Reinforcement Learning for Wireless Network Optimization

Deniz Gündüz
Imperial College London
Head, Information Processing and Communications Lab
Intelligent Systems and Networks Group
www.imperial.ac.uk/ipc-lab
twitter.com/Imperial_IPCL

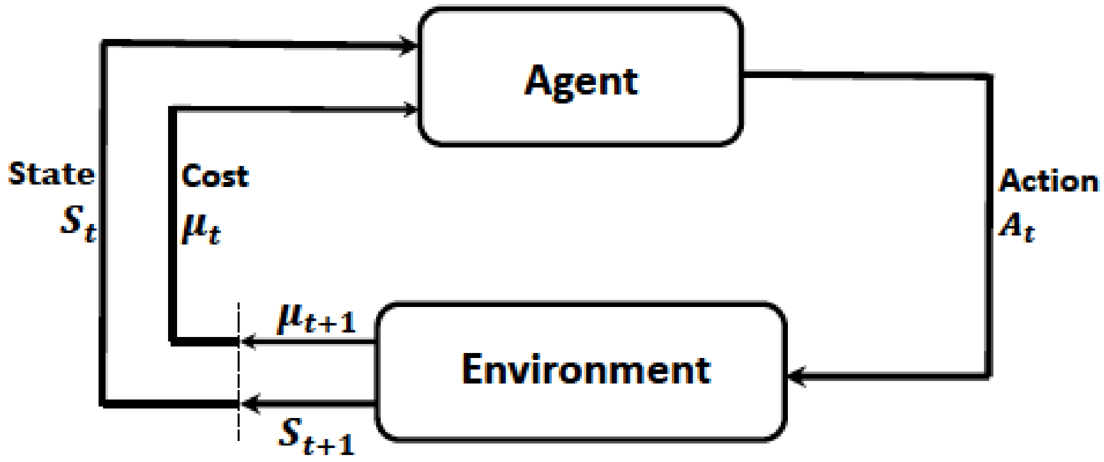
Workshop on Machine Learning for 5G
29 January 2018
ITU, Geneva, Switzerland

“The pace of change has never been this fast, yet it will never be this slow again.”

Justin Trudeau

Reinforcement Learning (RL)

- Agent- based learning: Agents learn by interacting with their environment
- Learning by trial and error
- Difference from supervised learning: No training data; wrong decisions are not corrected explicitly



- Learning happens online based on implicit feedback in the form of reward/ cost values
- Main challenge: Exploration vs. Exploitation

Some Recent RL Successes

- Minsky's Stochastic Neural Analogy Reinforcement Computer (SNARC) - 1951
- Matching world's best players in backgammon (Tesauro, 1992-95)
- Helicopter autopilot (Ng et al., 2006)
- Human level performance in Atari games through deep Q-networks (DeepMind, 2013-15)
- AlphaGo beats top Go player (DeepMind, 2016)
- Self thought AlphaGo Zero beats AlphaGo 100 to 0 (DeepMind, 2017)

Is RL Relevant for Wireless Networks?

- Wireless agents interact with their environment: Current action has future consequences.
- Feedback often is not explicit (low QoS, high error rate, high delay, etc.): exploration required.
- Training (exploration) consumes resources (spectrum, power, etc.) which should be accounted for.
- Resources shared among multiple agents.
- Proactive vs. Reactive network design

Proactive Network Design with RL

- Explore and learn network dynamics, and optimally exploit limited resources based on limited knowledge.

- User and context dependent resource provisioning: Know users better to provide user specific service

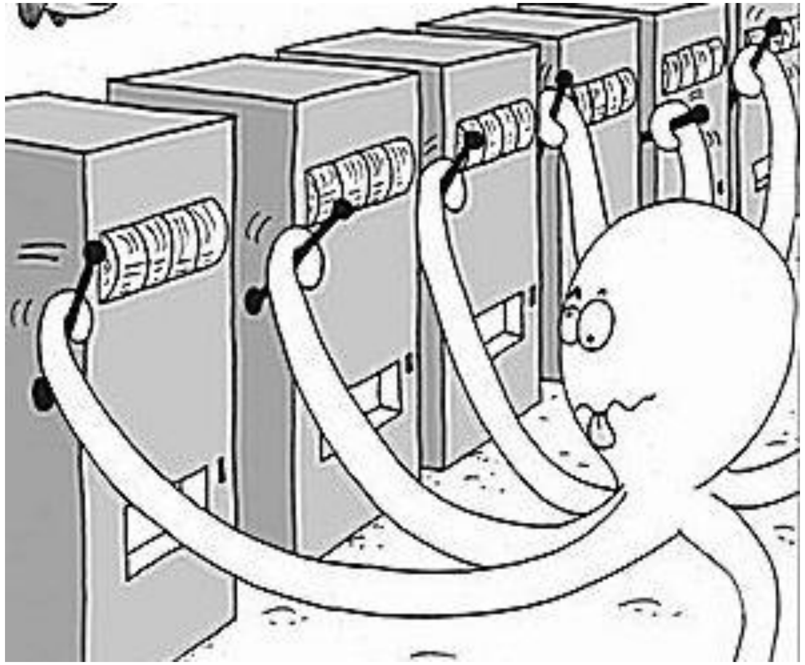
- Available huge amount of data (user mobility, traffic, connectivity, spectrum maps, etc.) provide unprecedented predictive capabilities (machine learning techniques). These can be exploited in an RL framework.

What Can Be Learned and Exploited?

- Mobility patterns
  - Future location, future cell association (trajectory) can be predicted accurately at user level
    - ➢Radio resource management, admission control, handover optimization
- Channel quality
  - Pathloss, shadow fading, distance, …
  - Radio Environment Maps (REM)
    - ➢Improve throughput, reduce channel sensing/ feedback resources
- Network traffic
  - Number of users (at the cell level), request patterns, content popularity
    - ➢Offloading, proactive caching, DASH/ transcoding optimization,

Multi-armed Bandit Machines

- A single-state RL problem
  - Slot machine with unknown arms.
  - Each time one arm is pulled, and yields a random reward with unknown mean.
  - Objective: maximize sum reward.
- Strategy:
  - Explore: good system knowledge, low reward.
  - Exploit: poor system knowledge, high rewards.
  - Exploration vs exploitation trade-off.
- Widely used in clinical trials, add placement, social influence maximization/ viral marketing, etc.



Picture credit: Microsoft Research

Multi-armed Bandits in Wireless Networks

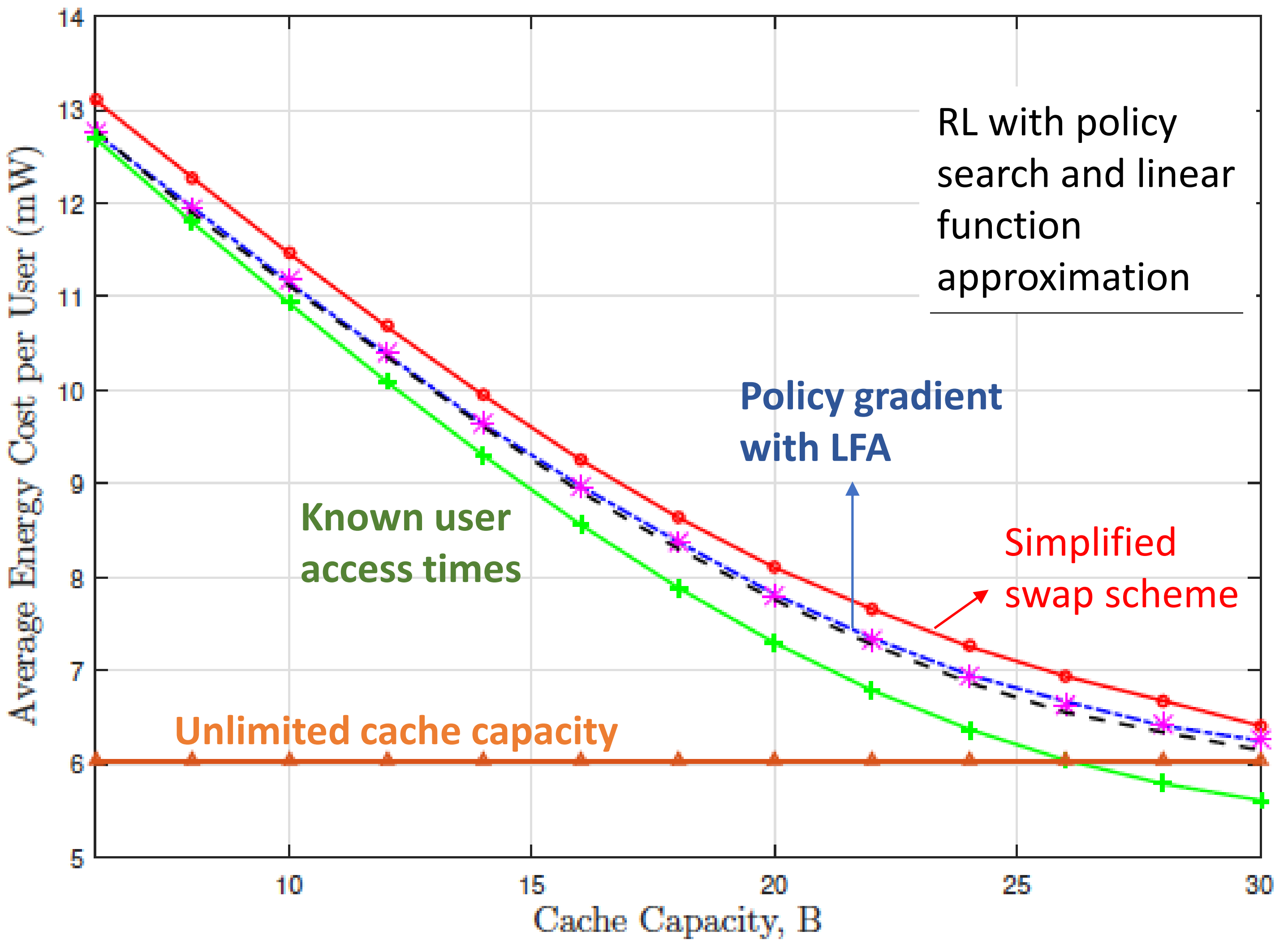
- Distributed channel access
- Scheduling with limited feedback (IoT, sensor, etc.) : Markov bandits
- Transmit power level / relay selection, base-station association
- Content placement (Blasco and Gunduz '2015)
  - Each content is an arm
  - Popularity of contents are not known *a priori*
  - Learn which content to place in a limited capacity cache storage (at an access point): combinatorial multi-armed bandit problem (with switching costs)

RL for Content Placement and Media Streaming

- Video dominates mobile data traffic
- Mostly pre-recorded video (YouTube, Netflix, BBC iPlayer, etc.)
- User behavior (mobility patterns, content requests) highly predictable
- Content popularity skewed: Few viral/ popular videos dominate demand traffic
- Recommendation systems can be used for content placement
- Predicted channel and traffic conditions, or mobility/ trajectory patterns can be used to optimize streaming
  - Buffer adaptation
  - Proactive video quality adaptation

Proactive Content Delivery

- Deliver content proactively during more favorable network conditions (better channel, less traffic, etc.)
- User demand instants, lifetime of contents, channel conditions are random and unknown
- Huge state space – optimal solution not feasible
- Propose a parametrized policy, apply policy search
- Linear function approximation (LFA)



Somuyiwa, Gyorgy, Gunduz ‘2017

What else are we up to @IPC-Lab?

- Distributed learning with communication constraints
- Code design with ML
- RL for *age of information* minimization
- Information bottleneck, privacy funnel (information theoretic analysis of fundamental limits of learning)

High Level Thoughts...

- We have the platform, data, users.. perfect background for building and applying ML.
- Reliability, security, privacy, …
- Well defined performance targets.
- Open (standardized?) test data.
- Lack of talent?

Thank you for your attention!