Multi-Armed Bandit Learning in IoT Networks
Poster for PhD Student Day, July 2017, Rennes (France)
By: Remi Bonnefoi & Lilian Besson
First.Last@CentraleSupélec.fr – SCEE Team, CentraleSupélec Rennes & IETR

1. Introduction & Goal

Goal: Fit more objects in a “Internet of Things” networks, keep a good Quality of Service.
- Hypothesis: objects choose channel \( k \in \{1, \ldots, K\} \), to use for each communication.
- Idea: use on-line Machine Learning algorithms ?
- Not so easy: each device takes its own decisions, without central control or communication, has light CPU/memory etc…
- \( \implies \) Solution: Decentralized MAB algorithms !

2. Model: Time/Frequency protocol Devices in the network

Model: One base station \( s = 1 \) RF channels (of same bandwidth).
\( S = D = 2000 \) end-devices in the network, with very low duty-cycle (one message every 1000 frame).
They are separated into two groups:
- \( S \) static devices \( s \) : poor RF abilities, and use only one channel to communicate with the base station. Their choice is fixed in time (stationary) and independent (i.i.d.).
- \( D \) dynamic devices \( d \) : richer RF abilities, can use all the available channels, by quickly reconfiguring their RF transceiver in the fly (dynamically).

3. Some baseline algorithms

Performance = successful transmission rate.
Three algorithms used for baseline comparison.
- Naive algorithm: all the \( D \) dynamic devices choose their channel \( k(t) = U(\{1, \ldots, K\}) \) purely uniformly at random.
- Optimal algorithms: exact algorithm (or a greedy approximation), when a centralized agent can affect the \( D \) dynamic devices to channels. Inapplicable in practice as we need a decentralized approach, but it gives a baseline for comparison.

4. Multi-Armed Bandits algorithms

Every time \( t \in \mathbb{N} \) a dynamic device needs to send:
1. it chooses a channel \( A(t) \in \{1, \ldots, K\} \)
2. it sends an uplink packet \( u \) on that channel
3. it observes a binary reward \( r(\{k\}) = \{0, 1\} \)
   (1 if \( A \) is well received, 0 if collision)

4.1. Upper Confidence Bound algo.

Simple frequentist approach :
- Selections of channel \( k \), up-to time \( t \):
  \( N_k(t) := \sum_{\tau \leq t} \mathbb{1}(A(\tau) = k) \)
- Accumulated rewards:
  \( X_k := \sum_{\tau \leq t} r(\tau, k) = \sum_{\tau \leq t} \mathbb{1}(A(\tau) = k) \)
- UCB\(_t\) uses a confidence term (parameter \( \alpha > 0 \))
  \[ B_t(k) := \sqrt{\alpha \log(t)/N_k(t)} \]
- To compute its index (upper confidence bound)
  \[ U(t) := X_k(t)/N_k(t) + B_t(k) = \frac{\mu_k(t)}{N_k(t)} + B_t(k) \]
- Use \( U(t) \) to decide the channel for next step:
  \( A(t + 1) = \arg\max_{k \leq K} U_k(t) \)
\( \implies \) UCB\(_t\) is a deterministic index policy.

4.2. Thompson Sampling algorithm

Old algorithm (1935), Bayesian approach :
- Start with a flat Beta prior, Beta(1, 1), on the (unknown) parameter \( \mu_k \in [0, 1] \)
- And at time \( t \), the posterior counts the successes and failures of channel \( k \):
  \( \Pi_k(t) = \beta(1 + X_k(t), 1 + N_k(t) - X_k(t)) \)
- Then sample a random index for each channel, from the posteriors:
  \( I_k(t) \sim \Pi_k(t) \)
- And choose:
  \( A(t + 1) = \arg\max_{k \leq K} I_k(t) \)
\( \implies \) TS is a randomized index policy.

5. Quick convergence of MAB algorithms

Figure 1: Time-frequency slotted protocol.
Frame = its 2017 conference in May 2017. \( \implies \) (end-devices transmit their packets) + Ack delay + downlink slot \( \checkmark \) (base station replies with Ack if packet well received).

Figure 2: Performance of 2 MAB algorithms, compared to baseline algorithms (naive or optimal), when the proportion of dynamic end-devices in the network increases, for 10%, 30%, 50% and to 100% (limit scenario).
\( \implies \) Almost optimal performances! \( \implies \) Very quick convergence!

Figure 3: Learning with UCB\(_t\) and TS, with more and more dynamic devices. \( \implies \) For any configuration, TS converges quickly to near optimal performances!

6. Near optimal performances

Figure 4: Gain compared to random channel selection.

7. Conclusions

Our approach is simple to set up: every dynamic object runs a simple on-line Multi-Armed Bandit algorithm to learn the quality of each channel, and aim at the most available channel.
- Economic: low runtime complexity, low memory requirements
- In a fully decentralized manner, dynamic devices learn to fit in the channels almost optimally!
- Convergence is very quick to attain: about 50 communications for each device is enough !
- Surprising result: stochastic MAB algorithms also work very well in non-stochastic environments !
\( \implies \) With lots of dynamic objects in a IoT network, using MAB learning helps to improve the successful transmission rate, and increase quality of service.

8. Main references


9. Thanks to...

- ADDI association for the “PhD Students Day” 2017!
- Magnet Intro team for Workshop on Decentralized ML 2017 !
- CentraleSupélec, BTR, Univ. Rennes 1, Inria (Lille) & CNRS
- Our advisors: Christophe Moy, Emilie Kaufmann & J. Palicot