Multi-Armed Bandit Problem

Sunith Suresh, Lin Xiao, Ilan Man and Sanjay Hariharan

April 25, 2016

1 Introduction

2 Multi-armed bandit review

2.1 Problem setup: Bernoulli Bandit

We formulate a simple bandit scheme involving 2 Machines and a Bernoulli Reward (0 or 1). We play one of the two machines at every iteration of the game for N iterations. Each Machine has a latent parameter governing its reward. At every step we draw a reward from the selected machine. Typically the objective is to maximize the expected total rewards. As such, the goal of the multi-armed bandit problem is to devise a scheme by which one can select the optimal machine to play at each time step.

A related and important concept to reward maximization is the concept of minimizing regret. Define R_N^* and $\hat{R_N}$ as the actual and expected maximum rewards, respectively, after N iterations. Therefore the regret of a given multi-armed bandit scheme is $R_N^* - \hat{R_N}$.

We designate the following:

$$\theta_1 \sim \text{Beta}(\alpha_1, \beta_1)$$

$$\theta_2 \sim \text{Beta}(\alpha_2, \beta_2)$$

$$R_1 = \text{Reward from Machine 1} \sim \text{Bernoulli}(\theta_1)$$

$$R_2 = \text{Reward from Machine 2} \sim \text{Bernoulli}(\theta_2)$$

In the beginning, we have no prior knowledge about either Machine, or more specifically, about the latent parameter governing each Machine. Therefore, we initially set $\alpha_1, \alpha_2, \beta_1, \beta_2 = 1$

2.1.1 Acquisition function

In order to maximize expected rewards, we define the acquisition function $U(\theta|X) = U(\alpha_1, \beta_1, \alpha_2, \beta_2|X)$ which is the function that determines how we choose the next machine to play. This function balances the trade off between selecting the best machine based on previous plays, and the possibility of a better machine that hasn't been select yet. This trade off is known as the exploration/exploitation dilemma.

As we begin playing and gathering information, we continuously update our parameters of interest based on the outcome. Specifically:

1. U(α_1,β_1,α_2, β_2)=j (We choose Machine j)
2. R_j=i (We play Machine j, and get Reward i)
3. If $R_j = 1 : \alpha_1 = \alpha_1 + 1$
4. If $R_j = 0 : \beta_1 = \beta_1 + 1$

The idea is to update our prior belief on the success of the individual Machines based on our empirical rewards. α_i represents success, while β_i represents failure

2.2 ϵ-greedy

- basic algorithm
- linear regret

2.3 upper confidence bounds

- explain concept of bounding regret
- application of Hoeffding's Inequality
- calculate UCB

3 Bayesian approach

- define rewards as bernoulli(π_i)
- select machine with probability θ_i
- update belief of machine
- goal is to maximize expecte reward

3.1 Thompson Sampling: Hueristic

- Lin

3.2 Analytical theory

- Sanjay/Lin

3.3 Dynamic programming and Gittins

- Sunith

4 Empirical Comparisons

4.1 Data set

5 Conclusion

6 References