Decision Making and Bayesian optimisation

Martin Dimitrov

1 Introduction and Motivation

1.1 Statistical decision theory

The main idea behind learning algorithms is to generalise past observations to make future predictions. Of course, given a set of observations, one can always find a function to exactly fit the observed data. However, without placing any restrictions on how future data is related to past, it would be impossible to construct a generalised algorithm. In other words, data alone cannot replace knowledge. Hence to proceed forward, we can try to follow the "rule":

$$Generalisation = Data + Knowledge \tag{1.1}$$

1.2 Bayesian optimisation

Big data applications are typically associated with systems involving a large numbers of tunable parameters. If optimized jointly, these parameters can result in significant improvements. Bayesian optimization is a powerful tool for the joint optimization of design choices that is gaining great popularity in recent years. It promises greater automation so as to increase both product quality and human productivity.

2 Choosing a restaurant

2.1 Theory background - Bayesian decision making

Suppose we have a set A of all possible actions (denotes by a_i, i.e $a_i \in A$). To picture this, think of slot machines in a casino. You can decide which slot machine to use to maximise your profit for example, each machine representing a different action. Once you play your first round, you obtain a result. You can think of this result as a random variable, hence having its own distribution. If the machines are different, the results we obtain are a samples from each slot machine X_i. Often times we want to update our beliefs of what slot machines produce and to do this we use Bayesian ideas. Hence:

$$X_i \sim p_{X_i}(\cdot|D_i, a_i), \tag{2.1}$$

where D_i and p_{X_i} are the past observed data and the posterior distribution for this particular action.

The thing we are trying to optimise is called the utility function $u(\mathbf{X}, a_i)$ and it generally depends on the available actions and the **state of the world** vector $\mathbf{X}$, which is a random quantity affecting the utility. For example, the utility function can be as simple as profit, but it could be complicated and also unknown to the decision maker.

In general, we are interested in taking the action that maximises the utility function, which can be expressed as:

$$\hat{a} = \underset{a_i \in A}{\arg\max}\ \mathbb{E}_{\mathbf{X} \sim p_{\mathbf{X}}(\cdot|D,a_i)}[u(\mathbf{x}, a_i)]. \tag{2.2}$$

2.2 Restaurant example

Here we are given 4 restaurants (McDonald's, Sultan, Blue Moon, QSF) and observed utility values of [(4,5); (2,6); (5); (3, 5, 6)], i.e. we went twice to McDonald's, twice to Sultan, etc.. We also assume that utility of visiting restaurant a_i is $N(\mu_{a_i}, \sigma^2_{a_i})$.

Lets clarify how the theory described above is applied here. Even though it does not specifically matter what exactly the utility function represents for the final answer, it is important to develop some intuition behind it, in case its "meaning" changes. For example, if we start with a **state of the world** vector, which records how much beans there were in the dishes we tried, and if the customer in particular likes beans, then a utility function of "best experience" might strongly be related to the food ingredients.

2.3 Mathematical reasoning

Since the utility is normally distributed, then the likelihood belongs to the exponential family of functions and we can use conjugacy arguments. In particular, for IID $\mu_{a_i}, \sigma^2_{a_i}$, the priors are given by:

$$p(\mu_{a_i}, \sigma^2_{a_i}) = p_{inverse-gamma}(\sigma^2_{a_i}|\alpha, \beta) p_{gaussian}(\mu_{a_i}|m, \kappa\sigma^2_{a_i}). \tag{2.3}$$

Therefore, the posteriors are also Gaussian with known parameters, where α, β, m and κ are the hyper-parameters of the priors.

Looking back at equation (2.2) we see that it does not specify a parametric model, hence we will have to average out over the unknown parameters μ_{a_i} and $\sigma^2_{a_i}$. Therefore adjusting equation (2.2) we get:

$$\begin{aligned}\hat{a} &= \underset{a_i \in A}{\arg\max}\ \mathbb{E}_{\mathbf{X}\sim p_{\mathbf{X}}(\cdot|D,a_i)}[u(\mathbf{x}, a_i)|D] = \\ &= \underset{a_i \in A}{\arg\max}\ \mathbb{E}_{\mathbf{X}}\left[\int_{-\infty}^{\infty}\int_{0}^{\infty} u(\mathbf{x}, a_i|\mu_{a_i}, \sigma^2_{a_i}, D) f(\mu_{a_i}, \sigma^2_{a_i}|D) d\sigma^2_{a_i} d\mu_{a_i}\right],\end{aligned} \tag{2.4}$$

where $f(\mu_{a_i}, \sigma^2_{a_i}|D)$ is the posterior distribution. W notice that the double integral is just the expectation of the parameters taken under the posterior. Hence, equation (2.4) can be compactly written as:

$$\hat{a} = \underset{a_i \in A}{\arg\max}\ \mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{(\mu_{a_i}, \sigma^2_{a_i})}[u(\mathbf{x}, a_i)|D]\right]. \tag{2.5}$$

Assuming some conditions, such as the ones in Fibini's theorem, we can interchange the order of the expectations. Also, since $u(\mathbf{x}, a_i) \sim N(\mu_{a_i}, \sigma^2_{a_i})$, then $u(\mathbf{x}, a_i) = \mu_{ai} + \sigma^2_{a_i} Z$, where $Z \sim N(0, 1)$. Therefore:

$$\begin{aligned}\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{(\mu_{a_i}, \sigma^2_{a_i})}[u(\mathbf{x}, a_i)|D]\right] &= \mathbb{E}_{(\mu_{a_i}, \sigma^2_{a_i})}\left[\mathbb{E}_{\mathbf{X}}[u(\mathbf{x}, a_i)|D]\right] \\ = \mathbb{E}_{(\mu_{a_i}, \sigma^2_{a_i})}\left[\mathbb{E}_{\mathbf{X}}[\mu_{a_i} + \sigma^2_{a_i} Z|D]\right] &= \mathbb{E}_{(\mu_{a_i}, \sigma^2_{a_i})}[\mu_{a_i}],\end{aligned} \tag{2.6}$$

where the last expectation is taken under the posterior. We know that:

$$\mathbb{E}_{(\mu_{a_i}, \sigma^2_{a_i})}[\mu_{a_i}] = \frac{m + \kappa n \bar{u}}{1 + \kappa n}, \tag{2.7}$$

where $\bar{u}$ and n are the average of the observed utility values and the number of observations for restaurant a_i respectively. We can summarise this result in the following table:

Restaurant a_i	$\mathbb{E}[\mu_{a_i}]$
McDonalds	$\frac{m+10\kappa}{1+2\kappa}$
Sultans	$\frac{m+8\kappa}{1+2\kappa}$
Blue Moon	$\frac{m+5\kappa}{1+\kappa}$
QSF	$\frac{m+14\kappa}{1+3\kappa}$

After some calculations, it can be seen that when $m > \frac{9-\kappa}{2}$, $\frac{m+5\kappa}{1+\kappa}$ is the largest, and if $m < \frac{9-\kappa}{2}$, $\frac{m+14\kappa}{1+3\kappa}$ is the largest. Since we were interested only in the mean parameter, our choice of restaurant is independent of α and β. This is nicely summarised in Figure 1, where the black line is $m = \frac{9-\kappa}{2}$.

3 Decision making

3.1 Theory Background

In this section we take a look at exploration-exploitation trade-off. Such an algorithm repeatedly makes decisions to maximise the reward - exploitation, but the overall reward is limited to how much knowledge is available, hence sometimes the algorithm decides to explore other options, gaining more knowledge, but not necessarily maximasing the reward at this step.

Mathematically, following the notation in the slides, we are presented with K options, called actions or arms. At each step t, $t \in \{1, \dots, T\}$, we select an action $a_t \in \{1, \dots, K\}$ and receive a reward. Let us assume that the rewards are independent across the taken actions and can be modelled by random variables, i.e.:

$$X_{a_t,t} \sim \nu_k\ i.i.d \quad for\ t \in \{1, \dots, T\}, \tag{3.1}$$

where the ν_k are unknown probability densities.

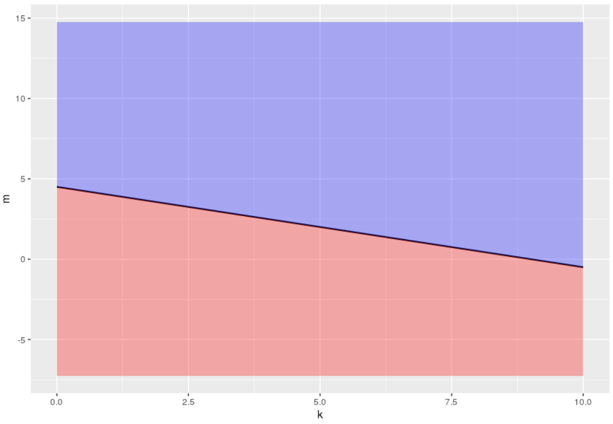


Figure 1: Blue region - “Blue Moon”, red - “QSF”

The goal is to identify a rule of selecting actions, such that it maximises the expected cumulative reward over T steps: $\max \sum_{t=1}^{T} \mathbb{E}(X_{a_t,t})$. Such a rule, which we are going to refer to as policy or algorithm, will be a mapping, π, from the previous data to the available arms. Since the distributions are unknowns, the algorithm needs to do initial exploration, till eventually finds the arm with the highest expected value. Therefore, we define the notion of regret, which would be difference between optimal reward (unattainable due to the exploration) and the overall reward achieved by a particular algorithm. Hence, if $\mu_{a_t} = \mathbb{E}(X_{a_t,t})$, for $a_t \in \{1, \dots, K\}$, and $\mu^* = \max_{a_t \in \{1,\dots,K\}} \mu_{a_t}$, we can define the regret as:

$$Reg_{\pi}(T) = T\mu_* - \sum_{t=1}^{T} \mathbb{E}_{\pi}(\mu_{a_t}). \tag{3.2}$$

From this definition, we can see that the regret will always be positive. We note that minimising the regret is equivalent to maximising the overall reward. Three points need to be made:

- No algorithm can achieve zero regret due to exploration
- It can be shown, see [TLL85], that there exists a constant C_k, such that $\lim_{T\to\infty} \frac{Reg_{\pi}(T)}{\log T} \geq C_k$
- It can easily be shown that fixed period exploration is insufficient to achieve $\mathcal{O}(\log T)$.

3.2 Upper confidence bound - UCB

The algorithm is executed as follows:

1. For each $t \in \{1, \dots, K\}$, choose each arm once, i.e. $a_t = t$
2. For t in $\{K+1, \dots, T\}$:
   (a) For each arm $i \in \{1, \dots, K\}$ calculate:

$$\bar{\mu}_{i,t} = \frac{\sum_{s=1}^{t-1} X_{i,s} \mathbb{1}\{a_s = i\}}{\sum_{s=1}^{t-1} \mathbb{1}\{a_s = i\}} + \sqrt{\frac{2 \log t}{\sum_{s=1}^{t-1} \mathbb{1}\{a_s = i\}}} \tag{3.3}$$

   (b) Select the arm with the largest $\bar{\mu}_{i,t}$

We briefly make a few points: each time an arm is selected, the uncertainty term for this arm decreases. However, if the next arm is different, then $\log t$ increases, but the denominator in the confidence interval stays the same, hence the uncertainty increases. The use of the natural logarithm, suggests that the increases in $\bar{\mu}$ will slow down. All arms will be selected, but the arms that have appeared frequently and the ones with lower estimates, will be selected less often.

3.3 Thompson Sampling

The algorithm goes as follows:

1. For each $t \in \{1, \dots, T\}$:
   (a) For each arm i draw a sample:

$$\bar{\mu}_{i,t} \sim p(\mu_k | \mathbf{X}_{i,1:t-1}) \tag{3.4}$$

   (b) Select the arm with the largest sample $\bar{\mu}_{i,t}$ and update its posterior

3.4 Example for Thompson Sampling and UCB

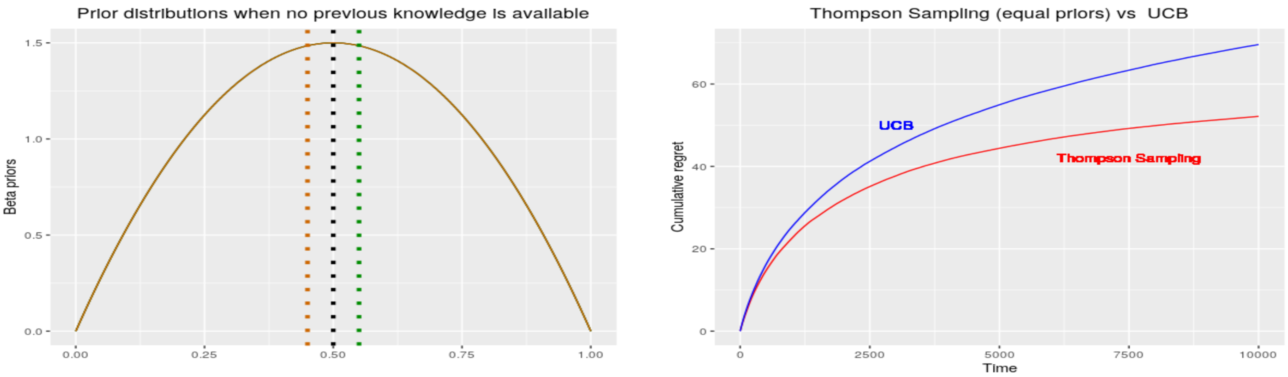
We consider three arms sampled from a Bernoulli distribution with true means: $(0.50, 0.55, 0.45)$. Clearly arm 2 is the best one. We quickly lay down the procedure behind Thompson Sampling.

We are given that $X_{i,t} \sim Bernoulli(\mu_i)$, where μ_i is the true mean of the i_{th} arm, $i \in \{1, \ldots, n\}$. In our case $n = 3$. We have to select a prior distribution in order to find the posterior. As we know, the conjugate prior to the Binomial distribution for arm i is a beta distribution: $Beta(\alpha_i, \beta_i)$, where the hyperparameters (α_i, β_i) need to be chosen in advance. In our first experiment, we will assume no prior knowledge, hence we set: $\alpha_1 = \beta_2 = \alpha_2 = \beta_2 = \alpha_3 = \beta_3 = 2$, see Figure 2a[1]. This leads to posteriors given by:

$$p(\mu_i|\mathbf{X}_{i,1:(t-1)}) \sim Beta\left(2 + \sum_{s=1}^{N} x_{i,s}\mathbb{1}(a_s = i),\ 2 + N - \sum_{s=1}^{N} x_{i,s}\mathbb{1}(a_s = i)\right), \tag{3.5}$$

where N is the number of observed values and a_s takes either $1, 2$ or 3, depending on which arm we sampled from. Since at the start we have no observed data, the posterior parameters will be the same as the prior ones.

We now perform 150 simulations over 10000 rounds and average out the regret at each round. The number of simulations was chosen in a way to reflect accuracy, yet keep the computational time relatively short. After coding the UCB and Thompson Sampling, we arrive at the following regret graph on the right: As we can see, when we have no prior information, the Thompson Sampling finds the best arm faster than UCB, hence leading to less cumulative regret over the rounds.



(a) Equal priors

(b) Total regret

Figure 2: Cumulative regret for Thompson Sampling with equal priors and UCB

We next decide to explore the strength and weaknesses of Thompson Sampling. We consider two additional scenarios:

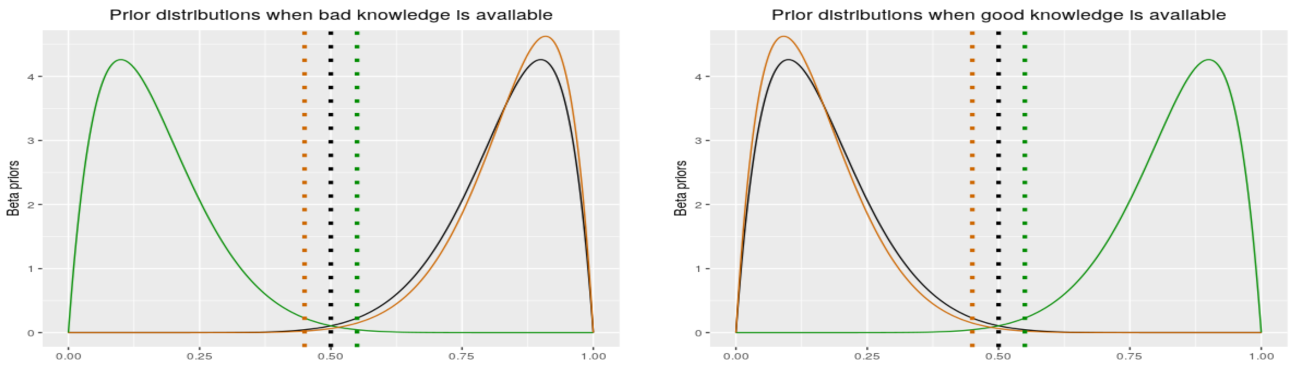
1. We have prior knowledge, but it is wrong, hence giving "weight" to the suboptimal arms
2. We have prior knowledge in favour of the best arm

Implementing the two scenarios basically means choosing proper hyperparameters. This involves some trial and error, but it is easily seen in Firgure 3. In the first scenario, the hyperparameters that we choose are $\alpha_1 = 10,\ \beta_2,\ \alpha_2 = 2,\ \beta_2 = 10,\ \alpha_3 = 11$[2], $\beta_3 = 2$. This offsets the prior distribution of the best arm to the left (green), meaning that in the Thompson Sampling, the probability of choosing it at each round is significantly less than the other two arms. This would imply that the algorithm will take a much larger number of trials till it starts to find the correct arm, hence accumulating more regret, as can be seen in Figure 4 (black).

Here, the black curve represents the total accumulative regret when bad hyperparameters were chosen and in this case, the Thompson sampling with no prior knowledge clearly performs better, but more importantly, UCB as well.

[1] The true means for each arm are given with vertical dotted lines in their respective colour

[2] The reason why α_3 is equal to 11, instead of 10, is simply to avoid overlapping of the two beta distributions



(a) Bad priors:
$(\alpha_1, \beta_2, \alpha_2, \beta_2, \alpha_3, \beta_3) = (10, 2, 2, 10, 11, 2)$

(b) Good priors:
$(\alpha_1, \beta_2, \alpha_2, \beta_2, \alpha_3, \beta_3) = (2, 10, 10, 2, 2, 11)$

Figure 3: Two scenarios, having “bad - good” prior information

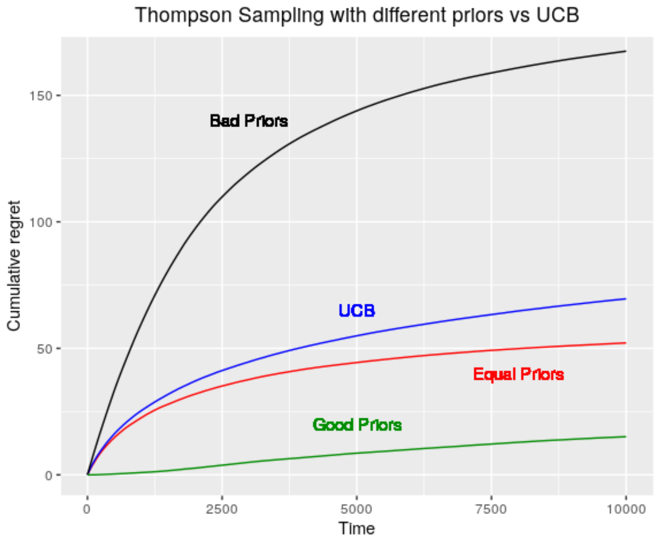


Figure 4: Total regret for all Thompson scenarios vs UCB

On the other hand, if we choose good hyperparemeters, as in Firgure 3b, we almost always choose the best arm, leading to less exploration, which is reflected by the linear behaviour of the accumulative regret, but also far less total regret as expected.

An important note is that, apart from scenario 2, all regret plots clearly follow a $log(T)$ behaviour as predicted from the [TLL85].

Finally, all the code, written in R, for the algorithms and plots can be found on my GitHub repository: `https://github.com/martind-hub/Thompson-Sampling-UCB/blob/main/TS_UCB_Regret.R`

There has been quite a lot of progress made in the online setting for the UCB algorithm too, for example in [GC11], and trying to implement Bayesian ideas to UCB in [KGC12].

4 Bayesian Optimisation

Here we follow the third problem in the sprint and consider a minimisation problem of the form:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}), \tag{4.1}$$

where $f : \mathcal{X} \to \mathbb{R}$ is a *black box* function defined over the domain $\mathcal{X}$[3]. Since the functional form is unknown, we have to resort to different methods to find its global minimum. In this case, let the chosen locations be: $(\mathbf{x}_1, \dots, \mathbf{x}_n)$, leading to collected data points: $D_n = \{(\mathbf{x}_i, y_i)_{i=1,\dots,n}\}$, where $y_i = f(\mathbf{x}_i)$ and further define $f_n^* = \min_{i=1,\dots,n}(y_i)$. Conditional on these observations, we can fit a Gaussian process that predicts the value of f for any $\mathbf{x} \in \mathcal{X}$ as: $f(\mathbf{x})|D_n \sim \mathcal{N}(\mu_n(\mathbf{x}), \sigma_n^2(\mathbf{x}))$. Let us consider the following utility function:

$$u(\mathbf{x}) = \max(0, f_n^* - f(\mathbf{x})). \tag{4.2}$$

We are interested in find a closed form expression of the acquisition function: $\alpha_n(\mathbf{x}) = \mathbb{E}[u(\mathbf{x})|D_n]$.

[3] We use boldface letters for the elements in the domain, as the domain could be multidimensional.

4.1 Derivation

We can rewrite $\alpha_n(\mathbf{x})$ as:

$$\mathbb{E}[\mathbb{1}_{f(\mathbf{x})\leq f_n^*}(f_n^* - f(\mathbf{x}|D_n))] = \mathbb{P}[f(\mathbf{x}) \leq f_n^*]\,\mathbb{E}[f_n^* - f(\mathbf{x})|D_n, f(\mathbf{x}) \leq f_n^*]. \tag{4.3}$$

We know that $\mathbb{P}[f(\mathbf{x}) \leq f_n^*] = \Phi\left(\frac{f_n^*-\mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)$, where Φ is the standard normal CDF. Using the linearity of the expectation operator, equation (4.3) becomes:

$$\alpha_n(\mathbf{x}) = \Phi\left(\frac{f_n^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)\big(f_n^* - \mathbb{E}[f(\mathbf{x})|D_n, f(\mathbf{x}) \leq f_n^*]\big). \tag{4.4}$$

Let us define $F_n^* = \frac{f_n^*-\mu(\mathbf{x})}{\sigma(\mathbf{x})}$. Then, the only unknown quantity left is the expectation in equation (4.4). Let $Z \sim \mathcal{N}(0,1)$. Then, we know that $\mathbb{P}[f(\mathbf{x}) \leq f_n^*)|D_n] = \mathbb{P}[Z \leq F_n^*] = \Phi(F_n^*)$. Hence,

$$\begin{aligned}\mathbb{E}[f(\mathbf{x})|D_n, f(\mathbf{x}) \leq f_n^*] = \mathbb{E}[\mu(\mathbf{x}) + \sigma(\mathbf{x})Z|D_n, Z \leq F_n^*] = \mu(\mathbf{x})\,\mathbb{E}[1|D_n, Z \leq F_n^*]\\ +\sigma(\mathbf{x})\,\mathbb{E}[Z|D_n, Z \leq F_n^*] = \left(\mu(\mathbf{x})\int_{-\infty}^{F_n^*}\phi(z)dz + \sigma(\mathbf{x})\int_{-\infty}^{F_n^*}z\phi(z)dz\right)\Big/\mathbb{P}[Z \leq F_n^*].\end{aligned} \tag{4.5}$$

We know that $\mathbb{P}[Z \leq F_n^*] = \Phi(F_n^*)$. Also, since $z\phi(z) = -\phi^{'}(z)$, using the Fundamental Theorem of Calculus, we finally arrive at:

$$\mathbb{E}[f(\mathbf{x})|D_n, f(\mathbf{x}) \leq f_n^*] = \mu(\mathbf{x}) - \sigma(\mathbf{x})\frac{\phi(F_n^*)}{\Phi(F_n^*.)} \tag{4.6}$$

Therefore, after some rearrangement, we the closed form expression of $\alpha(\mathbf{x})$ is:

$$\alpha_n(\mathbf{x}) = \underbrace{(f_n^* - \mu(\mathbf{x}))\Phi(F_n^*)}_{\text{Exploitation}} + \underbrace{\sigma(\mathbf{x})\phi(F_n^*)}_{\text{Exploration}}. \tag{4.7}$$

4.2 Quick overview

We called the first term in equation (4.7) exploitation. The reason behind is that, in order to decrease this term, we need to decrease $\mu(\mathbf{x})$. However, this means that we are choosing points with small means, hence exploiting our beliefs there. On the other hand, making the second term larger, requires increasing the variance, but this implies that we "explore" more of data, hence the name.

Finally, regret-based decision making applied to Bayesian Optimisation, works by choosing points $\mathbf{x}^{(i)}$, such that they minimise the total cumulative regret up to time T:

$$R(T) = \sum_{i=1}^{T} f(\mathbf{x}^{(i)}) - Tf(\mathbf{x}^*), \tag{4.8}$$

where $\mathbf{x}^*$ is the global minimum. For further information, see [BG20].

References

[TLL85] HERBERT ROBBINS T.L.LAI. "Asymptotically Efficient Adaptive Allocation Rules". In: 1985.

[GC11] Auŕelien Garivier and Olivier Cape. "The KL-UCB Algorithm for Bounded Stochastic Banditsand Beyond". In: 2011.

[KGC12] Emilie Kaufmann, Auŕelien Garivier, and Olivier Capp. "On Bayesian Upper Confidence Bounds for Bandit Problems". In: 2012.

[BG20] Kinjal Basu and Souvik Ghosh. "Adaptive Rate of Convergence of Thompson Sampling forGaussian Process Optimization". In: 2020.