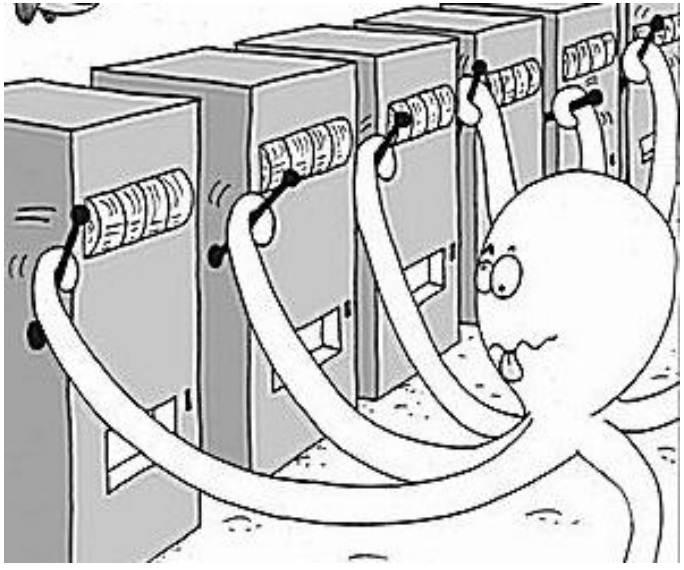
Multi-armed bandits and applications to Blackjack

Seminar in Statistics

Elena Asoni and Valentin Stalder

March 7, 2016

Summary



Problem

You are faced repeatedly with a choice among n different options. After each choice you receive a numerical reward. Your objective is to maximize the expected total reward.

- Slot machine are known as one-armed bandits, because they were originally operated by one lever on the side of the machine.
- A gambler strategically operating multiple machines in order to draw the highest possible profits is called a multi-armed bandit.

n-armed bandit problem

- The reward for each one-armed bandit has a different distribution, therefore a different expectation.
- If the expectations were known, it would be trivial to solve the n-armed bandit problem: select the action with the highest mean reward.
- This is why we assume that we don't know the expected rewards, although we might have estimates.

Exploring and exploiting problem

- At any time step there is at least one action whose estimated expected reward is greatest. We call this a greedy action.
- If you select a greedy action, we say that you are **exploiting** your current knowledge of the values of the actions.
- If instead you select one of the non-greedy actions, then we say you are **exploring**, because this enables you to improve your estimate of the non-greedy action's expected reward.

One-armed bandit

We denote the true mean reward of an action a as $q(a)$, and the estimated mean reward on the t-th time step as $Q_t(a)$.

Estimator for the mean reward (sample-average)

If by the t-th time step action a has been chosen $N_t(a)$ times prior to t, yielding rewards $R_1, R_2, \ldots, R_{N_t(a)}$, then its value is estimated to be

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{N_t(a)}}{N_t(a)} \tag{1}$$

Law of large numbers

As $N_t(a) \to \infty$, $Q_t(a)$ converges to $q(a)$.

Action-selection rules

- The greedy action selection method:

$$A_t = \underset{a}{\text{argmax}} Q_t(a) \tag{2}$$

- The ϵ-greedy methods: behave greedily most of the time, but every once in a while, say with small probability ϵ, select randomly from amongst all the actions with equal probability independently of the action mean rewards estimates (advantage: as $N_t(a) \to \infty$, we ensure that $Q_t(a)$ converge to $q(a)$).

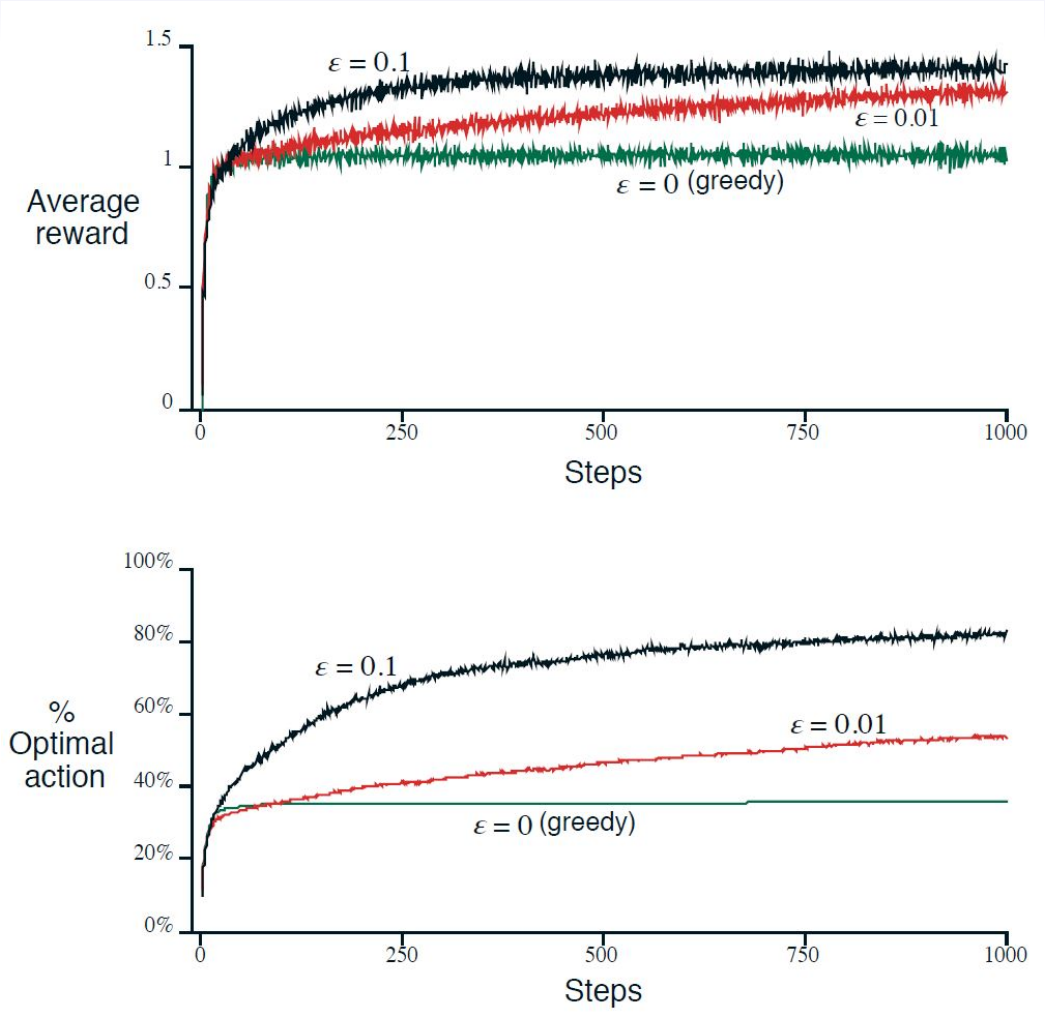


Figure: Average performance of ϵ-greedy action-value methods on the 10-armed testbed. These data are averages over 2000 tasks.

Incremental Implementation

- So far we estimated the action mean rewards as sample averages of observed rewards.
- Problem: the memory and computational requirements for the implementation grow over time without bound.
- Solution: let Q_k denote the estimate for the k-th reward, that is the average of its first $k-1$ rewards, and a k-th reward for the action, R_k. Then:

$$Q_{k+1} = Q_k + \frac{1}{k}[R_k - Q_k] \tag{3}$$

- General form:

$$\textit{New Estimate} \leftarrow \textit{OldEstimate} + \textit{StepSize}[\textit{Target} - \textit{OldEstimate}]$$

Non-stationary Problem

- So far: stationary environment; the bandit is not changing over time.
- In practice, this hypothesis is very often violated or impossible to verify, and we encounter non-stationarity.
- In such cases weight recent rewards more heavily than long-past ones.
- $$Q_{k+1} = Q_k + \alpha[R_k - Q_k] \tag{4}$$

  where the step-size parameter $\alpha \in (0, 1]$ is constant.

Non-stationary Problem - 2

- Weighted average of the past rewards and of the initial estimate Q_1:

$$Q_{k+1} = (1-\alpha)^k Q_1 + \sum_{i=1}^{k} \alpha(1-\alpha)^{k-i} R_i \tag{5}$$

- The quantity $1-\alpha$ is less than 1, and thus the weight given to R_i decreases as the number of intervening rewards increases.

Upper-Confidence-Bound Action Selection

- In the ϵ-greedy method, we choose another action (that is not the greedy one), with probability ϵ.
- Better would be to select among the non-greedy actions the one which has the highest probability to be the greedy action (with the biggest expectation of reward):

$$A_t = \underset{a}{\text{argmax}} \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right] \tag{6}$$

where $c > 0$ controls the degree of exploration. If $N_t(a) = 0$, then a is considered to be a maximising action.

- Set $\hat{\mu}_1 = \cdots = \hat{\mu}_k = 0$, $n_1 = \cdots = n_k = 0$
- For $t = 1 : T$:
  - For each arm i calculate $UCB(i) = \hat{\mu}_i + 2\sqrt{\frac{\ln t}{n_i}}$
  - Pick arm $j = \underset{i}{\text{argmax}} UCB(i)$ and observe y_t
  - Set $n_j \leftarrow n_j + 1$ and $\hat{\mu}_j \leftarrow \hat{\mu}_j + \frac{1}{n_j}(y_t - \hat{\mu}_j)$.

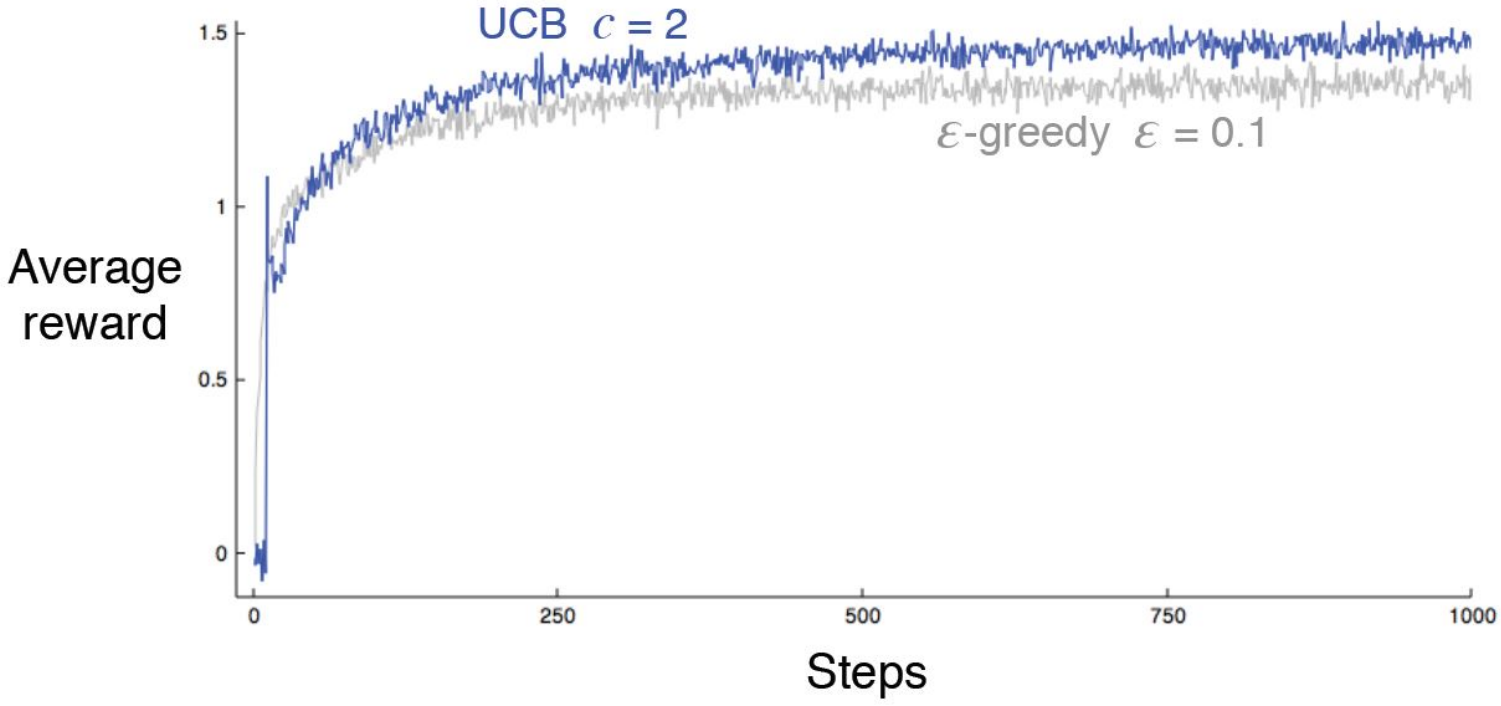


Figure: Average performance of UCB action selection on the 10-armed testbed.

Gradient Bandits

- So far, we considered methods that estimate the mean reward of an action and use those estimators to select an action.
- Here we consider learning a numerical *preference* $H_t(a)$ for each action a. The larger the preference, the more often that action is taken, but the preference has no interpretation in terms of reward:

$$\mathbb{P}[A_t = a] = \frac{e^{H_t(a)}}{\sum_{b=1}^{n} e^{H_t(b)}} = \pi_t(a). \tag{7}$$

- Initially all preferences are the same (e.g. $H_1(a) = 0$, $\forall a$).

Gradient Bandits - 2

- On each step, after selecting the action A_t and receiving reward R_t, the preferences are updated by:

$$H_{t+1}(A_t) = H_t(A_t) + \alpha(R_t - \bar{R}_t)(1 - \pi_t(A_t)), \text{and}$$
$$H_{t+1}(a) = H_t(a) - \alpha(R_t - \bar{R}_t)\pi_t(a), \ \forall a \neq A_t \quad (8)$$

where $\alpha > 0$ is a step-size parameter, and $\bar{R}_t \in \mathbb{R}$ is the average of all the rewards up through and including t.

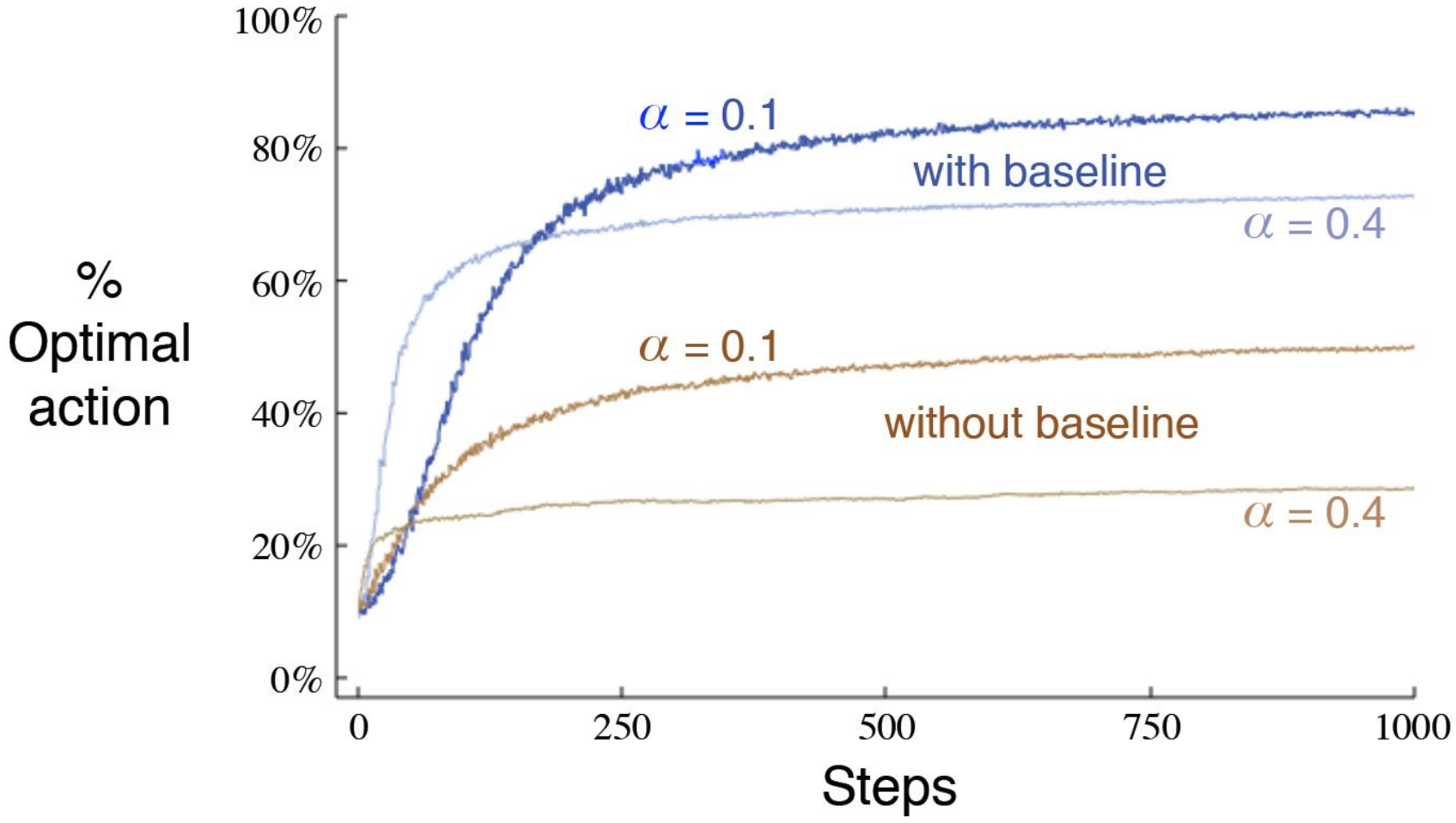


Figure: Average performance of the gradient-bandit algorithm on the 10-armed testbed.

- There are three fundamental formalizations of the bandit problem depending on the assumed nature of the reward process:

  stochastic, adversial and Markovian.

- We have already seen the idea of the stochastic bandit problem. Now we define it more formally.

The stochastic bandit problem

Set-up

Given $K \geq 2$ arms and sequences $X_{i,1}, X_{i,2}, \ldots$ of unknown rewards associated with each arm $i = 1, \ldots, K$, we study forecasters that at each time step $t = 1, 2, \ldots$ select an arm I_t and receive the associated reward $X_{I_t,t}$.

The stochastic bandit problem

Known parameters: number of arms K and of rounds $n \geq K$.
Unknown parameters: K probability distributions $\nu_1, \ldots, \nu_K$ on $[0, 1]$.
For each round $t = 1, 2, \ldots$

- the forecaster chooses $I_t \in \{1, \ldots, K\}$;
- given I_t, the environment draws the reward $X_{I_t,t} \sim \nu_{I_t}$ independently from the past and reveals it to the forecaster.

- The goal is the same than before: maximize the reward.
- Analogously to this is to minimize the regret.
- The regret is what we lose by not playing the optimal strategy:

$$R_n = \max_{i=1,\dots,K} \sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}. \tag{9}$$

- Expected regret:

$$\mathbb{E}[R_n] = \mathbb{E}\left[\max_{i=1,\ldots,K} \sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}\right] \tag{10}$$

- Pseudo-regret:

$$\bar{R}_n = \max_{i=1,\ldots,K} \mathbb{E}[\sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}] \tag{11}$$

- Note: The pseudo-regret is a weaker notion of regret, since one competes against the action which is optimal only in expectation. More formally:

$$\bar{R}_n \leq \mathbb{E}[R_n] \tag{12}$$

- In the stochastic setting, the pseudo-regret can be written as

$$\bar{R}_n = n \max_{i=1,\dots,k} \mu_i - \sum_{t=1}^{n} \mathbb{E}[\mu_{I_t}] \tag{13}$$

Thompson Sampling

- In one of the earliest works on stochastic bandit problems, Thompson proposed a randomized Bayesian algorithm to minimize regret.
- Basic idea: assume a simple prior distribution on the parameters of the reward distribution of every arm, and at any time step, play an arm according to its posterior probability of being the best arm.

Thompson Sampling - 2

- We assume that the reward $\mu_i \in [0, 1]$ have an initial distribution, and $\pi_{i,t}$ be the posterior distribution for μ_i at time t. The reward at time t of arm i is $\theta_{i,t} \sim \pi_{i,t}$, where $\theta_{i,t}$ are independent. The strategy is implemented by sampling from the posterior.
- The strategy is then given by $\underset{i=1,\ldots,K}{\operatorname{argmax}}\, \theta_{i,t}$.

Thompson Sampling for the Bernoulli bandit problem

- In this case, the rewards are either 0 or 1, and for arm i, the probability of success is μ_i.
- Beta distribution turns out to be a very convenient choice of priors for Bernoulli rewards:

$$\left(f(x;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}\right).$$

- Why? Because if the prior is a Beta(α,β) distribution, then after observing a Bernoulli trial,the posterior distribution is simply Beta$(\alpha+1,\beta)$ or Beta$(\alpha,\beta+1)$ depending on whether the trial resulted in a success or failure.

Thompson Sampling for Bernoulli bandit - Algorithm

The Thompson Sampling algorithm initially assumes arm i to have prior Beta$(1,1)$ on μ_i, which is natural because Beta$(1,1)$ is the uniform distribution on $(0,1)$. For each arm $i = 1, \ldots, N$ set $S_i = 0$, $F_i = 0$. For each $t = 1, 2, \ldots$ do:

- For each arm $i = 1, \ldots, N$, sample $\mu_i(t)$ from the Beta$(S_i + 1, F_i + 1)$ distribution.
- Play arm $i(t) = \text{armax}\mu_i(t)$.
- If $r = 1$, $S_{i(t)} = S_{i(t)} + 1$ else $F_{i(t)} = F_{i(t)} + 1$

The adversial bandit problem

- Suppose we are in a rigged casino.
- The owner/adversary sets the gain $X_{i,t}$ to some arbitrary value $g_{i,t} \in [0, 1]$.
- The owner is called oblivious if his choice does not depend on the strategy of the player.
- It's called non-oblivious if his choice depends on the strategy of the player, which means:

$$g_{i,t} = g_{i,t}(I_1, \ldots, I_{t-1}) \tag{14}$$

The adversial bandit problem - 2

Known parameters: number of arms $K \geq 2$ and (possibly) number of rounds $n \geq K$. For each round $t = 1, 2, \ldots$

(1) the forecaster chooses $I_t \in \{1, \ldots, K\}$, possibly with the help of external randomization,

(2) simultaneously, the adversary selects a gain vector $g_t = (g_{1,t}, \ldots, g_{K,t}) \in [0, 1]^K$, possibly with the help of external randomization,and

(3) the forecaster receives (and observes) the reward $g_{I_t,t}$ while the gains of the other arms are not observed.

The adversial bandit problem - 3

- For the adversial bandit problem, the pseudo-regret can be defined as

$$\bar{R}_n = \max_{i=1,\ldots,K} \mathbb{E}\left[\sum_{t=1}^{n} g_{i,t} - \sum_{t=1}^{n} g_{I_t,t}\right] \tag{15}$$

- Note: The adversial gains $g_{i,t}(I_1, \ldots, I_{t-1})$ could be different than those chosen by the player with the optimal tactic.

Markovian bandit problem

- In the Markovian bandit problem, the reward processes are neither i.i.d. nor adversial.
- Arms are associated with K Markov processes, each with its own state space.
- Each time an arm i is chosen in state s, a stochastic reward is drawn from a probability distribution $\nu_{i,s}$, and the state of the reward process for arm i changes in a Markovian fashion, based on an underlying stochastic transition matrix M_i.

Contextual bandit problem

- So far: only non associative tasks, in which there is no need to associate different actions with different situations. In these tasks the learner either tries to find a single best action when the task is stationary, or tries to track the best action as it changes over time when the task is non stationary.
- Now: in the associative search, we act differently according to the situation we are in, before taking an action.
- Example: you are confronted with several n-armed bandits chosen at random and your only clue about them is, for example, their colour.

News recommendation example

- Lets see an application of the contextual bandit problem ☺
- Web services try to adapt their services (advertisement, news, articles, ...) to individual users by making use of user information.
- This can be seen as a multi-armed bandit problem where we have additional information about the slot machine.
- One click of the user on the advertisement or new correspond to the reward.
- The action corresponds to the choice of the advertisement or new which we think the user will most likely click (so for which we would most likely got the reward).

Example of a contextual bandit problem

- We have the choice between four links and we have to decide which one we highlight on the yahoo front page, with the aim that the user clicks on it. We have also information about the user, and therefore highlight an article according to his interests.
- This problem can be seen as a contextual bandit problem, where the reward is a click on the link we have highlighted.

Figure 1: A snapshot of the "Featured" tab in the Today Module on Yahoo! Front Page. By default, the article at F1 position is highlighted at the story position.

Contextual bandit - algorithm

- A contextual bandit algorithm A proceeds in discrete trials $t = 1, 2, 3, \ldots$. In trial t:

1. The algorithm observes the current user u_t and a set $\mathcal{A}_t$ of arms or actions together with their feature vectors $x_{t,a}$ for $a \in \mathcal{A}$. The vector $x_{t,a}$ summarizes information of both the user u_t and arm a, and will be referred as the *context*.
2. Based on observed payoffs in previous trials, A chooses an arm $a_t \in \mathcal{A}_t$, and receives payoff r_{t,a_t} whose expectation depends on both the user u_t and the arm a_t.
3. The algorithm then improves its arm-selection strategy with the new observation $(x_{t,a}, a_t, r_{t,a_t})$.

T-trial regret R_A

- The total T-trial payoff of A is defined as $\sum_{t=1}^{T} r_{t,a_t}$.
- Define the optimal expected T-trial payoff as $\mathbb{E}[\sum_{t=1}^{T} r_{t,a_t^*}]$, where a_t^* is the arm with maximum expected payoff at trial t.
- Goal: write an algorithm A so that the expected payoff is maximized.
- As we have already seen this is equivalent to find an algorithm which minimizes the regret with respect to the optimal arm-selection strategy:

$$R_A(T) = \mathbb{E}[\sum_{t=1}^{T} r_{t,a_t^*}] - \mathbb{E}[\sum_{t=1}^{T} r_{t,a_t}]. \tag{16}$$

A special case of the contextual bandit problem

A special case of the contextual bandit problem is the n-armed bandit problem, in which:

- The arm set $\mathcal{A}$ remains unchanged and contains n arms $\forall t$.
- The situation s_t is the same $\forall t$.

The n-armed bandit problem is also called the context free bandit problem. While the context free bandit problems are extensively studied and well understood, the contextual bandit problem has remained challenging.

- We want to find an algorithm that solves the contextual bandit problem.
- Idea: use the UCB method

Linear Upper Bound Confidence - 1

- We assume the expected payoff of an arm a is linear in its d-dimensional feature $x_{t,a}$ with some unknown coefficient vector θ_a^*, namely, $\forall t$,

$$r_{t,a} = x_{t,a}^T \theta_a^* + \epsilon_t \tag{17}$$

  which is equivalent to $\mathbb{E}[r_{t,a} \mid x_{t,a}] = x_{t,a}^T \theta_a^*$. The parameters are not shared among different arms.
- We wish to minimize the square loss

$$\hat{\theta}_a = \underset{\theta_a^*}{\operatorname{argmin}} \sum_{t=1}^{m} (r_{t,a} - x_{t,a}^T \theta_a^*)^2 \tag{18}$$

Linear Upper Bound Confidence - 2

- D_a design matrix of dimension $m \times d$ at trial t, $b_a \in \mathbb{R}^m$ response vector.
- Applying ridge regression to the training data (D_a, b_a) gives $\hat{\theta}_a = (D_a^T D_a + I_a)^{-1} D_a^T b_a$, where I_d is the $d \times d$ identity matrix.
- Theorem: for the estimated coefficients it holds that

$$| x_{t,a}^T \hat{\theta}_a - \mathbb{E}[r_{t,a} \mid x_{t,a}] | \leq \alpha \sqrt{x_{t,a}^T (D_a^T D_a + I_d)^{-1} x_{t,a}} \quad (19)$$

for any $\delta > 0$ and $x_{t,a} \in \mathbb{R}^d$, where $\alpha = 1 + \sqrt{\frac{\ln(2/\delta)}{2}}$.

Linear Upper Bound Confidence - 3

- The inequality gives a reasonably tight UCB for the expected payoff of arm. At each trial t, choose:

$$a_t = \underset{a \in \mathcal{A}_t}{\text{argmax}}(x_{t,a}^T \hat{\theta}_a + \alpha \sqrt{x_{t,a}^T A_a^{-1} x_{t,a}}), \tag{20}$$

where $A_a = D_a^T D_a + I_d$.

LinUCB algorithm

Algorithm 1 LinUCB with disjoint linear models.

Inputs: $\alpha \in \mathbb{R}_+$
for $t = 1, 2, 3, \ldots, T$ **do**
  Observe features of all arms $a \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
  for all $a \in \mathcal{A}_t$ **do**
    if a is new **then**
      $\mathbf{A}_a \leftarrow \mathbf{I}_d$ (d-dimensional identity matrix)
      $\mathbf{b}_a \leftarrow \mathbf{0}_{d \times 1}$ (d-dimensional zero vector)
    end if
    $\hat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1} \mathbf{b}_a$
    $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}}$
  end for
  Choose arm $a_t = \arg\max_{a \in \mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_t
  $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t} \mathbf{x}_{t,a_t}^\top$
  $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t \mathbf{x}_{t,a_t}$
end for

Summary

Application to Blackjack

Setting: we have two choices

- In the first one we have to decide between n actions, e.g. hit, stand or double down. We ignore splitting.
- In the second one we always stand.

The first decision can be seen as a n-armed bandit problem. Our example can be seen as a 3-armed bandit problem, since we have the choice between 3 actions (hit, stand and double down). They have a different reward distribution and so a different reward expectation.
Goal: Maximize the reward.

Application to Blackjack - greedy methods

We have seen some methods to solve the n-armed bandit problem, let's apply them to our 3-armed bandit problem.

- The greedy method: choose at random e.g. $t = 1000$ times one of this 3 actions and then make your estimator for the expectation of the reward:

  $$Q(a) = \frac{R_1 + \cdots + R_{N_t(a)}}{N_t(a)}$$

  $N(a) =$ times we have chosen the action a until time t. Then exploit the greedy actions, i.e. choose always the action with the greatest estimator expectation.
- The ϵ greedy method: choose with with probability ϵ an action which is not the greedy action.

Application to Blackjack - other methods

- In the Upper-Confidence-Bound Action Selection Method we use the nice formula:

$$A_t = \underset{a}{\text{argmax}}[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}].$$

- The Thompson sampling methods if we adapt the reward such that they are bounded in $[0, 1]$, etc.

Application to Blackjack - Contextual bandit problem

- There is another way to approach this problem: we can see it as a contextual bandit problem.
- The idea is to look at every possible configuration of cards the dealer and the player have and what actions were the best in the different situations, i.e. we implement with lots of single bandits, without information sharing.
- So as in the contextual bandit problem we have to associate different actions with different situations (configuration of cards).

Application to Blackjack - UCB

- A way to solve the contextual bandit problem is to use the already seen linear UCB algorithm:
- Feature vector $x_{t,a}$ is the information we have on the players and dealers cards.
- $\mathbb{E}[r_{t,a} \mid x_{t,a}]$ is the expectation of the reward given $x_{t,a}$. Goal: maximize it.
- The rows of the design matrix D_a correspond to the different combinations of cards have.