Bandits

David S. Rosenberg

NYU: CDS

March 5, 2021

Contents

1. Motivation
2. Online vs. Offline learning
3. The bandit setting (informal)
4. Bandit setting (formalized)
5. Bandit strategy: ε-greedy

Motivation

William Thompson (Yale, Dept. of Pathology, 1933)

- Thompson was thinking about comparing two medical treatments
- Standard practice:
  - Run a randomized controlled trial (RCT)
  - If one treatment is significantly better than the other,
    - use the better one, going forward.
  - Otherwise,
    - back to the lab.

William Thompson (Yale, Dept. of Pathology, 1933)

- The more data we have, the more certainty we have in RCT conclusions
- But what if we only have a modest amount of data?
- Suppose data is strongly suggestive that one treatment is better,
  - but not yet "conclusive" (as defined by $p < 0.05$, or whatever).
- Should we adjust our actions in accordance with this information?

Thompon's 1933 paper

ON THE LIKELIHOOD THAT ONE UNKNOWN PROBABILITY EXCEEDS ANOTHER IN VIEW OF THE EVIDENCE OF TWO SAMPLES.

By WILLIAM R. THOMPSON. From the Department of Pathology, Yale University.

Section 1.

In elaborating the relations of the present communication interest was not centred upon the interpretation of particular data, but grew out of a general interest in problems of research planning. From this point of view there can be no objection to the use of data, however meagre, as a guide to action required before more can be collected; although serious objection can otherwise be raised to argument based upon a small number of observations. Indeed, the fact that such objection

From [Tho33]: Basically he's saying: use whatever data you have as a guide to action, and adjust appropriately as you get new data.

William Thompson

- Thompson says, that if

  "P is the probability estimate that one treatment . . . is better than a second, as judged by data at present available, then we might take some monotone increasing function of P, say $f(P)$, to fix the fraction of such individual to be treated in the first manner; until more evidence may be utilised . . . the remaining fraction . .. to be treated in the second manner."

- For example, we could take $f(P) = P$.
- Then if we're 90% sure that 1 is better than 2, then
  - 90% of people get treatment 1 and
  - 10% of people get treatment 2
- As we get more data, these percentages will adjust accordingly.
- This is the essence of "**Thompson sampling**" as we know it today.

Note that Thompson is taking a Bayesian approach from the first sentence: having a probability P that one treatment is better than a second is something that we have in a Bayesian approach, but not the classical frequentist approach. With modern terminology, we would say that P is the posterior probability of the event that treatment 1 is better than treatment 2 (or the prior probability, if no data has yet been observed).

Thompson's high level pitch

- Suppose based on initial experiments, we have probability $P > \frac{1}{2}$ that treatment 1 is better.
- 3 possibilities:
  - **Pure exploitation**: End all experiments and use Treatment 1 from now on.
  - **Pure exploration**: Continue experiment with random (50%) treatment assignment. '
  - Something in between, such as Thompson's proposal.
- With pure exploitation, we have probability $1 - P$ that we're using the worse treatment,
  - **from now until infinity...**
- With pure exploration, we're giving 50% of people the worse treatment
  - **until we end the experiment.**

Exploration / exploitation tradeoff

- We can choose the action that seems best
  - **according to the data we already have.**
- We can choose the action that is optimal for
  - **improving our knowledge about the value of different actions**.
- The **explore/exploit** tradeoff refers to the tradeoff between these extremes.
- The bandit setting is a natural setting for exploring this problem.

Online vs. Offline learning

Online supervised learning

Online supervised learning proceeds in rounds of the following steps:

1. Observe input x.
2. Take action a.
3. Observe label y (often called the "response" in regression).
4. Evaluate action in relation to the label with a **known loss function** $\ell(a, y)$

Typically the action is chosen by a prediction function $a = f_\theta(x)$.

- Supervised learning: choose $f_\theta(x)$ from a training set of historical (x, y) pairs
- Online supervised learning: $f_\theta(x)$ is updated after each round based on new (x, y)

Online vs. offline

- Online not much different from offline.
- Conceptually, just retrain model on all data after every round.
- In practice, we probably can't wait for a full retraining.
- In practice, an **online learning algorithm** is one that can update with a new round of training data (x, y), without looking at the previous training data.

SGD is online learning!

- SGD is an online learning algorithm.
- From this perspective, we no longer have epochs.
- Whenever we get a new data point or mini-batch, take an update step.
- In practice, periodically reinitialize your model with a full batch retraining.

- You might be tempted to think that online learning is more appropriate when the world is changing over time, and we want our model to adapt.
- This isn't really the case. Online algorithms generally assume the world is stationary. For example, SGD usually has a decaying step size because we want to "remember" what we've learned before. If we want to forget what we learned from old training data, we shouldn't decay the step size after a certain point.
- If you think the world is changing over time, it's better to explicitly bring your assumptions into the problem. For example, if you think the last 2 weeks of data are much more relevant than anything preceding it, you might try training on a rolling window of 2 weeks, or exponentially weighting training data so that most of the weight is given to data from the last two weeks.

The bandit setting (informal)

Observing the label y is very helpful!

Key feature of supervised learning

Once we observe y, it's usually straightforward to figure out what the optimal action would have been:

$$a^* = \arg\min_a \ell(a, y).$$

Example: Multiclass Classification and Regression

Once we observe the correct class label (multiclass) or the true response (regression), we know the optimal action would have been to produce the correct label or the true response as an action.

What if the outcome y is never revealed?

- Consider k-class classification with 0/1 loss.
- We get an x and we predict class label 3.
- We receive a loss of 1 $\implies$ we were wrong.
- The correct class label y is **not** revealed.
- What to do next time we receive the same or a similar x?
- Try a different label and hope for the best...
- This setting forces us to do trial & error to figure out the optimal action.
  - Makes it much harder to learn!

Recommendations with full rating feedback

- There are 5 hit movies and we want to recommend one to a user
- We get an x that describes the user.
- We choose a movie and recommend it (that's our action).
- In online learning, we would then get feedback on
  - which of the 5 movies the individual would actually have liked best
  - that would be the "label"
- This is known as **full feedback** since it's enough information to determine
  - what would have been the best movie to suggest (i.e. best action).
- If another individual with the same or similar x shows up, we would know what to do.

Recommendations with partial rating feedback

- A [slightly] more realistic scenario:
- User watches the movie we recommend and gives a rating
  - Suppose 4 out of 5 stars
- 4 out of 5 isn't bad.
- No feedback on the other 4 movies.
- If we get a similar user $x' \approx x$, should we recommend the same movie?
- Or should we try a different movie to see if we can get to 5 out of 5?
- This is another exploration / exploitation problem.

The bandit problem

A bandit problem proceeds in rounds of the following steps:

1. [Optional] Observe input/**context** x.
2. Take action a.
3. Receive loss $\ell \in \mathbb{R}$.

- Note that label y is never revealed to us.
- Bandit problems may not even have a label.

Types of bandit problems

- **Multiarmed bandit**: when the set of possible actions is finite
- **Contextual bandit**: when a bandit problem has a context x in each round
  - Context x can help determine the best action to take
- **Losses vs. Rewards:** In bandits (and reinforcement learning), we often speak in terms of receiving rewards $r \in \mathbb{R}$ rather than losses $\ell \in \mathbb{R}$.
- Depending on whether we have rewards or losses, we either try to
  - maximize the total rewards received, or
  - minimize the total losses received.

Bandit setting (formalized)

Stochastic k-armed bandit

Stochastic k-armed bandit

1. Environment samples **reward vectors** for all rounds:

$$R_1, \ldots, R_T \text{ i.i.d. } P,$$

   where $R_t = (R_t(1), \ldots, R_t(k)) \in \mathbb{R}^k$.
2. For $t = 1, \ldots, T$,
   1. Our algorithm **selects action**/arm $A_t \in \{1, \ldots, k\}$ based on history

$$\mathcal{D}_t = \Big((A_1, R_1(A_1)), \ldots, (A_{t-1}, R_{t-1}(A_{t-1})) \Big).$$

   2. Our algorithm **receives reward** $R_t(A_t)$.

- We **never observe** $R_t(a)$ for $a \neq A_t$.

- It might look cleaner to say that at the beginning of every round, the environment generates $R_t \in \mathbb{R}^k$ from P. But we want to be very clear that $R_1, \ldots, R_T$ are
  1. generated i.i.d. and are
  2. generated before any of the actions $A_1, \ldots, A_T$ are generated.
- In fact, the assumption is generally that the distribution is a product distribution: $P = P_1 \times P_2 \times \cdots \times P_k$. In other words, the rewards are independent, both across actions within a particular round and across rounds. But we won't really need this.
- We use a capital letter A_t for action because we're thinking of A_t as random variables, since 1) we allow algorithms to use randomness in action selection, but more fundamentally, 2) they should depend on the history of rewards received, which are assumed to be random.
- However, the "stochastic" in stochastic k-armed bandit refers to the randomness of the reward vectors.

A bandit algorithm

- At the beginning of round t, the previous **observation sequence** is

$$(A_1, R_1(A_1)), \ldots, (A_{t-1}, R_{t-1}(A_{t-1}))$$

- In each round t, a **bandit algorithm** chooses an action A_t
  - based **only on** the previous observation sequence and
  - a random number (for randomized algorithms)

The connection with missing data and RCTs

- If $k = 2$ we can write the observation sequence as

t	A	$R(1)$	$R(2)$
1	1	6.1	?
2	2	?	1.2
3	1	0.9	?
4	2	?	3.0
5	2	?	1.9

- The "full-data" for a bandit at the beginning of round t would be

$$(A_1, R_1(1), \ldots, R_1(k)), \ldots, (A_{t-1}, R_{t-1}(1), \ldots R_{t-1}(k)).$$

The connection with missing data and RCTs

t	A	$R(1)$	$R(2)$
1	1	6.1	?
2	2	?	1.2
3	1	0.9	?
4	2	?	3.0
5	2	?	1.9

- In causal inference setting, we want to estimate $\mathbb{E}[R(1) - R(2)]$ from this data.
- With bandits, we want to **choose actions** A_t that maximize rewards: $\sum_t R_t(A_t)$.
- The problems are clearly related... but how are they different?

Bandits vs. treatment effect estimation

- In treatment effect estimation, we want good estimates $\mathbb{E}R(1)$ and $\mathbb{E}R(2)$.
  - (Or $\mathbb{E}Y(1)$ and $\mathbb{E}Y(0)$ in the notation of that module.)
- Suppose $\mathbb{E}R(1) \ll \mathbb{E}R(2)$.
- In bandits, we don't really care about getting a precise estimate of $\mathbb{E}R(1)$.
- Once we're quite sure that $\mathbb{E}R(2) > \mathbb{E}R(1)$, we don't need
  - a precise estimate of $\mathbb{E}R(1)$, or
  - a precise estimate of $\mathbb{E}R(2)$ for that matter.
- Knowing $\mathbb{E}R(2) > \mathbb{E}R(1)$ is enough to determine optimal action choice.

$\implies$ Choose actions that have some chance of giving the best rewards and ignore the rest

Bandit strategy: ε-greedy

Strategies for stochastic bandits

- Let's try to optimize the expected cumulative reward:

$$\mathbb{E}\left[\sum_{i=1}^{T} R_i(A_i)\right].$$

- Need a strategy for selecting action A_t at time t,
  - based on the observation sequence $(A_1, R_1(A_1)), \ldots, (A_{t-1}, R_{t-1}(A_{t-1}))$.
- In any round t, the expected reward for playing action a is $\mathbb{E}R_t(a)$.
- So the optimal action is the action that has the largest expected reward.

Expected rewards and estimate

- Let's write $q_*(1) = \mathbb{E}R_t(1), \ldots, q_*(k) = \mathbb{E}R_t(k)$.
- Let $\hat{q}_t(1), \ldots, \hat{q}_t(k)$ be the natural estimators of these parameters (based on the observation sequence through time $t-1$):

$$\hat{q}_t(a) = \frac{\sum_{i=1}^{t-1} \mathbb{1}[A_i = a] R_i(a)}{\sum_{i=1}^{t-1} \mathbb{1}[A_i = a]} \qquad \text{for } a = 1, \ldots, k.$$

- If the denominator is 0, we'll take $\hat{q}_t(a) = 0$, for some other default value, say 0, or whatever our best guess is for $q_*(a)$.

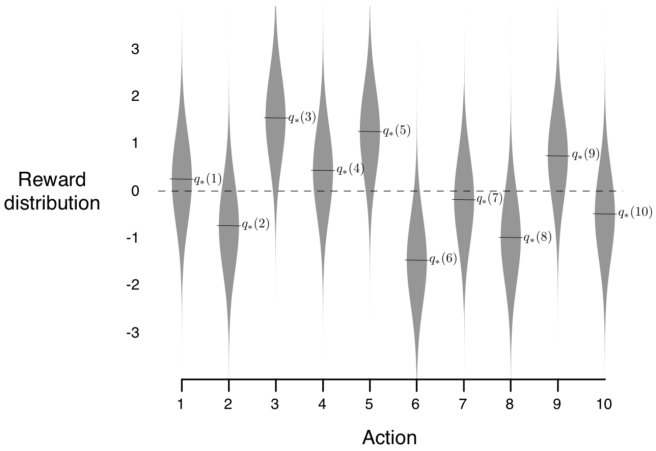
The ε-greedy algorithm

- Define the **greedy** action at time t to be the action with the largest mean estimate:

$$a_t^{\text{greedy}} = \arg\max_a \hat{q}_{t-1}(a).$$

- We'll assume that we break ties in the argmax at random.
- In the ε-greedy algorithm, in round t
  - With probability ε we **explore**: take $A_t \sim \text{Uniform}(1, \ldots, k)$
  - With probability $1-\varepsilon$ we **exploit**: take $A_t = a_t^{\text{greedy}}$.
- We're "exploiting" our knowledge with probability $1-\varepsilon$.
- We're "exploring" to gain better estimates of the arm rewards with probability ε.

The 10-armed bandit from Sutton and Barto[1]



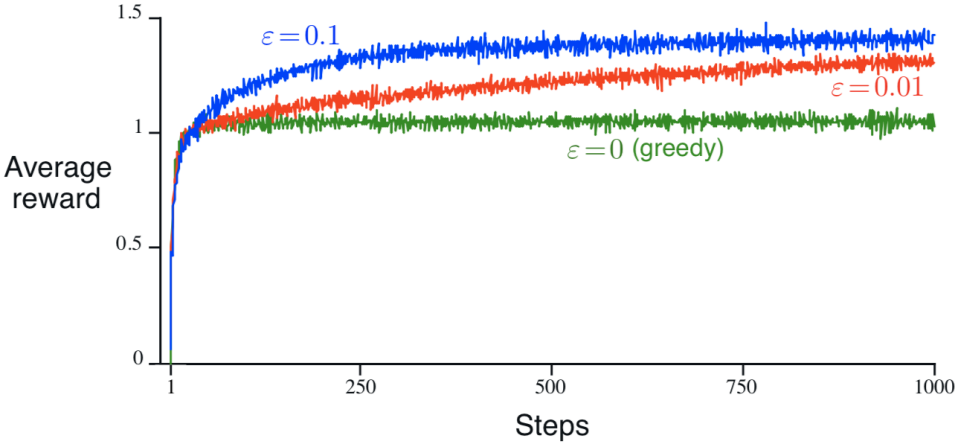
[1]Fig 2.1 in [SB18, Sec 2.3].

This is a single instance of the "10-armed testbed" from [SB18, Sec 2.3]. Each instance is a 10-armed bandit with reward distributions chosen randomly as follows:

- Let $q_*(a) = \mathbb{E}[R_t \mid A_t = a]$, for $a = 1, \ldots, 10$, denote the mean of the reward distribution for action a, for all time steps t.
- We choose each $q_*(a)$ i.i.d. from $\mathcal{N}(0, 1)$, for each a.
- Then the reward distribution for action a is given by $\mathcal{N}(q_*(a), 1)$.

We repeat this process 2000 times to generate 2000 random 10-armed bandits.

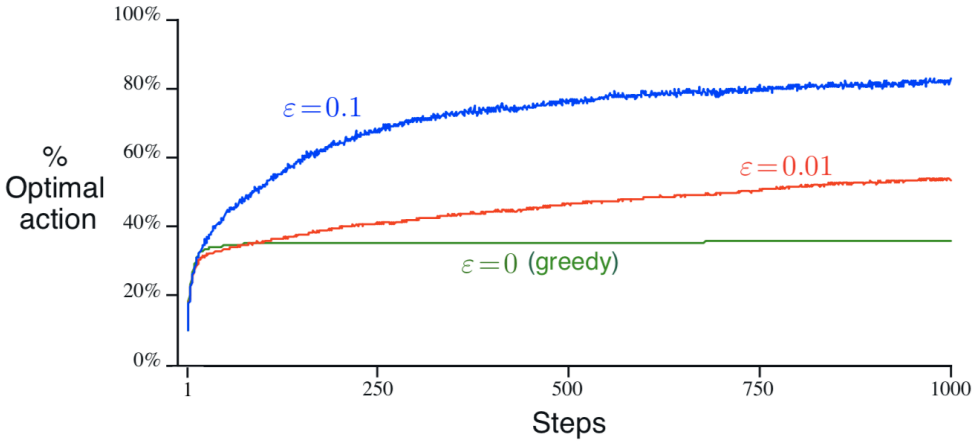
Performance of ε- greedy on 10-armed testbed[2]



Average reward at time t is $\frac{1}{t}\sum_{i=1}^{t} R_t(A_t)$.

[2]Fig 2.2 in [SB18, Sec 2.3].

Performance of ε- greedy on 10-armed testbed[3]



[3]Fig 2.2 in [SB18, Sec 2.3].

- Sutton and Barto ran the ε-greedy algorithms with 3 settings of epsilon on the 2000 bandit problems, each for 1000 steps. The average reward achieved at step t across the 2000 bandit problems is plotted over time. Note that ε-greedy with $\varepsilon = 0.1$ will never choose the optimal action more than about 90% of the time. Or $.9 + \varepsilon/k = .91$ of the time, to be more precise. The pure greedy strategy ($\varepsilon = 0$) gets stuck for lack of exploration.
- As the number of steps goes to infinity, we will eventually have perfect estimates of the expected reward for each action, and so we will choose the optimal action $1 - \varepsilon + \varepsilon/k$ fraction of the time.

References

Resources

- There's a chapter on bandits in Sutton and Barto's book on reinforcement learning [SB18, Sec 2.3], which is worth reading.
- *Bandit Algorithms* by Lattimore and Szepesvári is a relatively new book that is much more theoretical, but a great book if you're into that sort of thing [LS20].

References I

[LS20] Tor Lattimore and Csaba Szepesvári, *Bandit algorithms*, Cambridge University Press, 2020.

[SB18] Richard S. Sutton and Andrew G. Barto, *Reinforcement learning: An introduction*, A Bradford Book, Cambridge, MA, USA, 2018.

[Tho33] William R. Thompson, *On the likelihood that one unknown probability exceeds another in view of the evidence of two samples*, Biometrika **25** (1933), no. 3/4, 285.