MS&E 338: Reinforcement Learning
Lecture 3 & 4 Outline

Ian Osband

January 27, 2015

These notes are work in progress and principally meant to guide my own lecture preparation. Please take any questions/clarifications to Piazza.

1 Review and notation

We are looking at multi-armed bandit problems, or learning to optimize an unknown function $f \in \mathcal{F} : \mathcal{X} \to \mathbb{R}$. At each timestep the agent can choose an action $x_t \in \mathcal{X}$ and receive a reward $r_t = f^*(x_t) + \epsilon_t$. Here ϵ_t is zero mean sub-gaussian noise (obeys some boundedness assumption) that is conditionally independent of H_t the filtration to time t.

The agent's goal is to maximize the cumulative rewards through time

$$\mathbb{E}\left[\sum_{t=1}^{T} r_t \middle| x_t \sim \pi : H_t \to \mathbb{P}(\mathcal{X})\right].$$

A learning algorithm π is a mapping from data H_t to a distribution of actions over $\mathcal{X}$ which are chosen at time t by the agent. The key point of this problem is that the agent does not know the true underlying function f^*, but can learn about it through its interactions with the environment.

This leads to a fundamental tradeoff: **exploration vs exploitation**.

2 Performance metrics and regret bounds

We're faced with a fundamentally very practical task "obtain as much rewards as possible from the system", so at first glance there doesn't seem to be much need for analysis. The key difficulty in a multi-armed bandit problem is that **we do not how good is "good" for our problem**. We can look at a reward of 100 after 3 timesteps, but we have no idea what the optimal reward could be. A good algorithm could receive low rewards in a hard problem, whereas a bad algorithm could do well in an easy one. We need a way to assess our algorithms and to guide better performance.

2.1 Asymptotic convergence or “no regret”

One natural guarantee is that given enough time/data our algorithm will eventually converege to the optimal solution. This is sometimes called a “no regret” algorithm and is a reassuring sanity check on the properties of an algorithm.

In many cases however, the criteria for asymptotic convergence is really not very strong and fails to distinguish between good and bad algorithms. In particular any finite set of choices is irrelevant to asymptotic guarantees. Plus... in the long run we’re all dead, so let’s try to do a little better than that!

2.2 Statistical efficiency / PAC bounds

A common notion in algorithm design is that of computational efficiency, “how many floating point operations do you require to complete the algorithm?”. Motivated in a similar manner is the idea of statistical efficiency, “how many pieces of data do you require before you complete the algorithm?”. In this case the “algorithm” is not entirely properly defined, but for most cases we will settle for an ϵ-approximate answer or “Probably Approximately Correct”.

Definition 1 (Sample complexity bound)**.** *Algorithm π satisfies a sample complexity bound (aka PAC bound) with probability $\geq 1 - \delta$ the number of ϵ-suboptimal decisions is bounded by* $\mathrm{PAC}(\pi, \mathcal{F}, \epsilon, \delta)$.

Some papers may pose the bounds in a slightly different way, but the general idea is the same. Statistical efficiency results are nice since they mirror what we already do in computation... but what we really care about are rewards!

2.3 Problem-specific reward bounds

Our goal is to maximize expected rewards, so perhaps the most obvious criteria woudl be to bound the expected rewards from below... *does anyone see a problem with this*?

Of course this is not going to generalize well between problems. In general it doesn’t make any sense to bound the absolute level of rewards, what if I measure my money in USD vs GBP vs JPY? For a very specific setting we may require precise bounds on our performance (e.g. a guarantee that a passenger train will not explode) but this is not going to make sense in a general framework for bandit algorithms.

2.4 Regret bounds

If bounding the value of the rewards directly won’t work, we can actually accomplish something pretty similar through regret bounds. Regret is a notion that is pretty similar to the colloquial meaning:

“If I’d had perfect knowledge, how much better off could I have expected to do than what I actually did?”

Definition 2 (Regret)**.** *We define the regret of an algorithm π to time T:*

$$\mathrm{Regret}(\pi, T) = \mathbb{E}\left[\sum_{t=1}^{T} f^*(x^*) - f^*(x_t) \middle| x_t \sim \pi\right]$$

Where x^* is the optimal action according to perfect knowledge (but not of the system noise).

Here we can characterize all “no-regret” algorithms as those with Regret $o(T)$. Clearly, these algorithms can have extremely large regret... We note that regret bounds are similar in many ways to statistical efficiency results. In particular we can convert between them in most cases.

2.5 Key takeaway: algorithm design

Many of the naive bandit algorithms end up with regret bounds that are exponential in some problem parameter. These lead to real world performance degradation, but unlike other forms of machine learning or statistics it can be very difficult to point these out. You cannot conclusively demonstrate the rewards an algorithm would have received without *actually doing it.*

This problem is specific to bandits/RL and so it’s even more important to guide your algorithms through sound principles and analysis. If you don’t, you may never find out what you’re missing out on!

2.6 Aside: Bayes optimal

One other natural approach is to start with some prior ϕ over possible f^* and then solve for the algorithm with the optimal expected rewards. Formally this is coherent, principled and simple to state however it is usually computationally inractable.

There is a notable exception to this, the case of independent arms one can compute Gittins indices to guide a Bayes-otpimal policy. But for more complex problems the Bayes optimal strategy is fundamentally intractable due to exponential lookahead dependencies. There are algorithms however, which attempt to approximate it with some form of performance guarantees.

3 Optimism in the face of uncertainty

Perhaps the fundamental problem for multi armed bandits is that mis-estimation errors are not necessarily self-correcting. This is because the data collection is intimately related to the decision optimization. In particular, under-estimation at a given $x \in \mathcal{X}$ may lead to no more data being collected (since it appears to be a bad option) and so the algorithm may never learn. On the other hand, over-estimation generally will correct itself, since the agent likes to choose “good” things it will continue to gather data and converge to the truth. Most efficient bandit algorithms exploit this imbalance by artificially injecting “optimisim” into their estimates.

Algorithm 1
Optimism in the face of uncertainty (OFU)

```
for time $t = 1, 2, ..$ do
    compute $\mathcal{F}_t = \{f \in \mathcal{F} |$ statistically plausible given $H_t\}$
    choose $x_t \in \arg_x \max_{f \in \mathcal{F}_t, x \in \mathcal{X}} \overline{f}(x)$
    observe $r_t = f^*(x_t) + \epsilon_t$
end for
```

So this general pseudocode gives us an algorithm which can be adapted in lots of different ways.

3.1 UCB independent arms

We will now consider the case of K independent arms, which is the classic motivation for multi-armed bandits. For ease we will assume that all rewards $r_t \in [0,1]$. In fact sub-Gaussian noise will be just as good...

Proof outline:

- Imagined vs optimal decomposition (OFU)
- Azuma-Hoeffding inequality
- Contribution from missed confidence sets
- Concentration/widths within confidence sets
- Bringing it all together

First let us write actions/measurements $\in \{1, 2, ..K\}$ and assume that at each measurement point we construct upper confidence bounds U_t.

$$\begin{aligned} \text{Regret}(\text{UCB}, T) &= \sum_{t=1}^{T} f^*(x^*) - f^*(x_t) \\ &= \sum_{t=1}^{T} [U_t(x_t) - f^*(x_t)] + [f^*(x^*) - U_t(x_t)] \end{aligned}$$

We will construct U_t upper confidence bounds so that with high probability $U_t(x_t) \geq f^*(x^*)$. To do this we will use the Azuma-Hoeffding inequality. A very similar form of analysis can be completed without quite such rigid boundedness assumptions.

Lemma 1 (Azuma-Hoeffding)**.** *Suppose M_n is a martingale with bounded differences $|M_n - M_{n-1}| \leq c_n$ almost surely. Then for all integers N and $\beta > 0$:*

$$\mathbb{P}\left(|M_N - M_0| > \beta\right) \leq 2 \exp\left(\frac{-\beta^2}{2\sum_{n=1}^{N} c_n^2}\right)$$

The proof follows from elementary martingale arguments.

Now we note that the empirical estimate for $f^*(i)$ after time t, which we write as

$$\hat{f}_t(i) := \frac{\sum_{i=1}^{t} \mathbb{1}\{x_t = i\} r_t}{\sum_{i=1}^{t} \mathbb{1}\{x_t = i\}}$$

is a bounded zero mean martingale for the true value. Applying Azuma-Hoeffding gives:

$$\mathbb{P}\left(|f^*(i) - \hat{f}_t(i)| > \sqrt{2\frac{\log(\epsilon/2)}{n_t(i)}}\right) \leq \epsilon$$

Where $n_t(i) := \sum_{i=1}^{t} \mathbb{1}\{x_t = i\}$ is the number of observations from action i by time t. If we choose $\epsilon_t = \delta t^{-2}$ we can use that $\sum_{t=1}^{\infty} t^{-2} = \pi^2/6 < 2$ in our resulting analysis.

Algorithm 2
UCB for independent arms

```
**Input:** arms $1, .., K$ and confidence $\delta$
**for** time $t = 1, 2, ..$ **do**
    compute $U_t(i) = \hat{f}_t(i) + 2\sqrt{\frac{\log(tK2\delta)}{n_t(i)}}$ for each $i$
    choose $x_t \in \arg\max_{x \in \mathcal{X}} U_t(x)$
    observe $r_t = f^*(x_t) + \epsilon_t$
**end for**
```

With this in mind we are ready to formally state our algorithm, let $U_t(i) = \hat{f}_t(i) + \sqrt{\frac{4\log(t\delta)}{n_t(i)}}$. At each timestep we will pick $x_t \in \arg\max_x U_t(x)$. Now at any time t:

$$\begin{aligned}
\bigcup_{t=1}^{T} \mathbb{P}\left(f^*(x^*) > U_t(x_t)\right) &\leq \bigcup_{t=1}^{T}\bigcup_{i=1}^{K} \mathbb{P}\left(|f^*(i) - \hat{f}_t(i)| > \sqrt{\frac{4\log(t\delta)}{n_t(i)}}\right) \\
&\leq \bigcup_{t=1}^{T}\bigcup_{i=1}^{K} \delta t^{-2} \\
&\leq 2K\delta
\end{aligned}$$

So with probability at least $1 - 2K\delta$ this optimistic algorithm will always over-estimate the true optimal rewards. Hence we can ignore the regret contribution from $[f^*(x^*) - U_t(x_t)] \leq 0$ in these instances.

This step of analysis may seem pretty trivial but it is the key point to most of the proofs in the area. Bounding the regret is inherently hard, because we want to measure how well we do with respect to an *unkown* benchmark. Using the principle of optimism in the face of uncertainty (OFU) we have reduced the problem to a question of how well we have estimated our optimistic levels $U_t(x_t)$, which we know, compared to the actual rewards which we observe $f^*(x_t)$.

We condition on the event that all the confidence sets hold $f^* \in \mathcal{F}_t$ or equivalently $f^*(i) \leq U_t(i)$ $\forall i$ to say that with probability $\geq 1 - 2K\delta$:

$$\begin{aligned}
\text{Regret(UCB}, T) &\leq \sum_{t=1}^{T} [U_t(x_t) - f^*(x_t)] \\
&\leq \sum_{t=1}^{T} \sqrt{\frac{4\log(t\delta)}{n_t(x_t)}} \\
&\leq \sqrt{4\log(T\delta)} \sum_{t=1}^{T} \sqrt{\frac{1}{n_t(x_t)}}
\end{aligned}$$

Now we need to use an integral comparison to bound the contribution from this final sum (a picture is a helpful tool here which I suggest you should draw):

$$\sum_{t=1}^{T} \sqrt{\frac{1}{n_t(x_t)}} = \sum_{i=1}^{K} \sum_{n=1}^{n_t(i)} \sqrt{\frac{1}{n}}$$

We can now use that $\sum_{n=1}^{T} \sqrt{\frac{1}{n}} \leq 2\sqrt{T}$ by integral comparison. Therefore the entire summation is bounded

$$\sum_{t=1}^{T} \sqrt{\frac{1}{n_t(x_t)}} \leq 2\sqrt{KT}.$$

This completes the proof and shows that even for this very simple and loose analysis with probability at least $1-\delta$: $\text{Regret}(\text{UCB}, T) \leq 4\sqrt{KT \log(2K\delta T)}$. A more careful analysis can pare down some of the constant/log terms... but this is basically the flavour of the correct scaling.

4 Posterior sampling / Thompson sampling

We now reintroduce the Thompson sampling algorithm:

Algorithm 3
Posterior sampling / Thompson sampling

```
Input: prior $\phi$ for $f^*$
for time $t = 1, 2, ..$ do
    sample $f_t \sim \phi(\cdot|H_t)$
    choose $x_t \in \arg\max_{x \in \mathcal{X}} f_t(x)$
    observe $r_t = f^*(x_t) + \epsilon_t$
end for
```

This has been around a long time (1933) and is a very natural "probability matching" algorithm. It amounts to saying, I don't know what the true actions is so let me take an action randomly according to the probability it is optimal.

ASIDE: show some good empirical performance of the algorithm. This is a really interesting *heuristic* but for a long time it was really nothing more than that. How can we make any sort of analytical guarantees? In fact, we can show some pretty astonishing connections between UCB and Thompson sampling.

Analysis outline:

- Imagined vs optimal decomposition (Sampling)
- Posterior sampling lemma
- UCB-style regret bound decomposition
- The proofs will now go exactly the same
- Caveat both the imagined and optimal must lie in confidence set
- Discuss regret vs Bayes regret vs Expected regret vs Bayes Risk
- Maybe show how to do posterior updating? (normal/normal)

Lemma 2 (Posterior sampling)**.** *If ϕ is the distribution of f^* then for any $\sigma(H_t)$-measurable function g:*

$$\mathbb{E}\left[g(f^*)|H_t\right] = \mathbb{E}\left[g(f_t)|H_t\right]$$

This follows immediately from the definition of the posterior sampling algorithm since conditioned on H_t, the functions f^* and f_t are equal in distribution.

This lemma holds the key for analysing a posterior sampling (Thompson sampling) algorithm via an optimistic proof. We begin with the same regret decomposition at each stage t we add and subtract the *imagined* optimal reward $f_t(x_t)$:

$$\begin{aligned}\text{Regret}(\text{PS}, T) &= \sum_{t=1}^{T} f^*(x^*) - f^*(x_t) \\ &= \sum_{t=1}^{T} [f_t(x_t) - f^*(x_t)] + [f^*(x^*) - f_t(x_t)]\end{aligned}$$

Using the posterior sampling lemma we can see that the contribution of the second term is zero in expectation. Note that this expectation is now taken over the prior $\phi(\cdot)$ since f^* is no longer regarded as fixed.

4.1 Regret vs Bayesian Regret

Let's stop to discuss the different type of regret bounds and how they differ.

$$\text{Regret}(\pi, T, \theta) = \sum_{t=1}^{T} \mathbb{E}\left[f^*(x^*) - f^*(x_t) \middle| \theta \right]$$

$$\text{BayesRegret}(\pi, T) = \sum_{t=1}^{T} \mathbb{E}\left[f^*(x^*) - f^*(x_t) \right]$$

Note that via Markov's inequality means that bounds on BayesRegret are essentially asymptotic bounds on Regret.

$$\mathbb{P}\left(\frac{\text{Regret}(T, \pi, \theta)}{g(T)} \geq M \right) \leq \epsilon \ \forall T \in \mathbb{N}$$

Similarly if we use prior $\tilde{\phi}$ when the true prior is ϕ then the Bayes Regret satisfies:

$$\mathbb{E}_\mu[\text{Regret}(\pi, T, \theta)] \leq \| \frac{d\mu}{d\tilde{\mu}} \|_{\tilde{\mu}, \infty} \mathbb{E}_\mu[\text{Regret}(\pi, T, \theta)]$$

4.2 Posterior concentration via UCB concentration

We know that for any arbitrary upper confidence sequence we can write:

$$\begin{aligned}\text{BayesRegret}(\pi, T) &= \mathbb{E}\left[\sum_{t=1}^{T} [f_t(x_t) - U_t(x_t)] + \sum_{t=1}^{T} [U_t(x_t) - f^*(x_t)] \right] \\ &\leq \mathbb{E}\left[\sum_{t=1}^{T} |f_t(x_t) - U_t(x_t)| + \sum_{t=1}^{T} |U_t(x_t) - f^*(x_t)| \right] \\ &\leq 2\mathbb{E}\left[\sum_{t=1}^{T} |U_t(x_t) - L_t(x_t)| \right]\end{aligned}$$

And now we can *any* U_t and L_t upper and lower confidence bounds with the concentration rates that we find there to get Bayesian regret bounds for Thompson sampling.

5 Beyond independent actions

- Actually you can never do better than $\sqrt{KT}$ (proof / example)
- Most problems you can actually think of have K very large (infinite)
- In these settings we may just be hozed... can't expect to learn quickly
- But if there is some low-dimensional structure (e.g. linear) you might be able to do better.
- Key here will be to exploit inter-dependence between arms

Definition 3 (Sub-Gaussian random variable)**.** *X is σ-sub Gaussian if for all $\lambda \in \mathbb{R}$:*

$$\mathbb{E}[\exp(\lambda X)] \leq \exp(\lambda^2 \sigma^2 / 2)$$

Using Markov's inequality we can show that this implies for any σ-sub Gaussian random variable

$$\mathbb{P}(X \geq \lambda) \leq \frac{\mathbb{E}[e^{tX}]}{e^{t\lambda}} \leq \exp(-t\lambda + \sigma^2 t^2 / 2)$$

Now minimizing this bound over t we can conclude:

$$\mathbb{P}(X \geq \lambda) \leq \exp(-\lambda^2 / 2\sigma^2)$$

You should note that anything Gaussian is sub-Gaussian, and also anything bounded almost surely.

5.1 Linear bandits

We now consider the case of a "linear" bandit which is to say that $f_\theta(x) = \theta^T x$ for $x, \theta \in \mathbb{R}^d$ and imagine that ϵ_t is the usual σ-sub Gaussian noise. For our analysis we will imagine that $\theta \in \Theta$ is uniformly bounded by $\|\theta\|_2 \leq c_1$ and $\|x\|_2 \leq c_2$.

If we discretized each x we would have to make $O\left(\left(\frac{c_1}{\epsilon}\right)^d\right)$ separate x_i to get back to our discrete setting with ϵ precision. Also we could potentially lose out on optimal performance forever if the optimal x^* was not exactly on the right place!

Actually it's possible to do much better in our regret analysis if we are smarter about making our confidence sets (diagram). Using two separate confidence sets we can actually end up creating better confidence sets that are actually able to converge.

$$\text{Analysis 1: } O(d \log(T) \sqrt{T})$$

$$\text{Analysis 2: } O(\mathbb{E}\sqrt{\|\theta\|_0 dT})$$

The idea for analysis is to build ellipsoidal confidence sets around the least squares estimates with covariance drawn from the data.

5.2 OFU vs Sampling

- Building the correct confidence sets can be hard in general!
- Solving the resultant optimistic problem can be intractable.
- Sampling is usually much easier (normal/normal conjugates)
- Any analysis for UCB will hold for sampling algorithm!

5.3 General functional families

How can we generalize this notion beyond independent, linear and try to deal with the general problem of learning $f^* \in \mathcal{F}$?

- Examples of systems which aren't exactly these closed form type.
- Some classes are "simple" in supervised learning but hard for bandits
- Eluder dimension!
- Give the final result and a hint at how to do it.

For $f^* \in \mathcal{F}$ with $\|f^*\|_2 \leq C$ and σ-sub Gaussian noise then the BayesRegret of following the posterior sampling algorithm is boudned:

$$\text{BayesRegret}(\text{PS}, T) = \tilde{O}\left(\sigma C \sqrt{d_E(\mathcal{F}) d_K(\mathcal{F}) T}\right)$$

Where d_E is the Eluder dimension, d_K is the Kolmogorov dimension and $\tilde{O}$ ignores logarithmically small terms in the bounds.

6 Next steps

Read "Learning to Optimize via Posterior Sampling" for a great rundown of bandits and everything we've been through in these lectures.

When Ben is back we'll move onto Reinforcement Learning, where the actions we take in any single timestep can affect the future rewards of the system.