\left[\vec{v}_1^{(l)} \otimes \vec{v}_2^{(l)} \otimes \vec{v}_3^{(l)}\right]$$

The Multi-View Model

Construct matrices

$$M_2^{(l)} = \mathbb{E}\left[\vec{v}_1^{(l)} \otimes \vec{v}_2^{(l)}\right]$$
$$M_3^{(l)} = \mathbb{E}\left[\vec{v}_1^{(l)} \otimes \vec{v}_2^{(l)} \otimes \vec{v}_3^{(l)}\right]$$

$\Rightarrow$ M_3 is neither symmetric nor orthogonal!

The Multi-View Model

Construct matrices

$$M_2^{(l)} = \mathbb{E}\Big[\vec{v}_1^{(l)} \otimes \vec{v}_2^{(l)}\Big]$$
$$M_3^{(l)} = \mathbb{E}\Big[\vec{v}_1^{(l)} \otimes \vec{v}_2^{(l)} \otimes \vec{v}_3^{(l)}\Big]$$

$\Rightarrow$ M_3 is neither symmetric nor orthogonal!

$\Rightarrow$ skipping details on how to symmetrize and orthogonalize (*hint*: transform the views and use M_2)

Recovering the POMDP parameters

- Given one single trajectory of T steps
- Use empirical estimates of $M_2^{(l)}$ and $M_3^{(l)}$ for each action
- Symmetrize and orthogonalize the tensor
- Estimate the model of the views

$$\vec{v}_{1,t}^{(l)} = (a_{t-1}, y_{t-1}, r_{t-1}); \;\; \vec{v}_{2,t}^{(l)} = (y_t, r_{t-1}); \;\; \vec{v}_{1,t}^{(l)} = (a_{t+1}, y_{t+1}, r_{t+1})$$

- From estimated views reconstruct the densities $\widehat{f}_O$, $\widehat{f}_T$, and $\widehat{f}_R$ (this step is ***non-trivial***)

Recovering the POMDP parameters

Theorem

For any state i and action l, with prob. $1-\delta$

$$||\widehat{f}_O(\cdot|i) - f_O(\cdot|i)||_1 \leq \mathcal{B}_O := \min_{l=1..A} \frac{YC_O}{\lambda_2^{(l)}} \sqrt{\frac{d' \log(1/\delta)}{N_l}}$$

$$||\widehat{f}_R(\cdot|i,l) - f_R(\cdot|i,l)||_1 \leq \mathcal{B}_R := \frac{RC_R}{\lambda_2^{(l)}} \sqrt{\frac{d' \log(1/\delta)}{N_l}}$$

$$||\widehat{f}_T(\cdot|\cdot,l) - f_T(\cdot|\cdot,l)||_F \leq \mathcal{B}_T := \max_{l'=1,\ldots,A} \frac{C_T d^2 A}{\lambda_2^{(l')}} \sqrt{\frac{d \log(1/\delta)}{N_{l'}}}$$

Recovering the POMDP parameters

Theorem

For any state i and action l, with prob. $1-\delta$

$$||\widehat{f}_O(\cdot|i) - f_O(\cdot|i)||_1 \leq \mathcal{B}_O := \min_{l=1..A} \frac{YC_O}{\lambda_2^{(l)}} \sqrt{\frac{d' \log(1/\delta)}{N_l}}$$

$$||\widehat{f}_R(\cdot|i,l) - f_R(\cdot|i,l)||_1 \leq \mathcal{B}_R := \frac{RC_R}{\lambda_2^{(l)}} \sqrt{\frac{d' \log(1/\delta)}{N_l}}$$

$$||\widehat{f}_T(\cdot|\cdot,l) - f_T(\cdot|\cdot,l)||_F \leq \mathcal{B}_T := \max_{l'=1,\dots,A} \frac{C_T d^2 A}{\lambda_2^{(l')}} \sqrt{\frac{d \log(1/\delta)}{N_{l'}}}$$

with

- $d = A \cdot Y \cdot R$ (*can be improved*)
- $\omega_{\min}^{(l)} = \min_{x \in \mathcal{X}} \omega_\pi^{(l)}(x)$ (*forced by explorative policy and ergodicity*)
- $\lambda_2^{(l)} = \min\{(\sigma_{1,3}^{(l)})^{3/2}; (\sigma_{1,3}^{(l)})^3 (\omega_{\min}^{(l)})^{1/2}\}\pi_{\min}$

The Spectral-Method UCRL

Initialize $t = 1$, initial state x_1

- **for** $k = 1, \ldots, K$ (episodes)
  - Set $t^{(k)} = t$
  - Compute the estimated POMDP $\widehat{M}$ using the spectral algorithm
  - Compute the optimistic policy $\widetilde{\pi}^{(k)}$
  - **while** $(t - t^{(k)} < 2(t^{(k)} - t^{(k-1)}))$
    - Execute $a_t \sim f_{\widetilde{\pi}^{(k)}}(\cdot|x_t)$
    - Obtain reward r_t, observe next state x_t, and set $t = t + 1$
  - **endwhile**
- **endfor**

In short: just UCRL1 with spectral method to estimate the POMDP.

The Spectral-Method UCRL

Theorem

SM-UCRL run over T rounds achieves an ϵ-regret

$$R_T^{\epsilon} = O\left(poly(d, d') \frac{\log(T)}{\epsilon^2} \right)$$

Summary

Pros

- Extension of spectral methods for LVM to active settings and (relatively...) smooth integration with UCRL
- Current version uses UCRL1 but can be extended to UCRL2 ($=R_T = O(\sqrt{T})$)
- Dependency on X, Y, R, O can be improved

Summary

Pros

- Extension of spectral methods for LVM to active settings and (relatively...) smooth integration with UCRL
- Current version uses UCRL1 but can be extended to UCRL2 ($=R_T = O(\sqrt{T})$)
- Dependency on X, Y, R, O can be improved

Cons

- Constants are unknown
- Requires persistently explorative policies
- Bad dependency on probability of poorly visited states

Summary

Pros

- Extension of spectral methods for LVM to active settings and (relatively...) smooth integration with UCRL
- Current version uses UCRL1 but can be extended to UCRL2 $(=R_T = O(\sqrt{T}))$
- Dependency on X, Y, R, O can be improved

Cons

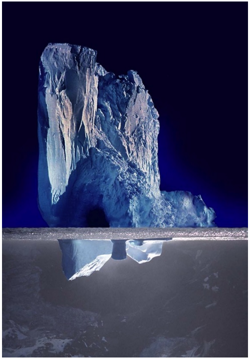
- Constants are unknown
- Requires persistently explorative policies
- Bad dependency on probability of poorly visited states

Questions

- Is it possible to use (partially) deterministic policies?
- Is it possible to remove ergodicity assumption (on bad policies)?

Outline

The Hidden World of Bandits

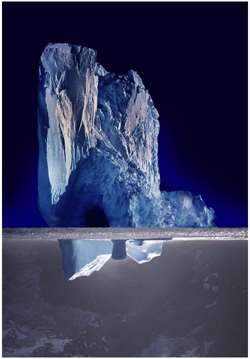


Many
bandit problems /
contexts / observations

Few hidden
structures

$\Rightarrow$ How do we learn *structures and solutions* at the same time?

The Hidden World of Bandits



Many
bandit problems /
contexts / observations

Few hidden
structures

⇒ How do we learn *structures and solutions* at the same time?

⇒ *Spectral tensor decomposition* for LVM and *MAB strategies* can be (often) integrated ***smoothly and effectively***.

Thank you!

Alessandro Lazaric
alessandro.lazaric@inria.fr
sequel.lille.inria.fr