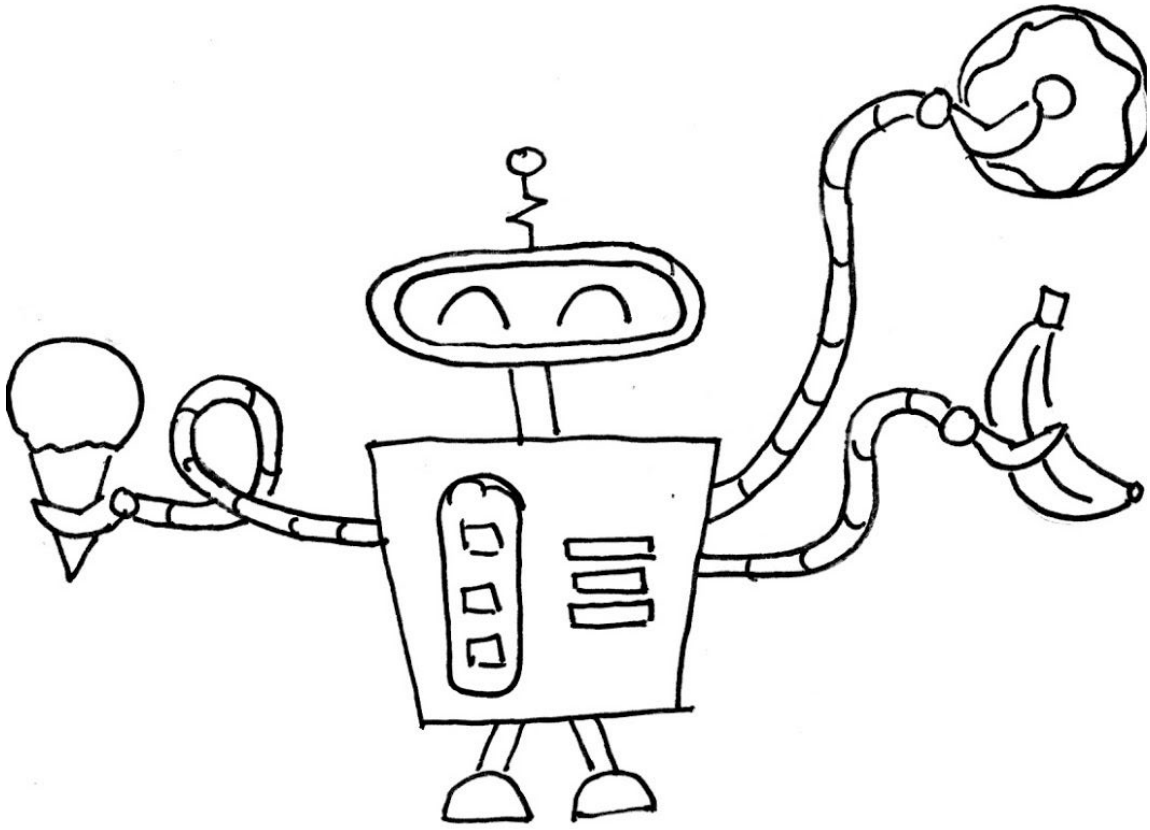
BANDITS



Credits:

Amy Tsai: Artist, Editor, Organizer

Alec Mori: Author

Zine Staff 2018

Table of Contents

Introduction **4**

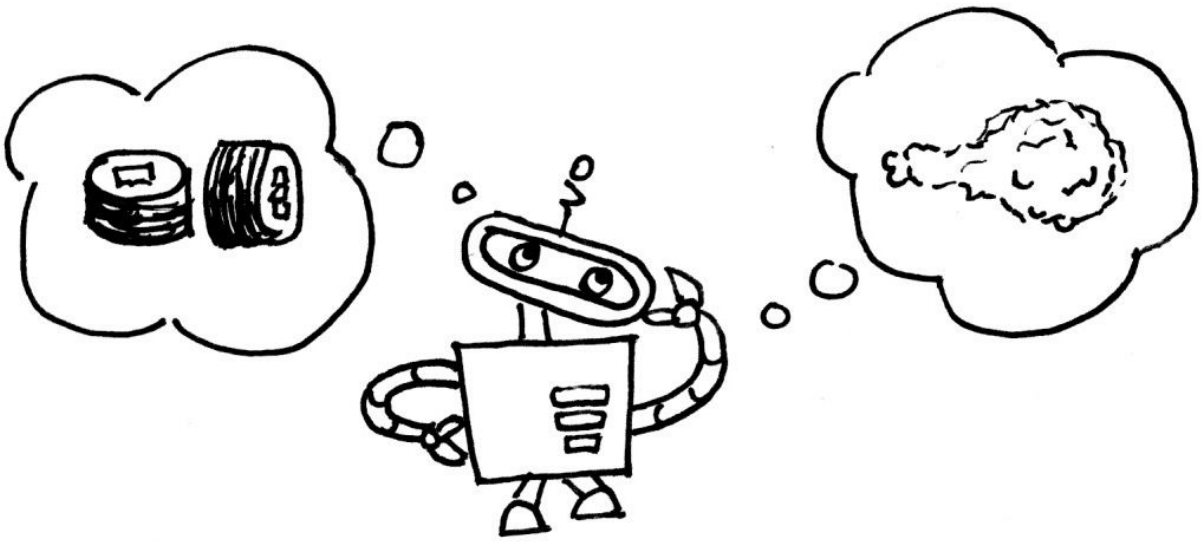
Multi-Armed Bandit **9**

Mortal Bandits **16**

Contextual Bandit **19**

Introduction

Every day we have to make tradeoffs. Do you want to go to the tiny sushi restaurant in the upstairs of a bar, or do you head towards the fried chicken place with the long line? Do you try the new Thor movie on Netflix or re-watch episodes of Bojack Horseman? Given your dozens of Tinder dates and what you learned from them, should you swipe right on this new person that appeared in your area?

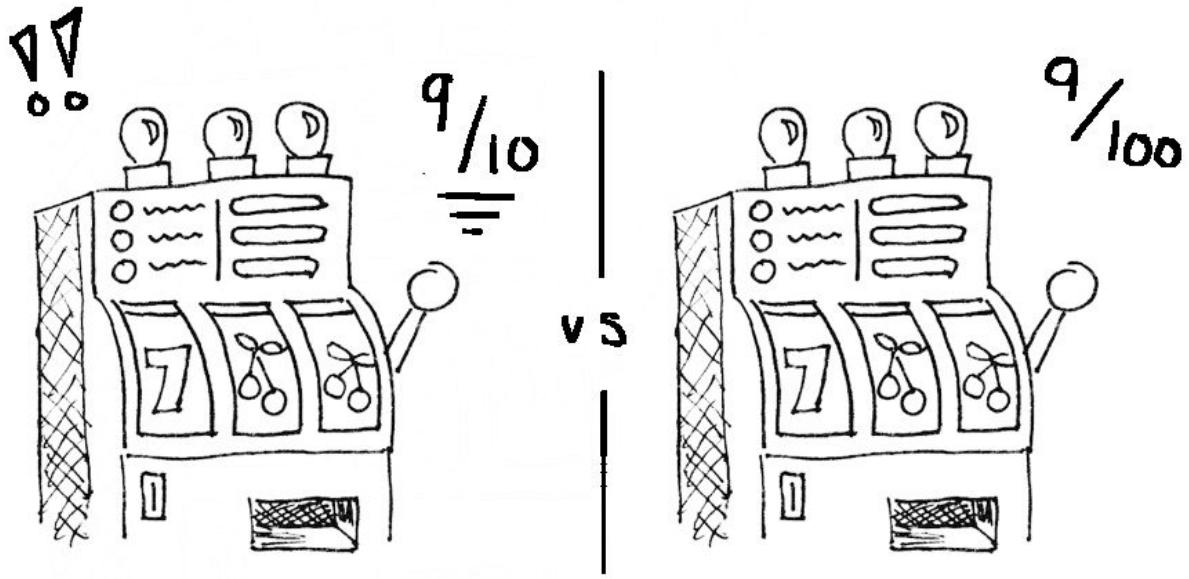


How we answer these questions depends on how we try to strike the balance between **exploration** and **exploitation**. We want to **exploit** the options that give us the most long-term happiness, but we also want to **explore** enough options so that we can be confident that we are **exploiting** the best option.

Scientists have been trying for years to answer this question and for years have been stumped. Fortunately, revelations started emerging after scientists turned their eyes toward a strange source of inspiration: slot machines.

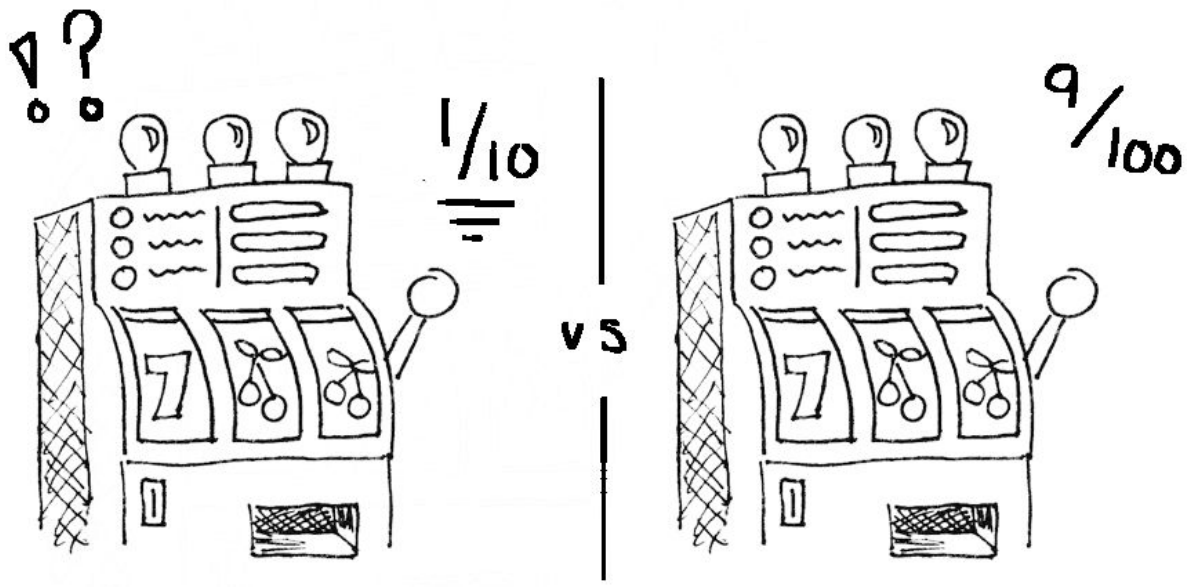
They structured the **explore/exploit** problem as follows: let's say we enter a casino and see all of the slot machines, each with a different chance of giving a payout. How do you decide which machine to pull the most often?

Scenario 1



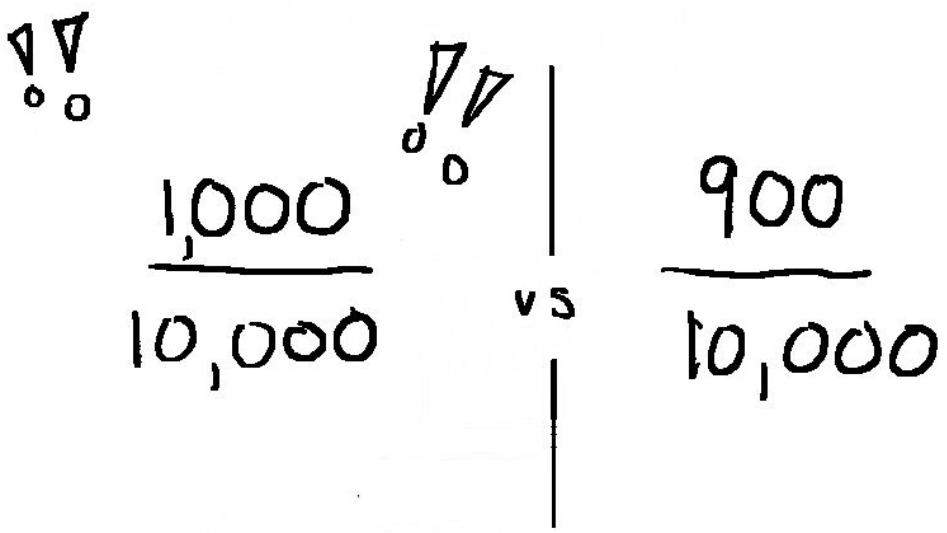
Let's say we had two slot machines. We play the first one ten times and win nine times. We play the second a hundred times and win nine times. Which is better? Most would say the first machine - it wins 90% of the time, while the second only wins 9%.

Scenario 2



What if we played the first slot ten times and won only one time instead of nine? The first machine is still better (10% > 9%) but you probably feel a lot less confident about deciding that machine one is better - it's not just about which machine has the strongest winning record.

Scenario 3



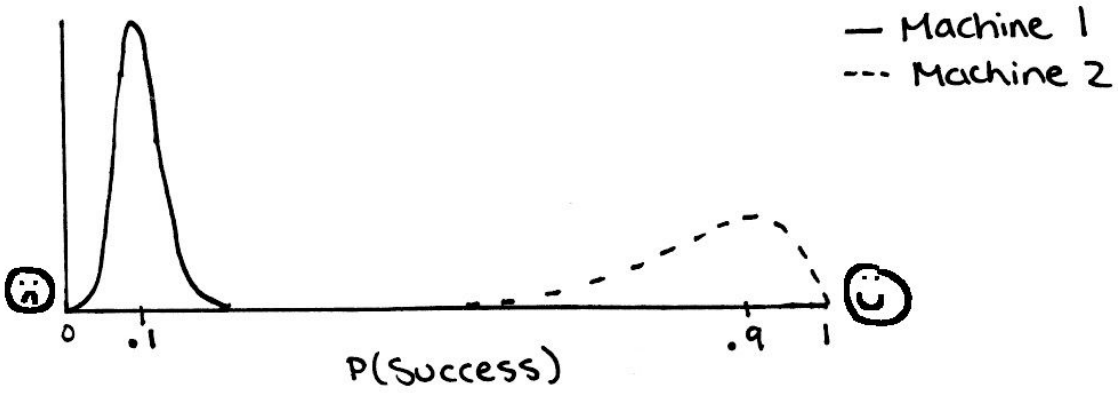
We might feel more confident if instead we played each machine 10,000 times: if the first machine won 1,000 times, and the second won 900, we feel more comfortable accepting that the second machine is in fact worse than the first.

Thinking about the slot machines (or “one-armed bandits”) in this way eventually led to the formulation of the algorithm that solved this problem: the **Multi-Armed Bandit**.

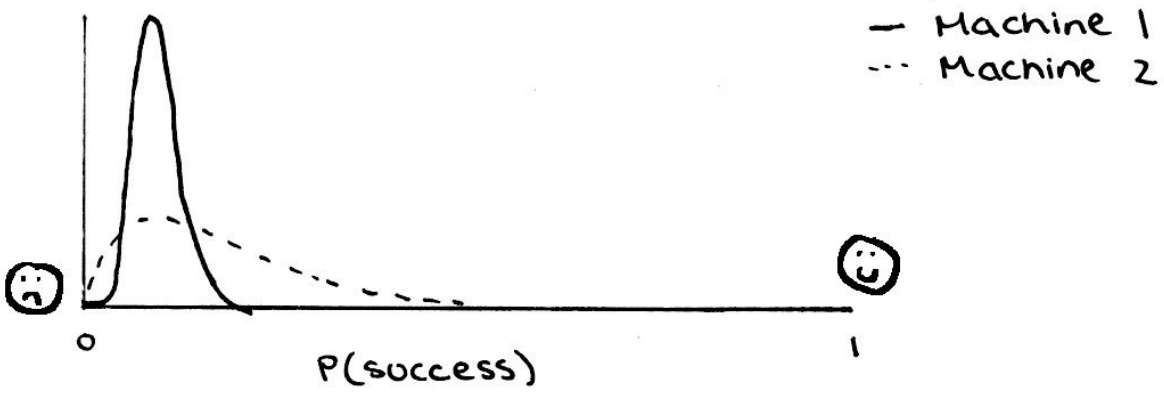
Multi-Armed Bandit

Each of the slot machines above can be described as probability distribution!

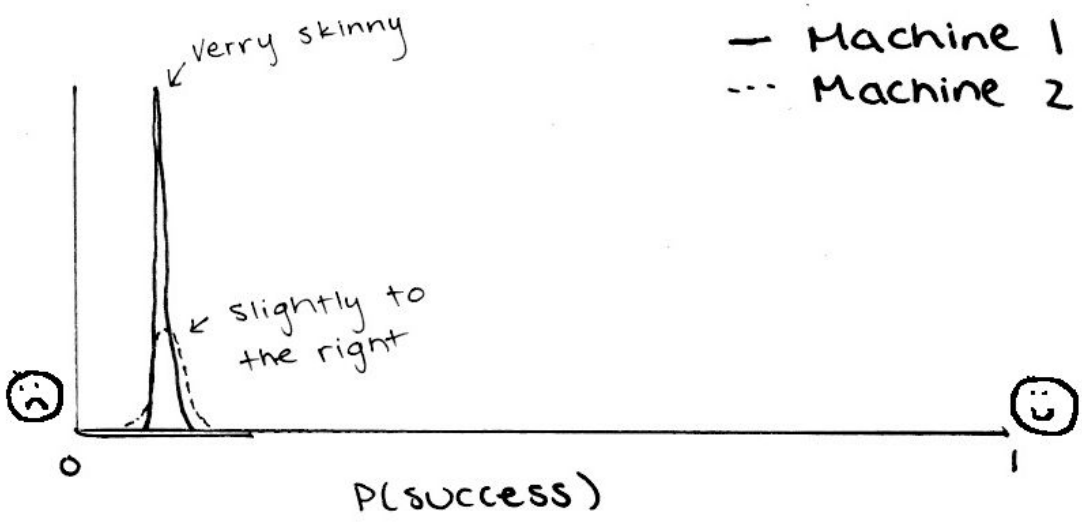
Scenario 1



Scenario 2



Scenario 3

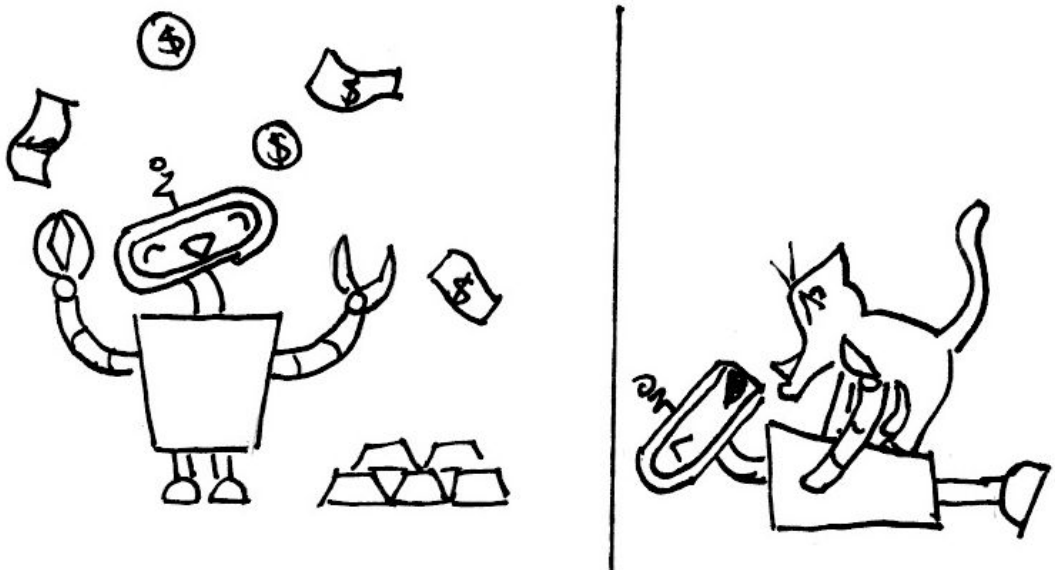


We can set the probability of success as the center of the distribution, and the variance depends on how many times you have tried that machine.

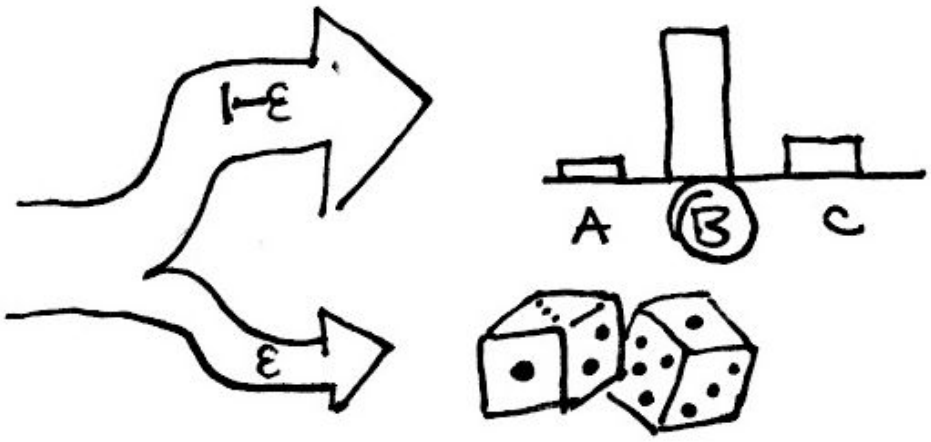
As long as you can turn a scenario into “success” and “failure,” you can use a multi-armed bandit.

- Eating at a restaurant? Success can be having a good experience!
- Showing a bunch of search results? Success can be the user interacting with a particular result!
- Trying a new way to get to work? Success can be if the new route beats the average time of your normal route!

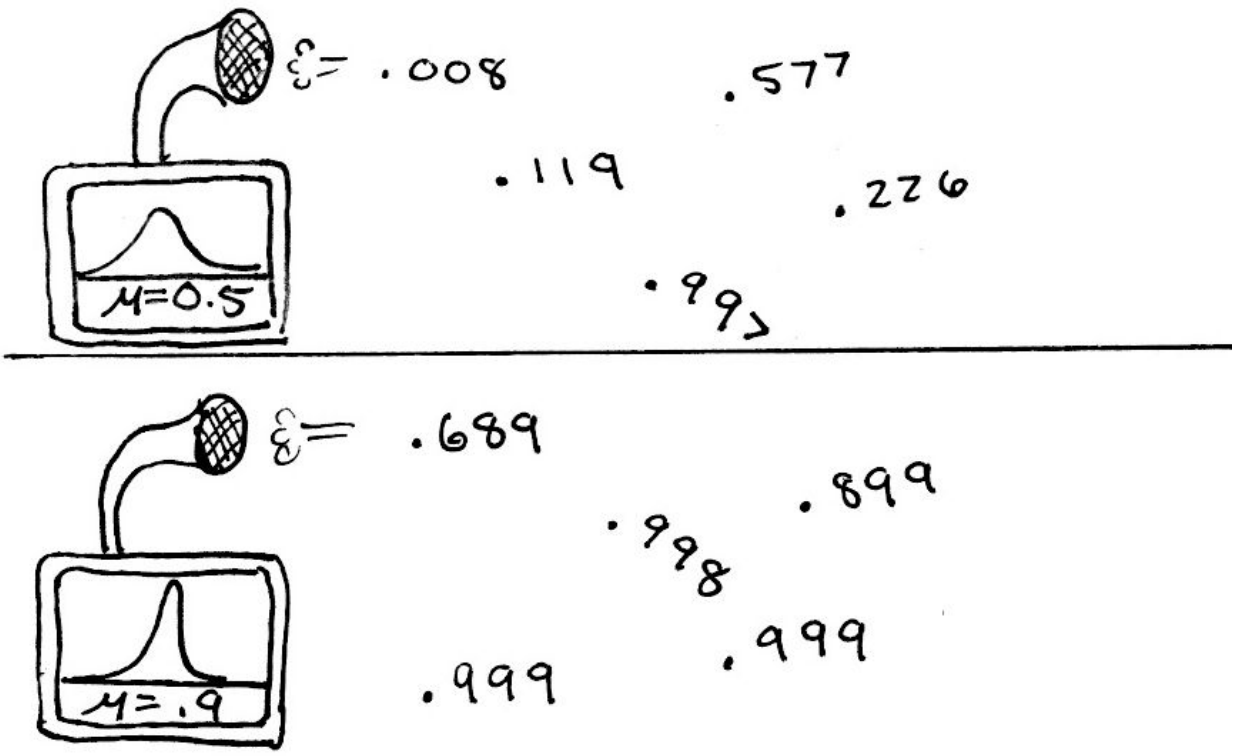
How you decide to define “success” and “failure” is up to you; as long as you can, the multi-armed bandit is yours to use.



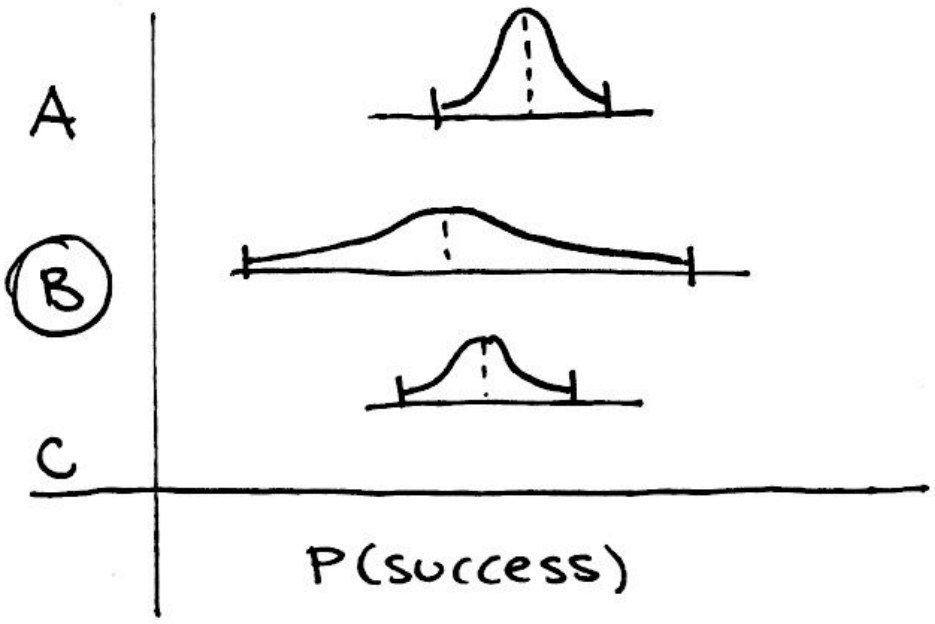
Once you have the distributions for each of your choices, however, you still have to decide which option to choose. There are a few popular algorithms to decide amongst your options, but the most popular are **epsilon-greedy**, **Thompson Sampling**, and **upper confidence bound (UCB)**.



Epsilon-greedy is the most straightforward of all the algorithms - choose the current best option most of the time, and for a small percentage of the time choose randomly from all the options. That way, you always get to try new things, but are constantly **exploiting**.

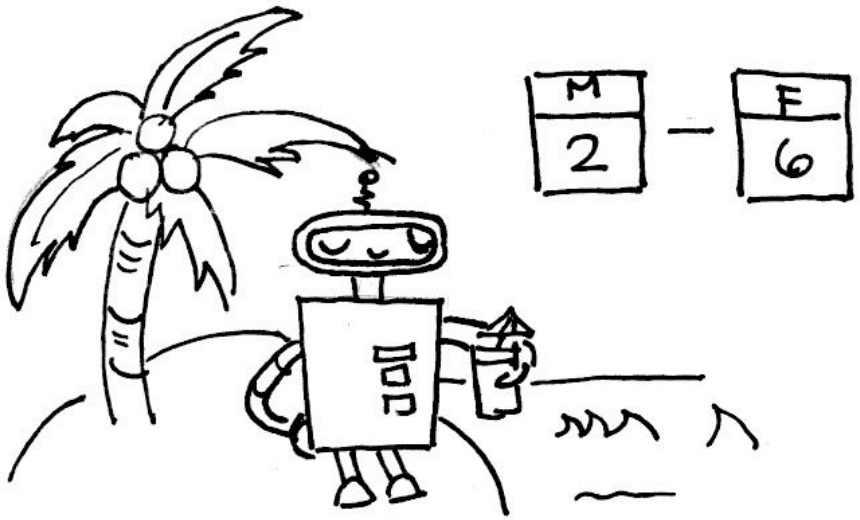


Thompson Sampling is a bit more complicated. From each distribution you sample a number, and choose the distribution that yielded the highest number (best chance of success). If a choice's distribution is really wide (high variance), you are likely to 'accidentally' show that choice, thus learning more information about it. The more you make that choice, the smaller the variance will become. Eventually you will consistently choose numbers around the mean.



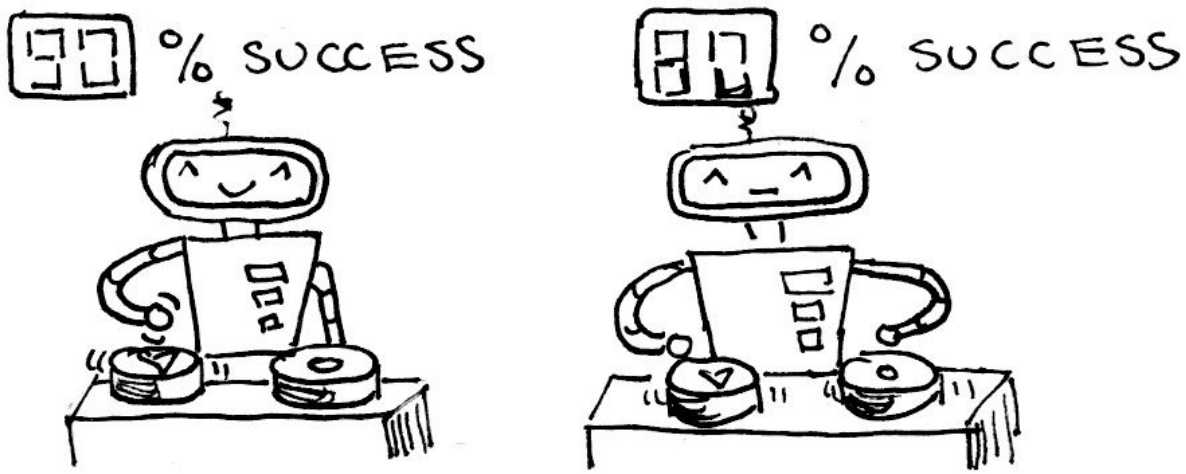
Like **Thompson Sampling**, **UCB** takes how wide the distribution is when deciding what choice to make. For each option you make an overly optimistic guess on what each choices success rate could be, such that most of the choice's distribution is smaller than the value you estimate. The choice with the largest optimistic guess is the option that you go with.

Mortal Bandits

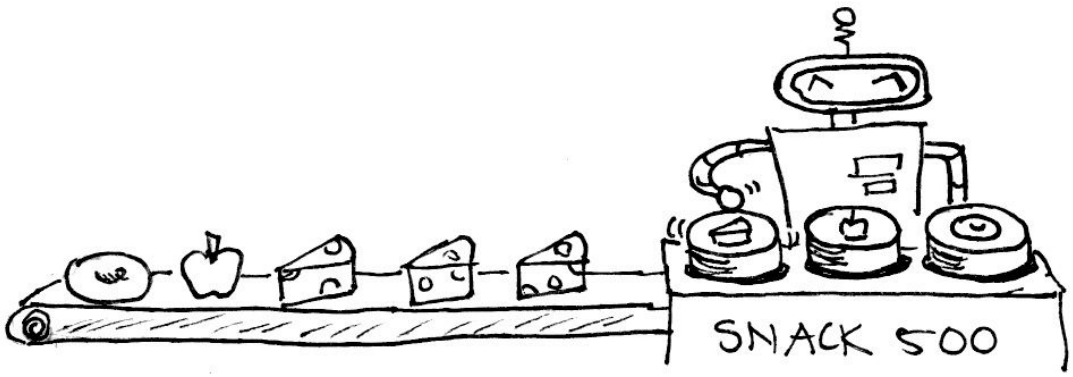


As great as multi-armed bandits are, there constraints on real world problems. There are always limitations on the amount of data that we have available. Maybe you are in trying to find out where to visit on your week-long vacation, or you can only remember the last few trips to restaurants in your area. If it's your last night in town before your vacation ends, there's no real reason to **explore** - you should go to your favorite

restaurant and eat it in your favorite park, as learning more won't help you make future choices. Any time you have a limit to the amount of data you have, you'll want to change how you trade off the balance between **exploring** and **exploiting**.



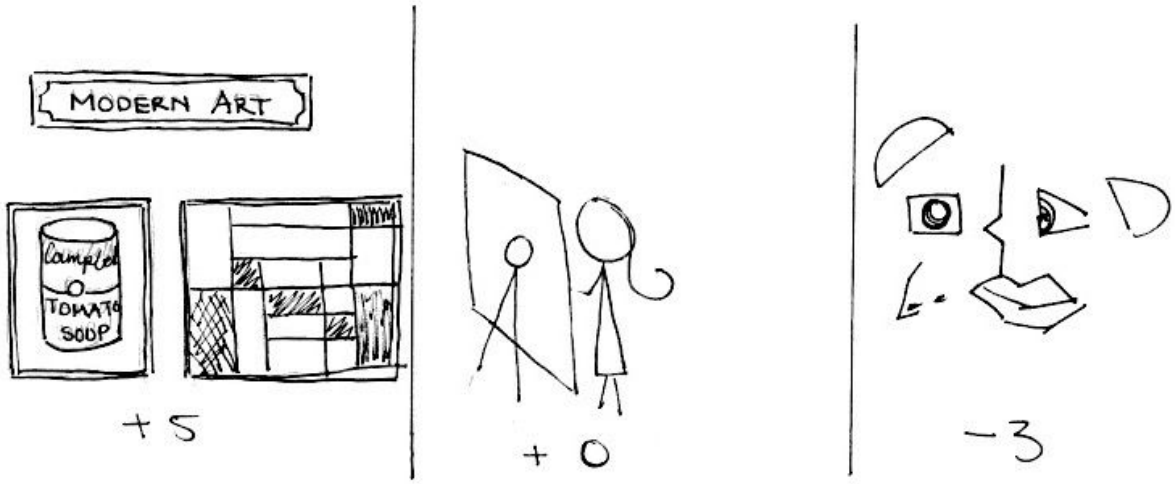
The **Adaptive Greedy** algorithm is similar to **epsilon-greedy**. Most of the time you do pick your favorite choice; however, you also need to set a threshold of success that you are okay with. As long as you are above the threshold, you keep choosing your favorite choice, never randomly **exploring**. If not, you only choose your best choice most of the time. The further you are below the threshold, the less often you choose the strongest option.



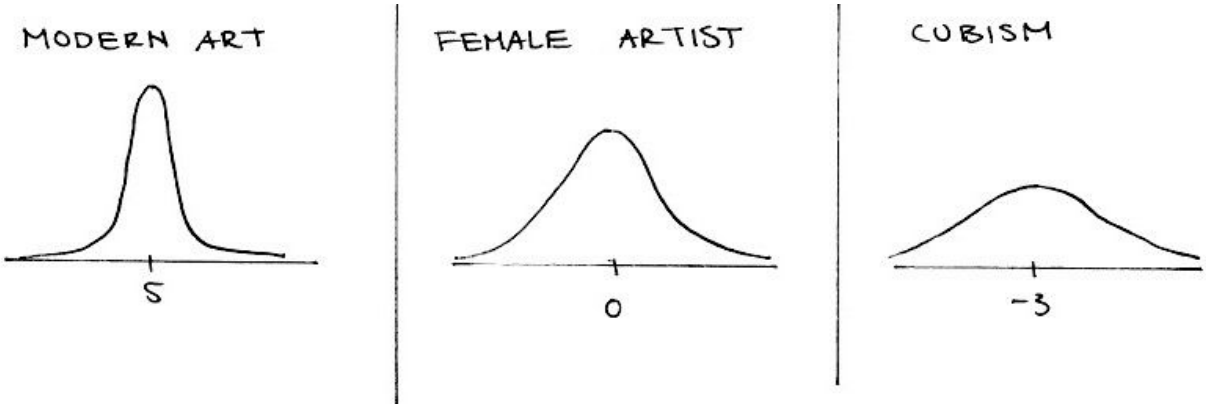
Contextual Bandit

Sometimes you just don't have enough information to learn about all your possible choices. You can't read every book in a library; you can't date everyone on Tinder; you can't try every restaurant in San Francisco. At a certain point, you have stop learning about each individual choice and start generalizing the choices. Instead of learning about "Un Mundo" by Ángeles Santos, you learn about "20th century Cubist work by a female painter." Instead of learning just about this single painting, you can generalize across all modern, all Cubist, and all artwork with female authors.

After seeing hundreds of paintings and marking whether you like them, you can use machine learning to help learn how important each generalization is. For instance, maybe we really like modern art (+5), are neutral towards female painters (+0), and don't like Cubism (-3), which would give "Un Mundo" a score of 2.



What traditional machine learning fails to capture, however, is how confident we are in each of these features. For instance, maybe we saw fifty modern paintings but only two Cubist paintings; we are a lot more confident that we like modern art and less confident that we don't like Cubist paintings.



Much like what we did when constructing the multi-armed bandit, we can build a distribution around each score. Our “modern art” distribution will be a skinny distribution centered around +5, while our “Cubism” distribution would be a wide distribution with a center at -3.

Instead of just adding up the scores, we will instead add samples from each of these distributions, similar to **Thompson Sampling**. That way, we would have the chance to learn about qualities we are less confident about while accurately assessing qualities we are more confident about.