Machine Learning Systems Design

Deployment and Monitoring

Lecture 23: Model Online Evaluation

CE 40959 Spring 2023
Ali Zarezade
SharifMLSD.github.io

Agenda

1. **Test in Production**

1. Test in Production

Why do online evaluation of models?

- Is my model working well in production env?
- How often should I retrain my models?

Monitoring and model online evaluation

If monitoring means passively keeping track of the outputs of whatever model is being used, test in production means proactively choosing which model to produce outputs so that we can evaluate it.

The **goal** of both monitoring and online evaluation is to *understand a model's performance* and figure out *when to update* it

Test in production

Two major test types for offline evaluation:

- **Test splits**: are usually static and have to be static so that you have a trusted benchmark to compare multiple models
- **Backtests**: method of testing a predictive model on data from a specific period of time in the past

But, why offline evaluation isn’t enough?

Test in production

- Test split: if you update the model to adapt to a new data distribution, it’s not sufficient to evaluate this new model on test splits from the old distribution.
- Backetest: because of data distributions shift, the fact that a model does well on the data from the last hour doesn’t mean that it will continue doing well on the data in the future.

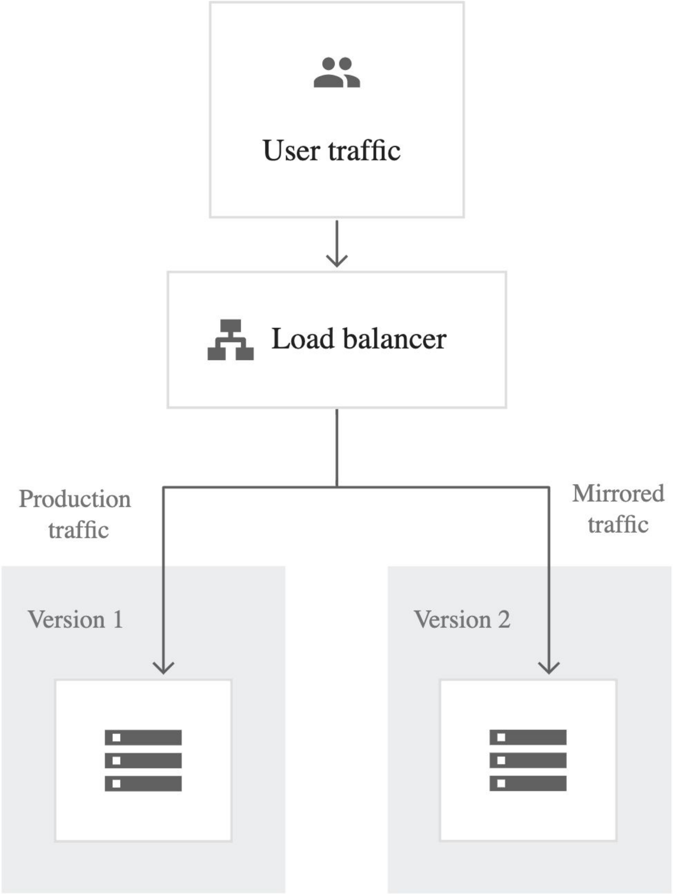
Test in production

The only way to know whether a model will do well in production is to deploy it. Some techniques to help you evaluate your models in production (mostly) safely.

- Shadow testing
- A/B testing
- Canary testing
- Interleaving experiments
- Bandits testing

Shadow testing

1. Deploy the candidate model in parallel with the existing model.
2. For each incoming request, route it to both models to make predictions, but only serve the existing model's prediction to the user.
3. Log the predictions from the new model for analysis purposes.



Application deployment and testing strategies (Google)

Shadow testing

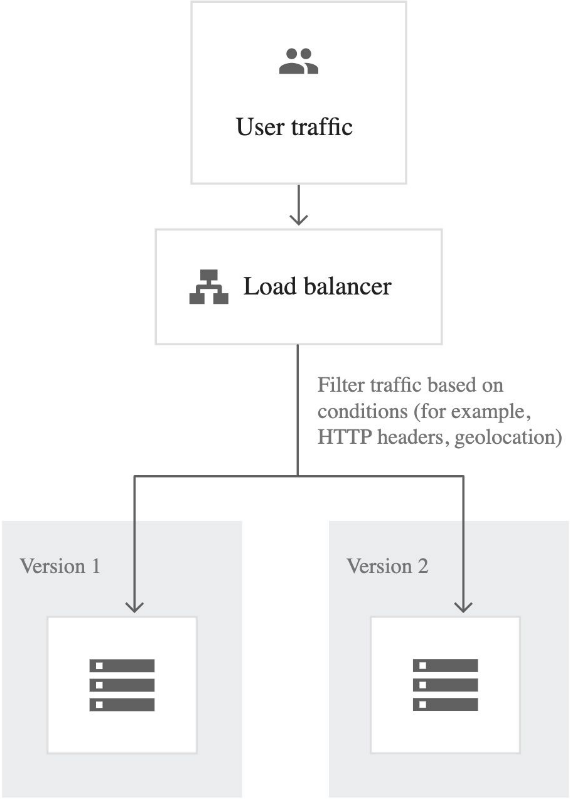
- Pros: the risk of this new model doing something funky is low
- Cons: it’s expensive, doubling your inference compute cost.

A/B testing

- An A/B test is usually formulated to answer the following question: “Does the new model lead to a statistically significant change in this specific business metric?”
- Statistical hypothesis testing maintains a null hypothesis and an alternative hypothesis.
- The null hypothesis states that the new model doesn’t change the average value of the business metric.
- The alternative hypothesis states that the new model changes the average value of the metric.

A/B testing

- New model alongside existing system
- A percentage of traffic is routed to new model based on routing rules
- Control target audience & monitor any statisticall significant differences in user behavior
- Can have more than 2 versions

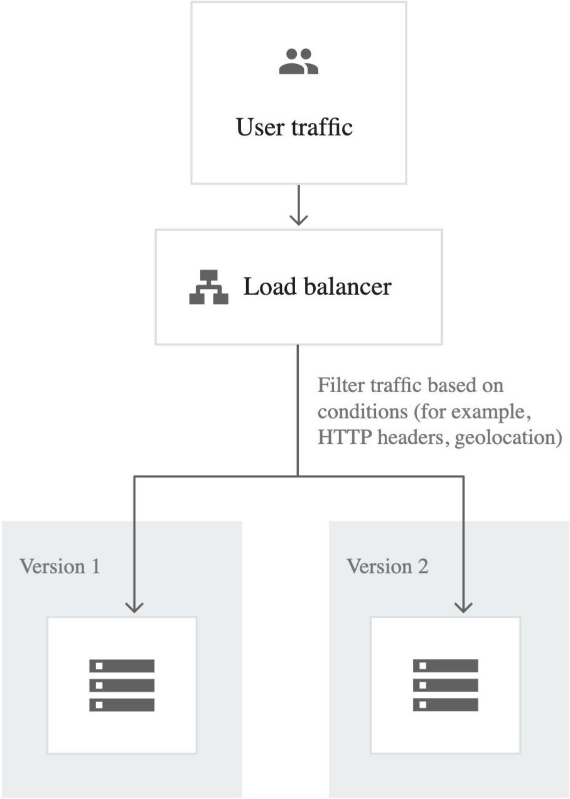

User traffic
Load balancer
Filter traffic based on conditions (for example, HTTP headers, geolocation)
Version 1
Version 2

Application deployment and testing strategies (Google)

A/B testing

- New model alongside existing system
- A percentage of traffic is routed to new model based on routing rules
- Control target audience & monitor any statisticall significant differences in user behavior
- Can have more than 2 versions

In cases that one model predictions affect another one (e.g. ride sharing) serve models in separate days

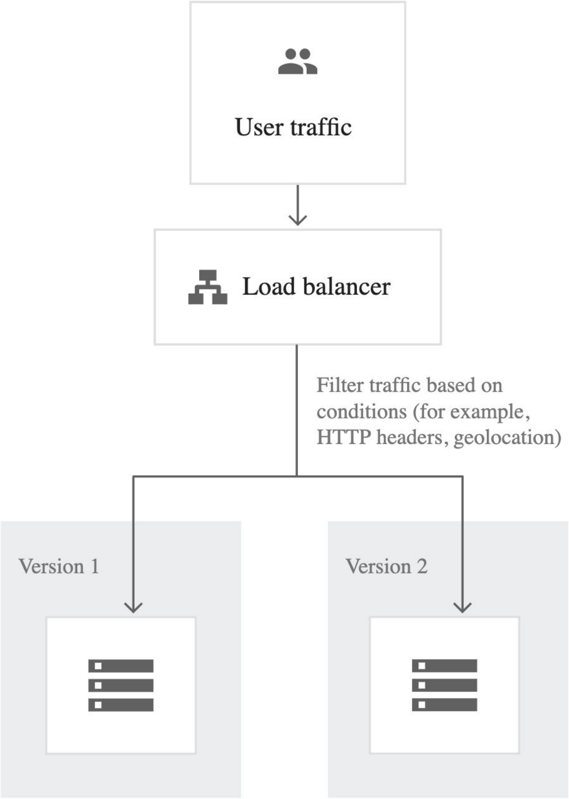

User traffic
Load balancer
Filter traffic based on conditions (for example, HTTP headers, geolocation)
Version 1
Version 2

Application deployment and testing strategies (Google)

A/B testing

- New model alongside existing system
- A percentage of traffic is routed to new model based on routing rules
- Control target audience & monitor any statisticall significant differences in user behavior
- Can have more than 2 versions

It’s possible to do A/B testing with more than two variants, which means we can have A/B/C testing or even A/B/C/D testing.


User traffic
Load balancer
Filter traffic based on conditions (for example, HTTP headers, geolocation)
Version 1
Version 2

Application deployment and testing strategies (Google)

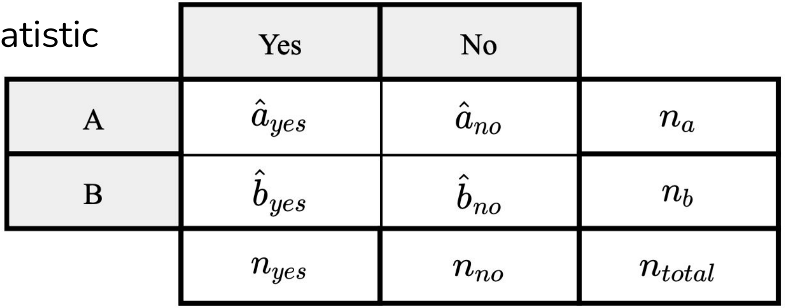
A/B testing

To do A/B testing the right way requires doing many things right:

- Traffic routed to each model has to be truly random (e.g. no selection bias)
- Run on statistically significant number of samples to gain enough confidence (A/B testing uses statistical hypothesis testing such as two-sample tests)

A/B testing: G-test

- The first formulation of A/B test is based on the G-test. It is appropriate for a metric that counts the answer to a “yes” or “no” question.
  - Whether the user bought the recommended article?
  - Whether the user has spent more than $50 during a month?
  - Whether the user renewed the subscription?
- To apply G-test for A/B testing, we need to:
  - Define a yes/no question that reflects our metric
  - Randomly assign users to group A (old version) or group B (new version)
  - Record the answer of each user as yes or no
  - Fill the following table and calculate the G-statistic



	Yes	No	
A	$\hat{a}_{yes}$	$\hat{a}_{no}$	n_a
B	$\hat{b}_{yes}$	$\hat{b}_{no}$	n_b
	n_{yes}	n_{no}	n_{total}

A/B testing: G-test

- Find the expected numbers of “yes” and “no” answers for A and B:

$$a_{yes} \overset{\text{def}}{=} n_a \, \frac{n_{yes}}{n_{total}},$$
$$a_{no} \overset{\text{def}}{=} n_a \, \frac{n_{no}}{n_{total}},$$
$$b_{yes} \overset{\text{def}}{=} n_b \, \frac{n_{yes}}{n_{total}},$$
$$b_{no} \overset{\text{def}}{=} n_b \, \frac{n_{no}}{n_{total}}.$$

- find the value of the G-test as:

$$G \overset{\text{def}}{=} 2\left(\hat{a}_{yes}\ln\left(\frac{\hat{a}_{yes}}{a_{yes}}\right) + \hat{a}_{no}\ln\left(\frac{\hat{a}_{no}}{a_{no}}\right) + \hat{b}_{yes}\ln\left(\frac{\hat{b}_{yes}}{b_{yes}}\right) + \hat{b}_{no}\ln\left(\frac{\hat{b}_{no}}{b_{no}}\right)\right).$$

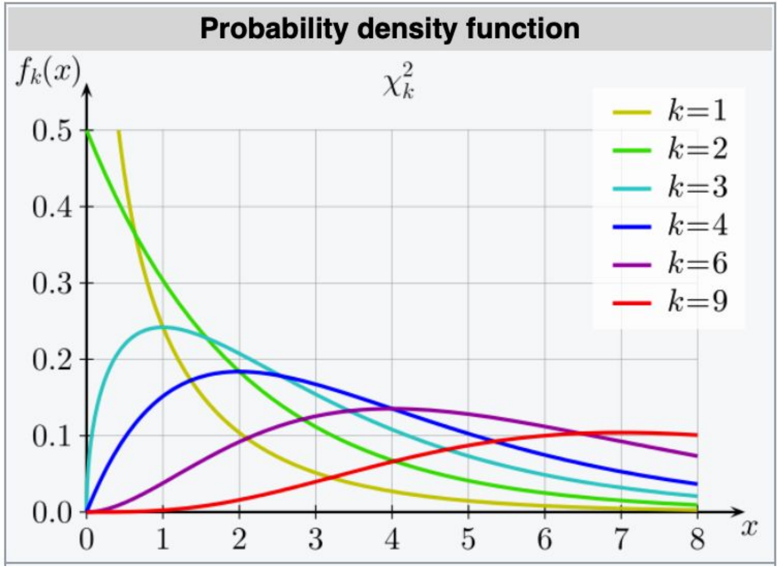
A/B testing: G-test

- G is a measure of how different the samples from A and B are.
- Under the null hypothesis (A and B are equal), G follows a chi-square distribution with one degree of freedom:

$$G \sim \chi^2_1$$

- In other words, if A and B were equal, we expect G to be small.

chi-squared



A/B testing: G-test

- For example, imagine you calculated G = 3.84. If A and B were equal the probability of observing G ≥ 3.84 is about 5% (p-value=0.05).
- If the p-value is small enough (e.g., below 0.05) then the performances of the new and the old model are very likely different.
- Statistically, the result of the G-test is valid if we have at least 10 “yes” and “no” results in each of the two groups

A/B testing: Z-test

- The second formulation of A/B test applies when the question for each user is, “How many?” or, “How much?”
    - How much time a user has spent on the website during a session?
    - How much money a user has spent during a month?
    - How many news articles a user has read during a week?
- Lets measure the time a user spends on a website where our model is deployed. Users are routed to versions A (old model) and B (new model) of the website.
- The null hypothesis is that users of both versions spend, on average, the same amount of time.
- The alternative hypothesis is that they spend more time on website B than on website A.

A/B testing: Z-test

- First compute sample mean and sample variance for A and B:

$$\hat{\sigma}_A^2 \overset{\text{def}}{=} \frac{1}{n_A} \sum_{i=1}^{n_A} (\hat{\mu}_A - a_i)^2, \quad \hat{\mu}_A \overset{\text{def}}{=} \frac{1}{n_A} \sum_{i=1}^{n_A} a_i,$$

$$\hat{\sigma}_B^2 \overset{\text{def}}{=} \frac{1}{n_B} \sum_{j=1}^{n_B} (\hat{\mu}_B - b_j)^2. \quad \hat{\mu}_B \overset{\text{def}}{=} \frac{1}{n_B} \sum_{j=1}^{n_A} b_j,$$

- ❖ nA is the number of users routed to version A and nB is the number of users routed to version B.
- ❖ ai and bj is the time spent on the website by, respectively, users i and j

- Then, the value of the Z-test is then given by:

$$Z \overset{\text{def}}{=} \frac{\hat{\mu}_B - \hat{\mu}_A}{\sqrt{\frac{\hat{\sigma}_B^2}{n_B} + \frac{\hat{\sigma}_A^2}{n_A}}}.$$

A/B testing: Z-test

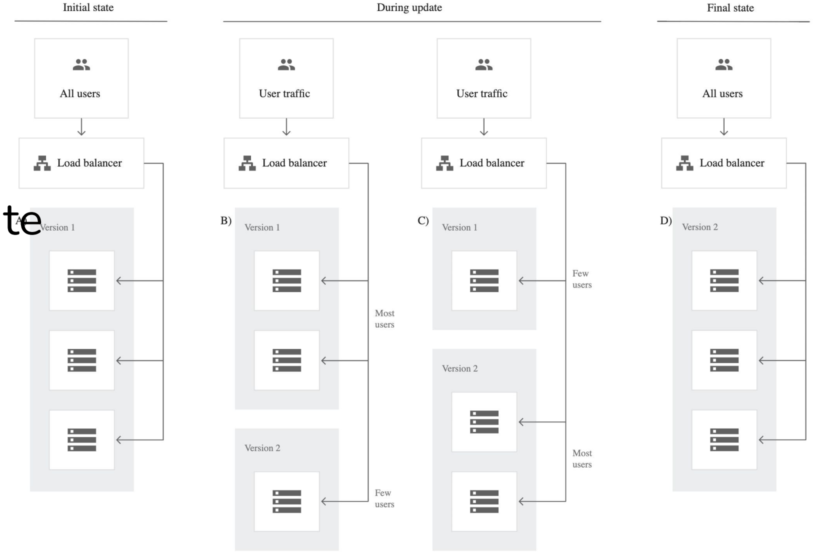
- Under the null hypothesis (i.e. A and B are equivalent), Z approximately follows a standardized normal distribution

$$Z \approx \mathcal{N}(0,1)$$

- The larger Z, the more likely the difference between A and B is significant
- For example, let’s imagine your sample gave you Z = 2.64. If A and B were equal, the probability of observing Z ≥ 2.64 is about 5% (p-value=0.05).
- If the p-value is less than or equal to 0.05, then we reject the null hypothesis.

Canary testing

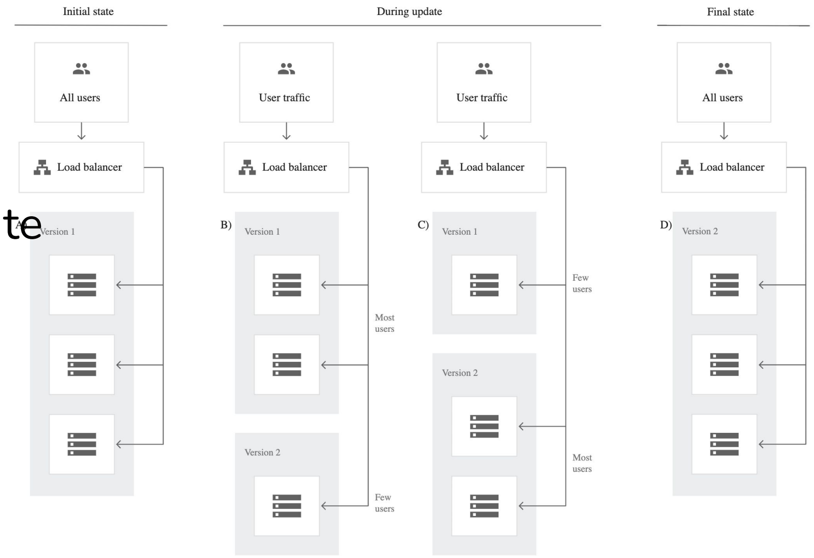
- Deploy the candidate model (canary) alongside the existing model.
- A portion of the traffic is routed canary.
- If its performance is satisfactory, increase canary traffic. If not, abort the canary and route all the traffic back to the existing model.
- Stop when either the canary serves all the traffic or when the canary is aborted.



Automated Canary Analysis at Netflix with Kayenta

Canary testing

- Deploy the candidate model (canary) alongside the existing model.
- A portion of the traffic is routed canary.
- If its performance is satisfactory, increase canary traffic. If not, abort the canary and route all the traffic back to the existing model.
- Stop when either the canary serves all the traffic or when the canary is aborted.

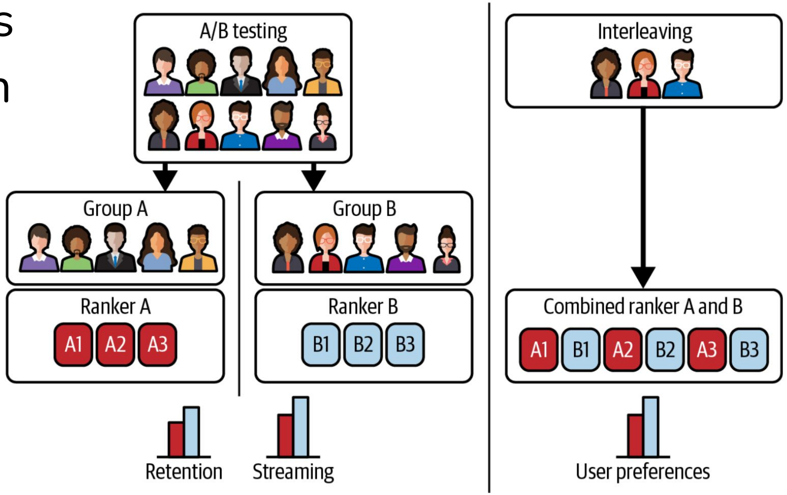


Canary releases can be used to implement A/B testing.
However, you can do canary analysis without A/B testing

[Automated Canary Analysis at Netflix with Kayenta](#)

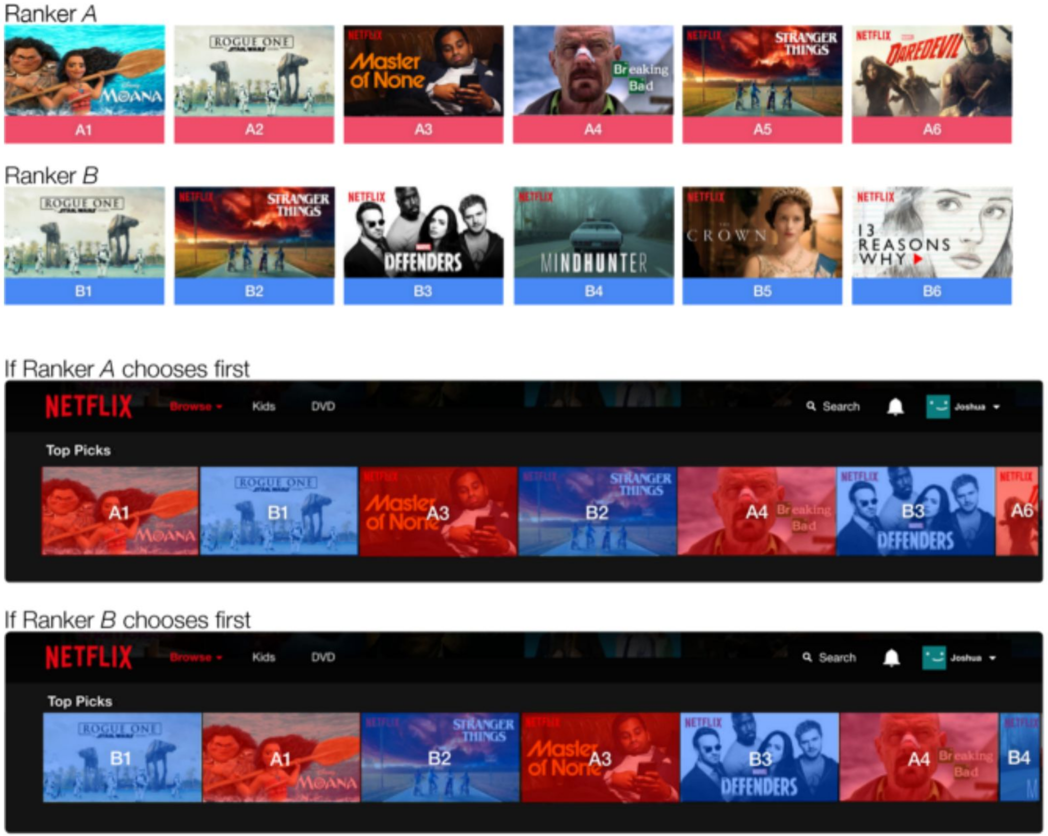
Interleaved experiments

- Especially useful for ranking/recsys
- Take recommendations from both models A & B
- Mix them together and show them to users
- See which recommendations are clicked on



Interleaved experiments

In experiments, Netflix found that interleaving “reliably identifies the best algorithms with considerably smaller sample size compared to traditional A/B testing



Bandits for online evaluation

- Your model must be able to make online predictions.
- Preferably short feedback loops on whether a prediction is good or not to update the payoff of each model quickly.
- A mechanism to collect feedback, calculate and keep track of each model's performance, and route prediction requests to different models based on their current performance.

Bandits for online evaluation

- Bandits are well-studied in academia and have been shown to be a lot more data- efficient than A/B testing (in many cases, bandits are even optimal)
- In an experiment by Google, A/B testing required over 630,000 samples to get a confidence interval of 95%, whereas a simple bandit algorithm requires less than 12,000 samples.
- However, bandits are a lot more difficult to implement than A/B testing because it requires computing and keeping track of models' payoffs.
- Two of the most popular exploration algorithms are Thompson Sampling and Upper Confidence Bound (UCB).

Contextual bandits as an exploration strategy

- If bandits for model evaluation are to determine the payout (i.e., prediction accuracy) of each model, contextual bandits are to determine the payout of each action.
- In the case of recommendations/ads, an action is an item/ad to show to users, and the payout is how likely it is a user will click on it.
- Contextual bandits are algorithms that help you balance between showing users the items they will like (exploit) and showing the items that you want feedback on (explore).

Machine Learning Systems Design

Deployment and Monitoring

Next Lecture: Model Online Evaluation

CE 40959 Spring 2023
Ali Zarezade
SharifMLSD.github.io