Frugal Reinforcement Learningg for Stochastic Networks

Urtzi Ayesta

CNRS & IRIT

ANITI 26/11/2024

Team

IRIT
Urtzi Ayesta, DR CNRS,
Maaike Verloop, CR CNRS

LAAS
Céline Comte, CR CNRS,
Matthieu Jonckheere, DR CNRS (co-chair)
Balakrishna Prabhu, CR CNRS,

CNRS Team
https://solace.cnrs.fr/

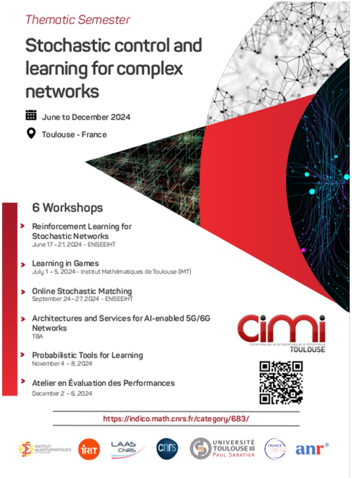
SOLACE CIMI Thematic Semester : Stochastic control and learning for complex networks

Shift from “model based” analysis to “data based” and ”machine learning”

Organize a series of workshops on methodological aspects and applications of machine learning for stochastic networks

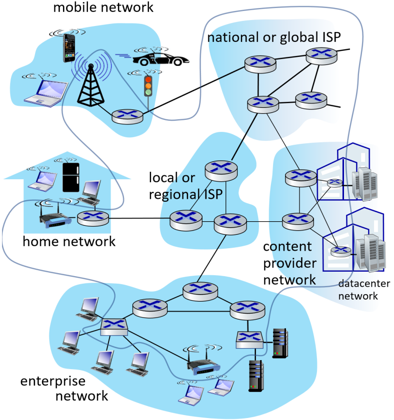
RL4SN, Online Stochastic Matching, Prob Tools for Learning, Learning in Games etc.

https://solace.cnrs.fr/

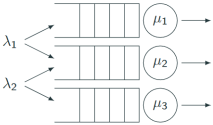


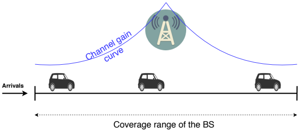
Stochastic Networks ??

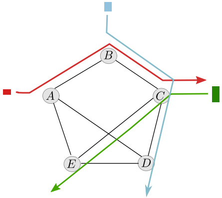
Models for computing infrastructure: Networks and data centers



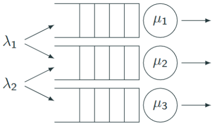
Resource Sharing in Networks

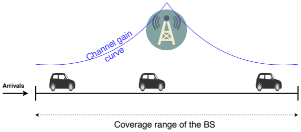


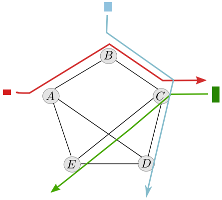




Resource Sharing in Networks

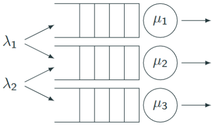


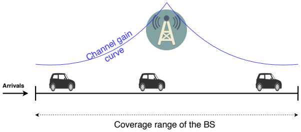


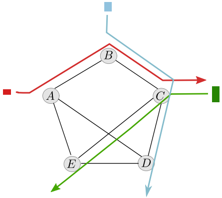


- **Challenge:** randomness, large-scale ...

Resource Sharing in Networks







- **Challenge:** randomness, large-scale ...

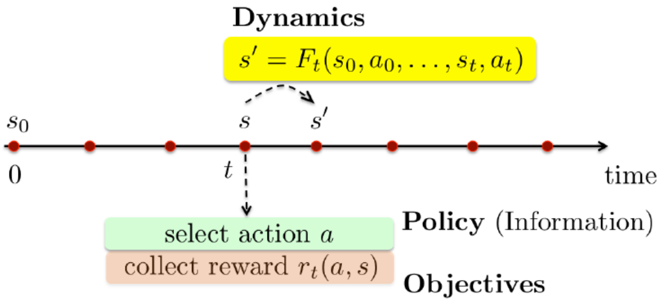
Stochastic Network: Discrete time, stochastic model restricted to positive orthant, long-run behavior, analysis and optimization

Control over time: How to take decisions over time in order to optimize certain objective function

Outline

- Basics Sequential Decision
- Basics RL
- Large scale RL
- Frugal RL

Sequential Decision Making



Objectives, a few examples:

Infinite discounted cost: $\max_\pi \mathbb{E}(\sum_{t=0}^{\infty} \alpha^t r(a_t^\pi, s_t^\pi))$

Average cost: $\max_\pi \lim_{T\to\infty} \frac{1}{T}\mathbb{E}(\sum_{t=0}^{T} r(a_t^\pi, s_t^\pi))$

Finite Horizon

- Discrete time $t = 1, 2, 3, \ldots,$
- A finite set of states, finite state of actions,
- Arbitrary Markov dynamics $p(s'|s, a)$
- Finite horizon T

Finite Horizon

- Discrete time $t = 1, 2, 3, \ldots,$
- A finite set of states, finite state of actions,
- Arbitrary Markov dynamics $p(s'|s, a)$
- Finite horizon T

$$V_T(i) \quad = \quad \max_\pi \mathbb{E}_\pi(\sum_{t=1}^{T} R_t)$$

Finite Horizon (cont.)

Consider horizon T. Assume $V_{T-1}(j)$ is known for all j. Take action a, which yields reward $r(i, a)$.

What is the best reward we can get ?

Finite Horizon (cont.)

Consider horizon T. Assume $V_{T-1}(j)$ is known for all j. Take action a, which yields reward $r(i, a)$.

What is the best reward we can get ?

$$r(i, a) + \sum_j p(j|i, a) V_{T-1}(j)$$

Finite Horizon (cont.)

Consider horizon T. Assume $V_{T-1}(j)$ is known for all j. Take action a, which yields reward $r(i,a)$.

What is the best reward we can get ?

$$r(i,a) + \sum_j p(j|i,a) V_{T-1}(j)$$

Then, the best action is

$$V_T(i) = \max_a \left(r(i,a) + \sum_j p(j|i,a) V_{T-1}(j) \right)$$

Finite Horizon (cont.)

Consider horizon T. Assume $V_{T-1}(j)$ is known for all j. Take action a, which yields reward $r(i,a)$.
What is the best reward we can get ?

$$r(i,a) + \sum_j p(j|i,a) V_{T-1}(j)$$

Then, the best action is

$$V_T(i) = \max_a \left(r(i,a) + \sum_j p(j|i,a) V_{T-1}(j) \right)$$

We can first solve $V_1(i) = \max_a \{ r(i,a) \}$, and then $V^2(i) = \max_a \left(r(i,a) + \alpha \sum_j p(j|i,a) V^1(j) \right)$,
then $V_2(i), \ldots, V_T(i)$

Finite Horizon (cont.)

Consider horizon T. Assume $V_{T-1}(j)$ is known for all j. Take action a, which yields reward $r(i,a)$.
What is the best reward we can get ?

$$r(i,a) + \sum_j p(j|i,a) V_{T-1}(j)$$

Then, the best action is

$$V_T(i) = \max_a \left(r(i,a) + \sum_j p(j|i,a) V_{T-1}(j) \right)$$

We can first solve $V_1(i) = \max_a \{ r(i,a) \}$, and then
$V^2(i) = \max_a \left(r(i,a) + \alpha \sum_j p(j|i,a) V^1(j) \right)$,
then $V_2(i), \ldots, V_T(i)$
Known as **Optimality Equation**, **Dynamic Programming**, **Bellman's equation** ...

Richard Bellman

1920 - 1984
American applied mathematician

Introduced **Dynamic Programming** (DP) as a method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions.

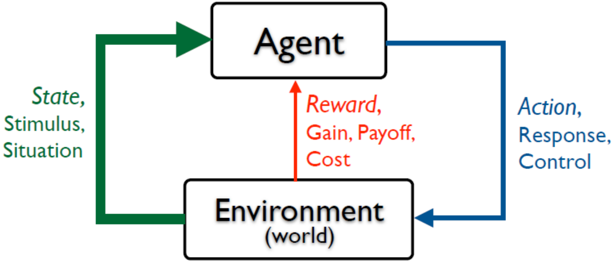
Outline

- Basics Sequential Decision
- Basics RL
- Large scale RL
- Frugal RL

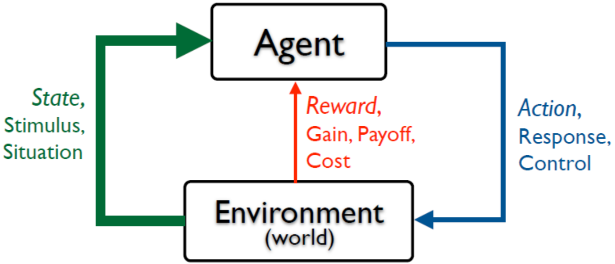
What is Reinforcement Learning

- Agent-oriented learning – learning by interacting with an environment to achieve a goal
- Learning by trial and error
  - can tell for itself when it is right or wrong
  - explore vs. exploit trade-off

The RL setting

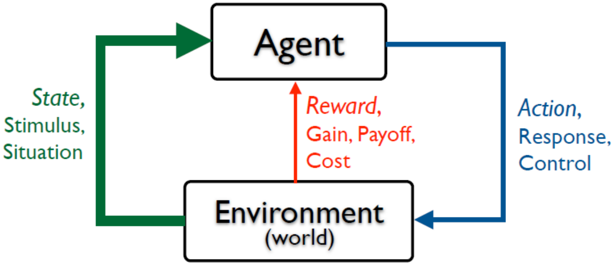


The RL setting



- environment is the "Markov Chain".

The RL setting



- environment is the "Markov Chain".
- Agent is given state S_t, takes an action A_t, and is returned a sample of the reward R_{t+1} and next state S_{t+1}.

Agent wants to learn $V(S_t)$ ...

Q-function

Watkins, PhD thesis, 1994

RL allows us to estimate $V(S_t)$ from samples of the $S_t, A_t, R_{t+1}, S_{t+1}, \ldots$

$$\hat{V}(S_t) \leftarrow R_{t+1} + \hat{V}(S_{t+1})$$

Q-function

Watkins, PhD thesis, 1994

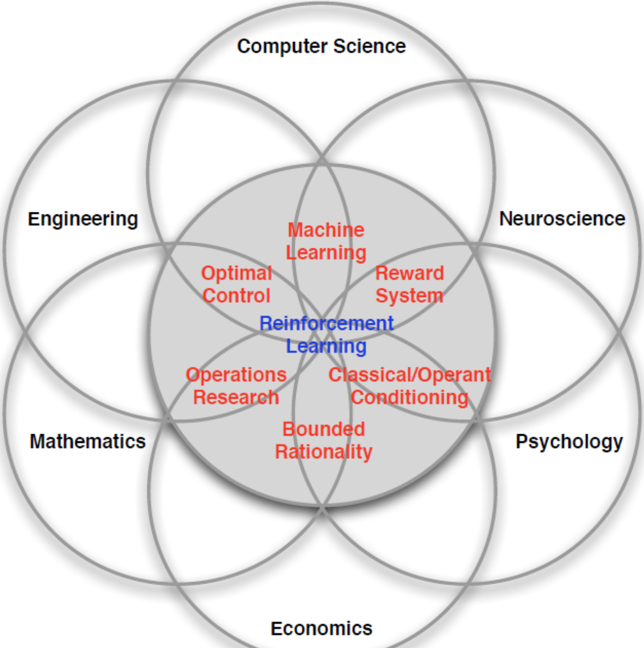
RL allows us to estimate $V(S_t)$ from samples of the $S_t, A_t, R_{t+1}, S_{t+1}, \ldots$

$$\hat{V}(S_t) \leftarrow R_{t+1} + \hat{V}(S_{t+1})$$

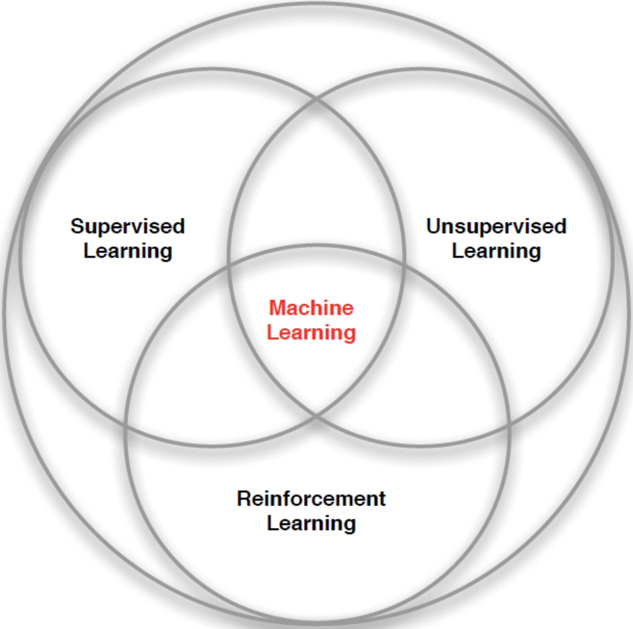
Theorem If all state and action pairs are **"observed"** infinitely many times, then

$$\hat{V}(S_t) \Longrightarrow V(S_t)$$

Many faces of RL



Branches of machine learning

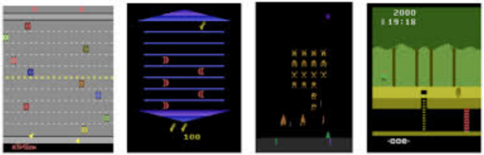


Outline

- Basics Sequential Decision
- Basics RL
- Large scale RL
- Frugal RL

Applications of RL

- Robotics
- Medicine
- Advertisement
- Resource management
- Game playing …

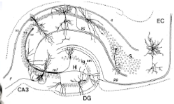


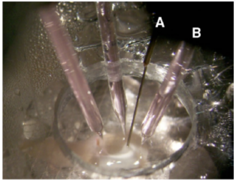
Some RL Successes

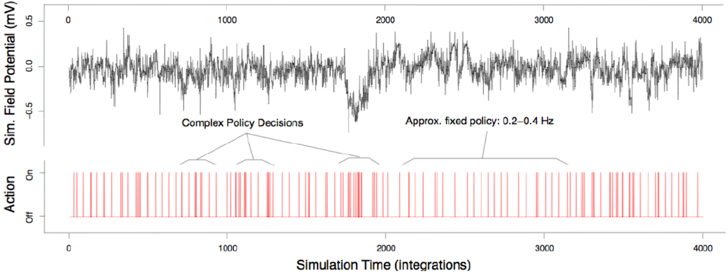
- Learned the world's best Backgammon player (Tesauro 1995)
- Helicopter autopilot (Ng, Coates et al. 2006+)
- ad placement, web site morphing, recommendation systems
- Human-level performance (Google Deepmind, 2015+)

DeepMind’s AlphaGo

Neurostimulation for epilepsy suppression







Approximate Solution Methods

- RL finds optimal policies if policies and functions can be saved in tables
- real world complex too large and comples
- Backgammon 10^{20} states, Go 10^{170} states, Helicopter continuous state space

How can we scale up the model-free methods for prediction and control?

Value function approximation

- So far we have represented value function by a lookup table $\hat{V}(s)$

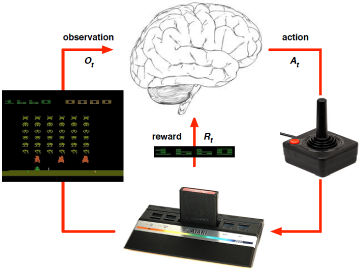
There are too many states and/or actions to store in memory

- It is too slow to learn the value of each state individually
  - Estimate value function with function approximation

$$\hat{V}(s, \boldsymbol{w}) \approx V(s)$$

  - Generalise from seen states to unseen states
  - Update parameter $\boldsymbol{w}$

Atari Example: Learning



- Rules unknown
- Learn directly from game-play
- Pick actions on joystic, see pixels and scores

Outline

- Basics Sequential Decision
- Basics RL
- Large scale RL
- Frugal RL

Democratic RL?

- RL (at large) has many success stories in the last two decades...
- but it elies on very demanding computational/data volume possibilities. E.g., games, data center control,...
- What about more "democratic" algorithms, especially for networking?
- Explorations mechanisms can be made more efficient by leveraging known structure/information?

Specifities of SN

- "rare" events
- sparse rewards
- physical queues, "infinite" state space

$\Longrightarrow$ we can use the underlying structure

Specifities of SN

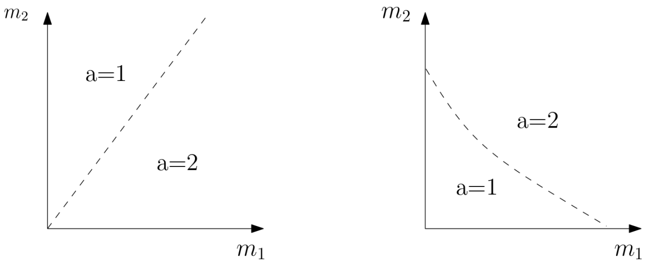
- "rare" events
- sparse rewards
- physical queues, "infinite" state space

$\Longrightarrow$ we can use the underlying structure

Objectives

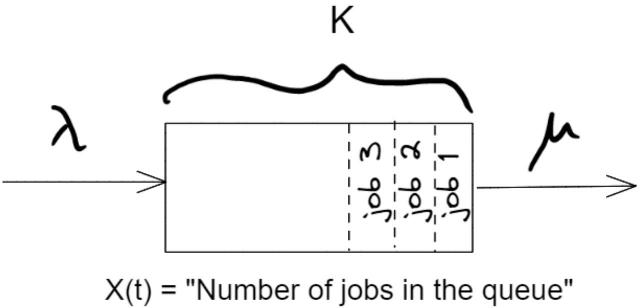
- Improve exploration
- Improve efficiency with lower data requirements
- Learning approximate optimal policies for SN

Specifities of SN



- Optimal policy might have a clear structure

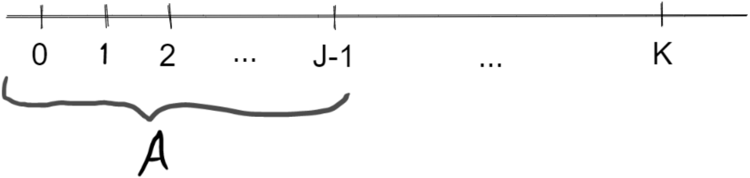
Example Improve exploration: Toy example - birth and death system



- Simplest model: M/M/1/K queue, constant BD rates.
- Parameters are unknown - in particular K
- Costs occur when blocking
- rare and sparse

Fleming-Viot particle systems to improve estimation

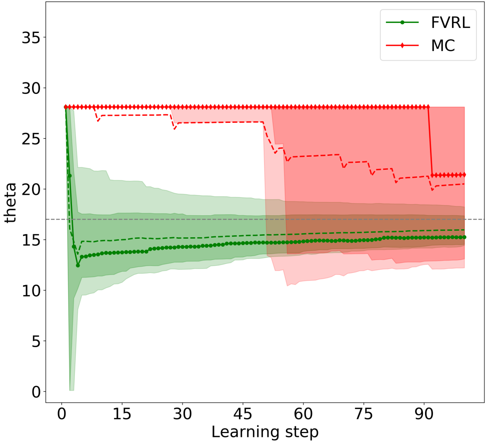
- Studied by Burdzy et al. in 1996 as genetic particle system



- N particles evolve independently - same dynamics
- When **absorbed** $\rightarrow$ **reactivation** to one of other $N-1$

Fleming-Viot particle systems for probability estimation (cont.)

Theorem: With FV, the estimation of blocking probability converges to the real value



Lagrangian relaxation

Original problem

$$\min_{\phi} \sum_{k=1}^{K} \mathbb{E}\left[C_k(N_k^{\phi}, S_k^{\phi}(N^{\phi}))\right]$$

$$\sum_{k=1}^{K} S_k^{\phi}(\vec{N}^{\phi}(t)) \leq M$$

Lagrangian relaxation (cont.)

Relax the constraint

$$\min_{\phi} \sum_{k=1}^{K} \mathbb{E}\left[C_k(N_k^{\phi}, S_k^{\phi}(N^{\phi}))\right]$$

$$\mathbb{E}\left(\sum_{k=1}^{K} S_k^{\phi}(\vec{N}^{\phi}(t))\right) \leq M$$

Lagrangian relaxation (cont.)

Unconstrained problem

$$\min_{\phi} \sum_{k=1}^{K} \mathbb{E}\left[C_k(N_k^{\phi}, S_k^{\phi}(N^{\phi}))\right] - W\left(M - \mathbb{E}\left(\sum_{k=1}^{K} S_k^{\phi}(N^{\phi})\right)\right)$$

K-dimensional problem $\Longrightarrow$ **K unidimensional** problems

$$\min_{\phi} \mathbb{E}\left[C(N^{\phi}, S^{\phi}(N^{\phi}))\right] - W\mathbb{E}\left(\mathbf{1}_{S^{\phi}(N^{\phi})=0}\right)$$

Whittle's index heuristic

Definition (Whittle's index)

$W_k(n_k) \equiv$ subsidy W such that is indifferent of action taken in state n_k.

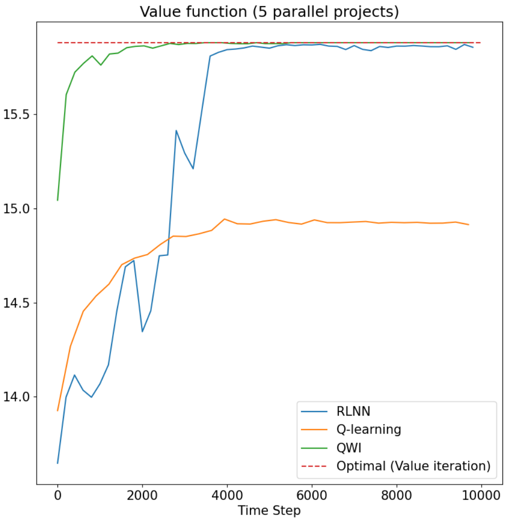
Whittle's index heuristic

Definition (Whittle's index)

$W_k(n_k) \equiv$ subsidy W such that is indifferent of action taken in state n_k.

- Serve bandit k if $W_k(n_k) \geq W$ **optimal** for relaxed problem
- **Heuristic** for original problem:
  Serve the M bandits with highest value for $W_k(n_k)$.

QWI : Learning Whittle's indices



Concluding remarks

- MDP and RL share a long history, with an elegant mathematical framework,
- Microcosm within AI, including planning, acting, learning, world modeling, knowledge representation
- Surge of interest in the SN community.
- To leverage the specific structures of the underlying of stochastic network problems to develop tailored learning algorithms