Multi-Arm Bandits

Cary Goltermann

Galvanize

2016

Overview

Multi-Arm Bandit

- Motivation
- Exploration vs. Exploitation
- Formalization
- Regret

Strategies

- Epsilon-Greedy
- UCB1
- Softmax
- Bayesian Bandit

Overview

Multi-Arm Bandit

- Motivation
- Exploration vs. Exploitation
- Formalization
- Regret

Strategies

- Epsilon-Greedy
- UCB1
- Softmax
- Bayesian Bandit

Motivation

Consider the following scenario. (You may have seen something like this before.)

- The company you work for is testing out a new version of it's website.

Motivation

Consider the following scenario. (You may have seen something like this before.)

- The company you work for is testing out a new version of it's website.
- After running an A/B test for an afternoon the new version of the site appears to performing slightly better than the old version.

Motivation

Consider the following scenario. (You may have seen something like this before.)

- The company you work for is testing out a new version of it's website.
- After running an A/B test for an afternoon the new version of the site appears to performing slightly better than the old version.
- After running the test over night, the new version of the site is performing better than the old version with a statistically significant p-value of 0.04.

Motivation

Consider the following scenario. (You may have seen something like this before.)

- The company you work for is testing out a new version of it's website.
- After running an A/B test for an afternoon the new version of the site appears to performing slightly better than the old version.
- After running the test over night, the new version of the site is performing better than the old version with a statistically significant p-value of 0.04.

Do you stop the test, or do you keep running it?

Overview

Multi-Arm Bandit

- Motivation
- Exploration vs. Exploitation
- Formalization
- Regret

Strategies

- Epsilon-Greedy
- UCB1
- Softmax
- Bayesian Bandit

Exploration vs. Exploitation

When we are trying to determine which option, from a potential pool, is the best option what we are faced with is an information gathering challenge.

Exploration vs. Exploitation

When we are trying to determine which option, from a potential pool, is the best option what we are faced with is an information gathering challenge.

- **Exploration**: Testing out the different options (of the website), to determine how good each one is. Acquiring more knowledge about the reward associated with the options.

Exploration vs. Exploitation

When we are trying to determine which option, from a potential pool, is the best option what we are faced with is an information gathering challenge.

- **Exploration**: Testing out the different options (of the website), to determine how good each one is. Acquiring more knowledge about the reward associated with the options.
- **Exploitation**: Leveraging your current knowledge about the options to get the highest expected reward at that time.

Traditional A/B Testing

A/B testing in terms of exploration vs. exploitation:

- Start with **pure exploration**, in which the same number of users are assigned to see each of the options. This is the testing phase.

Traditional A/B Testing

A/B testing in terms of exploration vs. exploitation:

- Start with **pure exploration**, in which the same number of users are assigned to see each of the options. This is the testing phase.
- Once the test is complete and some conclusion has been reached you switch to **pure exploitation**. In this phase all of the users see the version that chosen as a result of the test.

Potential Inefficiencies

- Need to wait for the experiment to conclude for certain statistical guarantees to be provided.

Potential Inefficiencies

- Need to wait for the experiment to conclude for certain statistical guarantees to be provided.
- Only after the experiment concludes can we capitalize on a potentially better option.

Potential Inefficiencies

- Need to wait for the experiment to conclude for certain statistical guarantees to be provided.
- Only after the experiment concludes can we capitalize on a potentially better option.
- This wastes time - and money - showing users the suboptimal site.

Multi-Arm Bandit Approach

- Show each user the site that you think is best **most** of the time.

Multi-Arm Bandit Approach

- Show each user the site that you think is best **most** of the time. (The definition of most will be dictated by your strategy. These will be discussed shortly.)

Multi-Arm Bandit Approach

- Show each user the site that you think is best **most** of the time. (The definition of most will be dictated by your strategy. These will be discussed shortly.)
- As the experiment runs and you send users to different sites, update you beliefs about each site.

Multi-Arm Bandit Approach

- Show each user the site that you think is best **most** of the time. (The definition of most will be dictated by your strategy. These will be discussed shortly.)
- As the experiment runs and you send users to different sites, update you beliefs about each site.
- Run until a clear best site emerges.

Depending on your strategy you can balance **exploration** and **exploitation** instead of having to choose either one or the other.

Origins

- This problem was originally motivated from the problem that a gambler faces when deciding which slot machine (individually called one-armed bandits) to play at.

Origins

- This problem was originally motivated from the problem that a gambler faces when deciding which slot machine (individually called one-armed bandits) to play at.
- Assuming that the gambler wants to be smart about their strategy they are faced, not only, with the choice of which machines to play, how many times they should play them and in what order.

Origins

- This problem was originally motivated from the problem that a gambler faces when deciding which slot machine (individually called one-armed bandits) to play at.
- Assuming that the gambler wants to be smart about their strategy they are faced, not only, with the choice of which machines to play, how many times they should play them and in what order.
- All they know is that each bandit will provide a reward when its lever is pulled according to its individual (and static) distribution.

Origins

- This problem was originally motivated from the problem that a gambler faces when deciding which slot machine (individually called one-armed bandits) to play at.
- Assuming that the gambler wants to be smart about their strategy they are faced, not only, with the choice of which machines to play, how many times they should play them and in what order.
- All they know is that each bandit will provide a reward when its lever is pulled according to its individual (and static) distribution.

The gambler's objective, then, is to **maximize** the sum of the rewards they receive through a series of lever pulls.

Use Cases

- Dynamic A/B Testing.
- Budget allocation amongst competing projects.
- Clinical trials.
- Adaptive routing in attempts to minimize network delays.
- Reinforcement learning.

Fun Fact

"Originally considered by Allied scientists in World War II, it proved so intractable that, according to Peter Whittle, the problem was proposed to be dropped over Germany so that German scientists could also waste their time on it."

— Wikipedia

Overview

Multi-Arm Bandit

- Motivation
- Exploration vs. Exploitation
- Formalization
- Regret

Strategies

- Epsilon-Greedy
- UCB1
- Softmax
- Bayesian Bandit

Terminology

The model is given by a set of real distributions $B = \{R_1, ..., R_k\}$, where each distribution is associated with a reward delivered by one of the $K \in \mathbb{N}^+$ levers.

Terminology

The model is given by a set of real distributions $B = \{R_1, ..., R_k\}$, where each distribution is associated with a reward delivered by one of the $K \in \mathbb{N}^+$ levers.

We will define the mean values associated with each of these distributions as $\mu_1, ..., \mu_k$.

Terminology

The model is given by a set of real distributions $B = \{R_1, ..., R_k\}$, where each distribution is associated with a reward delivered by one of the $K \in \mathbb{N}^+$ levers.

We will define the mean values associated with each of these distributions as $\mu_1, ..., \mu_k$.

The agent plays one lever per round and observes the associated reward. The goal of the agent is to maximize the sum of the collective rewards, or alternatively, minimize the **regret**.

Overview

Multi-Arm Bandit

- Motivation
- Exploration vs. Exploitation
- Formalization
- Regret

Strategies

- Epsilon-Greedy
- UCB1
- Softmax
- Bayesian Bandit

Regret

The regret, ρ that an agent experiences after T rounds is the difference between the reward expected reward sum associated with the optimal strategy and the sum of the rewards collected.

$$\rho = T\mu^* - \sum_{t=1}^{T} \hat{\rho}_t$$

μ^* Maximal reward mean, $\mu^* = max_k\{\mu_k\}$.

$\hat{\rho}_t$ The reward at time t.

Regret

The regret, ρ that an agent experiences after T rounds is the difference between the reward expected reward sum associated with the optimal strategy and the sum of the rewards collected.

$$\rho = T\mu^* - \sum_{t=1}^{T} \hat{\rho}_t$$

μ^* Maximal reward mean, $\mu^* = max_k\{\mu_k\}$.

$\hat{\rho}_t$ The reward at time t.

Regret can, therefore, be seen as a measure of how often you choose the suboptimal bandit. We can view this as a cost function we are trying to minimize.

Zero-Regret Strategy

A **zero-regret strategy** is defined as one who's average regret per round, ρ/T, goes to zero in the limit where the number of rounds, T goes to infinity.

Zero-Regret Strategy

A **zero-regret strategy** is defined as one who's average regret per round, ρ/T, goes to zero in the limit where the number of rounds, T goes to infinity.

The interesting thing is that a zero-regret strategy does *not* guarantee that you will never choose a suboptimal outcome. Instead it guarantees that, as you continue to play you will tend to choose the optimal outcome.

Overview

Multi-Arm Bandit

- Motivation
- Exploration vs. Exploitation
- Formalization
- Regret

Strategies

- Epsilon-Greedy
- UCB1
- Softmax
- Bayesian Bandit

Epsilon-Greedy

What if we always want to explore a certain amount?

Epsilon-Greedy

What if we always want to explore a certain amount?

- **Explore** with some probability ϵ, often chosen as 10%.
- All other times, **exploit**, a.k.a. choose the bandit with the best performance so far $\rightarrow \underset{j=1,\ldots,K}{argmax} \{\hat{\mu}_j(t)\}$.

  $\hat{\mu}_j(t)$ Our best guess for μ_j at round t.
- Update our knowledge about $\mu' s$ after each round.

Epsilon-Greedy

What if we always want to explore a certain amount?

- **Explore** with some probability ϵ, often chosen as 10%.
- All other times, **exploit**, a.k.a. choose the bandit with the best performance so far $\rightarrow \underset{j=1,\ldots,K}{argmax}\{\hat{\mu}_j(t)\}$.

  $\hat{\mu}_j(t)$ Our best guess for μ_j at round t.
- Update our knowledge about $\mu's$ after each round.

Is this a zero-regret strategy?

Overview

Multi-Arm Bandit
- Motivation
- Exploration vs. Exploitation
- Formalization
- Regret

Strategies
- Epsilon-Greedy
- UCB1
- Softmax
- Bayesian Bandit

UCB1 (Upper Confidence Bound)

The UCB1 algorithm is greedily chooses the bandit with the highest $\hat{\mu}(t)$ but with a clever additional factor that automatically balances exploration and exploitation.

UCB1 (Upper Confidence Bound)

The UCB1 algorithm is greedily chooses the bandit with the highest $\hat{\mu}(t)$ but with a clever additional factor that automatically balances exploration and exploitation.

The choice at round t, after each lever is pulled once for a baseline, is made as follows:

$$\underset{j=1,\ldots,K}{argmax} \left\{ \hat{\mu}_j(t) + \sqrt{\frac{2ln(t)}{n_j}} \right\}$$

$\hat{\mu}_j(t)$ Best guess for the μ_j at time t.

n_j Number of times that bandit j has been pulled.

t Number of rounds that have been played in total.

Is this a zero-regret strategy?

Overview

Multi-Arm Bandit

- Motivation
- Exploration vs. Exploitation
- Formalization
- Regret

Strategies

- Epsilon-Greedy
- UCB1
- Softmax
- Bayesian Bandit

Softmax

The softmax creates a probability distribution over all the levers in proportion to how good we think each lever is.

Softmax

The softmax creates a probability distribution over all the levers in proportion to how good we think each lever is.

$$P_t(choosing\ bandit\ j) = \frac{e^{\hat{\mu}_j(t)/\tau}}{\sum\limits_{j=1}^{K} e^{\hat{\mu}_j(t)\tau}}$$

τ, tau
: Temperature, controls the "randomness" of the distribution. Usually chosen around 0.001.

Softmax

The softmax creates a probability distribution over all the levers in proportion to how good we think each lever is.

$$P_t(\textit{choosing bandit } j) = \frac{e^{\hat{\mu}_j(t)/\tau}}{\sum\limits_{j=1}^{K} e^{\hat{\mu}_j(t)\tau}}$$

τ, tau
: Temperature, controls the "randomness" of the distribution. Usually chosen around 0.001.

At each round the above distribution is sampled from and that bandit is chosen.

Softmax

The softmax creates a probability distribution over all the levers in proportion to how good we think each lever is.

$$P_t(\textit{choosing bandit } j) = \frac{e^{\hat{\mu}_j(t)/\tau}}{\sum_{j=1}^{K} e^{\hat{\mu}_j(t)\tau}}$$

τ, tau
: Temperature, controls the "randomness" of the distribution. Usually chosen around 0.001.

At each round the above distribution is sampled from and that bandit is chosen.

A member of probability matching algorithms, so-called for their property of creating the probability distribution.

Overview

Multi-Arm Bandit

- Motivation
- Exploration vs. Exploitation
- Formalization
- Regret

Strategies

- Epsilon-Greedy
- UCB1
- Softmax
- Bayesian Bandit

Bayesian Bandit

We can use the Bayesian beta-binomial conjugate prior techniques used to model the click-through rate in the morning exercise to as our base model for each of the bandits.

This is another probability matching algorithm where we have a separate model for each of the bandits.

Bayesian Bandit

We can use the Bayesian beta-binomial conjugate prior techniques used to model the click-through rate in the morning exercise to as our base model for each of the bandits.

This is another probability matching algorithm where we have a separate model for each of the bandits.

Process

- Sample from the distributions for each bandit.
- Choose the bandit with the highest sampled μ
- Update the distribution of the chosen bandit with the knowledge gained from choosing it.

Bayesian Bandit