Today, I am going to introduce Thompson Sampling. Thompson Sampling is an algorithm mainly used in online learning to balance between exploitation and exploration. This trade-off is the key question during out trip of online learning. So to help us better understand, first we need to know a famous model in online learning: multi armed bandits problem.

Suppose we enter a casino and faced with several bandits. We can choose any one of them and the bandit we choose would give us a reward randomly. Now the problem is, how can I maximize our cumulative reward in, let say, 100 chances. At first we have some prior information of them, for example, we believe the reward they provide, on average, are the same. Or we may choose our lucky number. After we pull several time, we obtain some observation of the reward and we need to make our decision. For example, we believe bandit 3 has the most reward. Whether I keep exploiting bandit 3 to get a good reward in a high probability or should I try other bandits to explore and identify the potential bandits.

Many problems can be formulated into such a model. For example, news recommendation. Jinri tou tiao has numerous users today. For each users, it has to decide which kind of news should be posted on their phone. So now, action, or bandits are the kind of the news. Another example is adaptive routing. Suppose we want to know the expectation of the time we need to spend from here to dong gangcheng. One way to finish is that first we post a time to all the bus drivers. Every bus driver report whether he spent more or less time than what we post, after he finishes a single trip. We update our estimation about the time based on their report. Since the traffic is random and the there is also noise in their report, therefore we need to learning the true expectation of the time.

In order to outline Thompson Sampling, let us consider a simpler model: Bernoulli bandit problem, which means the reward follows Bernoulli distribution with parameter θ. To facilitate out analysis, we can use beta distribution to describe our prior, which is conjugate with Bernoulli distribution. Thanks to the these two distributions, the posterior distribution can be expressed in such a clean manner after Bayesian update.

So here comes the essential part of today presentation. After we get posterior, what can we do? We want to identify the best bandit, and the expectation of the posterior distribution can be a good indicator to finish it. So we can select the bandit with the highest expectation of reward, this is called greedy algorithm. However, Thompson Sampling also selects the bandits with the highest indicator. However, that indicator is sampled from the posterior.

So what's the difference? Intuitively, consider we are faced with two bandits. One with parameter 0.5, and the other is 0.4. At the very beginning suppose we get 0 from bandit 1 and 1 from bandit 2. Then if we adapt greedy algorithm, we will choose bandit 2. There are also some probability we keep get reward 1 from bandit 2. From the law of large numbers, we know that the average reward we get from bandit 2 will approach 0.4. Although it is sub-optimal, but we will keep choosing it since we believe bandit 1 produce no reward. 0.4>0 and we stop exploration.

Generally, Thompson Sampling can beat greedy algorithm because greedy algorithm is not actively explore. One modification to greedy algorithm is called epsilon-greedy algorithm. It means each time when we make decision, there is probability epsilon for us to just randomly select an action to provide exploration. However, Thompson Sampling can beat it as well, since that exploration is rough, just uniformly distribution exploration effort without utilize more information from the posterior.

Here are the results of the experiment of epsilon greedy algorithm and Thompson Sampling. The main difference is that ever action 1 is suboptimal, greedy algorithm still spend a lot of time on it, while Thompson Sampling quickly identify it.

I think I finish the basic concepts in Thompson Sampling. Let's go to the next chapter. In fact, sampling from the posterior is not always easy since the likelihood is determined by the problem itself and maybe there is no prior conjecture with it. Therefore, we need approximation to help us sample from an arbitrary distribution. Of course we can use Gibbs Sampling but it is computationally demanding. So firstly I want to introduce Laplace approximation. Since the posterior is usually in the product form, take log of it would facilitate out calculation. And then Laplace approximation means we can use second order taylor expansion around the mode to estimation. Node that the overline phi is the mode, therefore its first order is 0 and we only need to deal with two term. Also note that the distribution density of phi is proportional to exponential with quadratic form, which is very similar to normal distribution. Then we normalize it and finally we use a normal distribution to approximate it. Another way to understand it is that from the law of large numbers, when the amount of the data increase, they behave more like a normal distribution and it is reasonable to approximate it with normarl distribution. Come back to Bernoulli distribution, we can calculate overline phi and C

写一下 Bernoulli 公式

Lavigen Monte Carlo: use gradient information to quickly approach stationary distribution while injecting randomness

Bootstrapping: generates a hypothetical history , which is made up of t-1 action observation pairs, each sampled uniformly with replacement from true history