1 Preliminaries

In this short section we present the purpose of the field of Reinforcement Learning and introduce the fundamental concepts associated with it. A simple but broad characterization of RL is the following: ***Reinforcement Learning is a set of computational methods designed to automate learning in unknown environments where the emphasis is on making decisions to achieve predefined goals***. The core paradigm is that the system learns through direct interactions with the environment (i.e. there is continual feedback loop between the agent and the environment) and does not necessarily require any outside supervision. Since the only signal the agent receives comes from the interactions with the environment, there is constant conflict between two aspects of these interactions: ***exploration*** – trying different actions to see the outcome and ***exploitation*** – using the actions that yielded the best results in the past. The fundamental concepts of reinforcement learning are:

- ***agent*** – an entity that can: ***observe*** (at least partially) the environment, ***perform actions*** that change itself and the environment and ***receive rewards*** in connection with the interactions with the environment
- ***environment*** – the reality outside of an agent
- ***policy / control*** – defining the behavior of the agent

2 Multi-armed bandits

Our first example of an RL system will be the well-known problem of k-armed bandits. In this problem we have a finite set of k possible actions $\mathcal{A}$ and a conditional probability distribution of the rewards r given the chosen action a i.e. $p(r \mid a)$. This probability distribution is obviously not known a priori and thus our goal is to find a policy which maximizes our expected reward over some finite number of steps T or in other words we want a policy that learns as quickly as possible a very good approximations of the true conditional expected values $\mathbb{E}[r \mid a]$. Note that if we knew the probability distributions ahead of time (or even just their expectations) then in each step we should just choose the action a^* such that

$$a^* = \arg\max_{a\in\mathcal{A}} \mathbb{E}[r \mid a]$$

since

$$\mathbb{E}_{r_1\sim p(r|a_1),\ldots,r_T\sim p(r|a_T)}\left[\sum_{t=1}^{T} r_t\right] = \sum_{t=1}^{T}\mathbb{E}[r \mid a_t]\,.$$

2.1 Value function methods

One of two classes of methods used in the k-armed bandits problem are based on the so called ***value function*** which really is just an estimator of the conditional expected reward given the action a, namely

$$q_t(a) = \frac{1}{n_t(a)}\sum_{i=1}^{t-1} r_i\delta_{a_i,a}\,, \quad n_t(a) = \sum_{i=1}^{t-1}\delta_{a_i,a}\,,$$

where $a_i \in \mathcal{A}$ is the action chosen at step i, $n_t(a)$ is the number of times action a was chosen up until timestep t and δ denotes the Kronecker's delta. Let's note that these expressions satisfy the following recursive equations

$$\boxed{\begin{aligned} n_{t+1}(a) &= n_t(a) + \delta_{a_t,a} \\ q_{t+1}(a) &= q_t(a) + \frac{\delta_{a_t,a}}{n_{t+1}(a)}\big[r_t - q_t(a)\big]\end{aligned}}$$

Based on the definition of value function and the recursive equations for the update we can devise several methods of choosing the action at given step which differ by the way they balance exploration and exploitation.

2.1.1 ε-Greedy

Given the value function at step t, the simplest possible choice of action is the greedy one i.e.

$$a_t = \arg\max_{a\in\mathcal{A}} q_t(a)\,.$$

This approach is purely exploitative and thus as expected it doesn't perform very well. A simple modification of this method which allows for any exploration is the ϵ-greedy choice where with probability ϵ we randomly choose any action from $\mathcal{A}$ and with probability $1-\epsilon$ greedily choose the one with maximal value function. The value of ϵ is chosen beforehand and can have a big impact on the overall performance of this algorithm. Of course if $\epsilon = 0$ then we just go back to the greedy choice of action.

Two possible modifications of this simple algorithm can make it perform better. First of all if we look at the equation for the value function update we can generalize it a bit and write it as

$$q_{t+1}(a) = q_t(a) + \delta_{a_t,a}\alpha_t(a)\big[r_t(a) - q_t(a)\big]\,,$$

where $\alpha_t(a)$ is called step-size. We can thus choose step-size beforehand and update the value function according to this equation instead of the one before. Additionally we can use the so called ***optimistic initialization*** i.e. choose suitable values of $q_0(a)$ different than 0 (which is on the contrary called the ***realistic initialization***). This makes the algorithm prefer exploration in the initial phase.

2.1.2 Upper Confidence Bound

Another approach is based on a somewhat lesser known result from the probability theory, namely the ***Hoeffding's inequality*** which states that for any i.i.d. sequence of random variables $X_1, \dots, X_n$ with expected value $\mathbb{E}[X]$ and any $u \geq 0$ we have

$$\Pr\{\mathbb{E}[X] > \overline{X}_n + u\} \leq \mathrm{e}^{-2nu^2} \,,$$

where $\overline{X}_n = \dfrac{1}{n}\sum_{i=1}^{n} X_i$. The idea is to introduce for any action a, the upper confidence bound $u_t(a)$ on the value function $q_t(a)$, since from the Hoeffding's inequality we have

$$\Pr\{\mathbb{E}[r \mid a] > q_t(a) + u_t(a)\} \leq \mathrm{e}^{-2n_t(a)u_t(a)^2}$$

and assuming that we want this probability to be less then some given p we have

$$u_t(a) = \sqrt{\frac{-\ln p}{2n_t(a)}} \,.$$

A good heuristic in practice for choosing the value of p is assuming that it changes with the number of time steps according to $p \propto t^{-4}$ so that the upper bound can be written as

$$\boxed{u_t(a) = c\sqrt{\frac{\ln t}{n_t(a)}}}$$

where c is a hyper-parameter. We then choose the action according to

$$\boxed{a_t = \operatorname*{arg\,max}_{a \in \mathcal{A}} \left[q_t(a) + c\sqrt{\frac{\ln t}{n_t(a)}} \right]}$$

2.2 Gradient based methods

The two previous methods for choosing actions in the k-armed bandits problem used the value function $q_t(a)$. This is however not the only option to choose an action. We can for example introduce a ***preference function*** $h_t(a)$ which qualitatively describes how good an action a is at timestep t. The question however appears – how to choose an action based on the preference function. A good answer is to introduce the probability distribution over actions based on the preference function through the softmax function i.e. the probability of choosing action a at step t is given by

$$\boxed{\pi_t(a) = \frac{\exp h_t(a)}{\sum_{a' \in \mathcal{A}} \exp h_t(a')}}$$

and choose the action by sampling from the categorical distribution described by the probabilities $\pi_t(a)$. How do we update the preference function?

2.3 Thompson sampling

We will assume that the conditional probability $p(r \mid a)$ comes from some parametric family of probability distribution and is parametrized by the parameters θ_a, namely

$$p(r \mid a) = p(r \mid \theta_a) \,.$$

Thompson sampling algorithm is based on a simple Bayesian update rule, namely let $\pi_t(\theta_a)$ be the prior distribution of the parameters θ_a at timestep t, then using the Bayes theorem we have

$$p(\theta_a \mid r_t) \propto p(r_t \mid \theta_a)\pi_t(\theta_a)$$

and can update the prior as

$$\pi_{t+1}(\theta_a) = \frac{1}{Z(r_t)} p(r_t \mid \theta_a)\pi_t(\theta_a) \,,$$

where

$$Z(r_t) = \int p(r_t \mid \theta_a)\pi_t(\theta_a)\,\mathrm{d}\theta_a$$

is the normalization constant. The action choice is then as follows, we ***sample*** parameter values θ_a for every action $a \in \mathcal{A}$ using distributions $\pi_t(\theta_a)$ and obtain samples

$$\{\underline{\theta_a} \mid a \in \mathcal{A}\}$$

and choose the action maximizing the conditional expected reward namely

$$a_t = \operatorname*{arg\,max}_{a \in \mathcal{A}} \mathbb{E}_{r \sim p(r \mid \underline{\theta_a})}[r]$$

We apply the action, obtain the reward and update the corresponding prior distribution of the parameters θ_{a_t}.

In the actual applications we often use statistical models $p(r \mid \theta_a)$ for which there exist so called conjugate priors i.e. priors which result in posterior from the same parametric family as the prior (***not*** the likelihood $p(r \mid \theta_a)$) but with modified parameters since this greatly simplifies the required computation. Below we describe an example adaptation of the Thompson sampling algorithm to the Bernoulli bandits problem.

2.3.1 Bernoulli bandits

Let's assume that each bandit returns as a reward either 0 or 1 (e.g. we have a recommender system and our feedback is whether user clicked a given link, ad, etc.) i.e. $r \in \{0, 1\}$. We will model conditional distributions $p(r \mid a)$ using Bernoulli distribution i.e.

$$p(r \mid a) = \mathrm{Ber}(r \mid \theta_a) = \theta_a^r (1-\theta_a)^{(1-r)}\,,$$

where $\theta_a \in [0; 1]$ is the probability of getting 1. Let's assume that the prior distribution $\pi(\theta_a)$ is given by a beta distribution with some parameters α_a, β_a

$$\pi(\theta_a) = \mathrm{Beta}(\theta_a; \alpha_a, \beta_a) \propto \theta_a^{\alpha_a - 1}(1-\theta_a)^{\beta_a - 1}\,.$$

Let's note that the posterior is given by

$$\begin{aligned} p(\theta_a \mid r) &\propto \theta_a^r (1-\theta_a)^{(1-r)} \cdot \theta_a^{\alpha_a - 1}(1-\theta_a)^{\beta_a - 1} \\ &\propto \theta_a^{\alpha_a + r - 1}(1-\theta_a)^{\beta_a + (1-r) - 1} \end{aligned}$$

and thus is just the beta distribution with parameters $\alpha_a + r$ and $\beta_a + (1-r)$. We therefore see that beta distribution is the conjugate prior to the Bernoulli likelihood. Thompson sampling algorithm for the Bernoulli bandits is thus following. Let $\alpha_t^{(a)}$, $\beta_t^{(a)}$ be the parameters of the prior distribution $\pi_t(\theta_a)$ at timestep t (values at $t = 1$ are hyper-parameters), obtain samples $\underline{\theta_a}$ from $\mathrm{Beta}(\alpha_t^{(a)}, \beta_t^{(a)})$ for each action $a \in \mathcal{A}$, perform action a_t chosen as

$$a_t = \operatorname*{arg\,max}_{a \in \mathcal{A}} \mathbb{E}_{r \sim \mathrm{Ber}(\underline{\theta_a})}[r] = \operatorname*{arg\,max}_{a \in \mathcal{A}} \underline{\theta_a}$$

and observe reward r_t, finally update the prior distributions by updating the parameters according to

$$\begin{aligned} \alpha_{t+1}^{(a)} &= \alpha_t^{(a)} + \delta_{a_t, a} r_t \\ \beta_{t+1}^{(a)} &= \beta_t^{(a)} + \delta_{a_t, a}(1 - r_t) \end{aligned}$$

3 Dynamic Programming

4 Monte Carlo methods