Running example: You are charged with making a website for your company. You have two candidate designs. How would you pick between them?

Ad-hoc approach:

- Pick the one you like best.
- Pick the one your manager likes best.
- Advantage: Happy manager, quick decision.
- Drawback: Unclear whether you or your manager is representative of the average visitor to your site.

Data-driven approach:

- Try both out, for a month each. Pick the one that performs the best.
    - What do we mean ‘best’?
    - To make question more concrete, consider the *widescope site*.
    - What metrics could you define?
    - **Metrics:** *Fraction of serious submissions, Total submissions, Repeat visits...*
- Advantage: Data-driven, simple. No cost other than analysis cost, if you happened to have deployed the two site designs in the past.
- Drawback
    - It could be that the type of visitors to the site differ from one month to the next, accounting for the difference in performance between the two candidates.
    - E.g.: Suppose you deploy candidate A over summer and candidate B in fall. And it happens that Econ professors are busy teaching in fall and so the summer has more serious submissions.

A/B Experiment:

- Need a controlled experiment
    - Basic idea: Randomize traffic into the two candidates A and B. This way, any difference between the behavior of A and B must be due to the differences between A and B and not some other factor like the traffic pattern; *subject to noise*
    - Dealing with **noise**: How many samples would you need to be sure that candidate A is better than candidate B?
    - Need some baseline for how good these sites are. Suppose we believe that both candidates have a rate of at least 0.1, and we only care if the difference between them is 10%.
        - Worst-case one has value 0.1 and the other has value 0.11
    - How many samples would you need so that:
        - If true rate for candidate B is >= 0.11, we would detect it with probability 0.95.
        - If true rate for candidate B is < 0.11, we would detect it with probability 0.05

        - In **R**: power.prop.test(p1 = .11, p2 = .10, power = .95, sig.level=0.05)
        - Try it online
            - ~24,000 samples in each arm; at 2000 visits a day, 24 days
    - Advantages: Highly defensible decision based on science.
    - Disadvantage: costly, takes time, takes expertise
    - Arguably the evaluation framework for many internet based companies
        - Internet conducive to experimentation
        - Experiments lead to objective decisions
        - Objective decisions reduce the role of ‘managers’
        - Combined this with a good incentive structure
        - Easier for new employees to have an impact on company, its users
        - Does this apply to Apple? Intel? Microsoft?
        - Search engine’s innovation speed depends on the #of experiments it can perform.
            - Overlapping experiments: Nice Google paper.
            - See Figure 2 of this paper:
            - Traffic experiments v/s Cookie experiments
            - Launch layer

    - Issues beyond noise/statistical significance: Nice Microsoft Research paper
        - Picking metrics: queries and revenue were deemed to be metrics to increase . A ranking change caused both to increase, but was bad. Why?
        - Logging issues: A piece of javascript was added to a page, was expected to slow the page down, but caused an increase in clicks. Why?
        - An experiment shows a strong positive trend initially, and later flattens out. Why?

- **Multiarmed Bandit** (what problem does this solve?)
    - Algorithm(informal):
        - Have prior distribution on the rate of each arm (candidates). Both arms have the same prior. So, each arm is equally likely to be the best.
        - In each round, pick arm in proportion to probability that it is the best arm
        - Update prior based on observed data
        - Do this until one arm is the best with overwhelming probability, or both arms are very comparable
    - Advantage:
        - More traffic diverted to better arm sooner.
        - Experiment stops sooner. In terms of power analysis, much easier to separate two arms with rates that are far apart, than ones that are near each other. This could be achieved by using confidence intervals too.
    - Disadvantage: needs a well defined objective function

- **Regression/Machine learning**
    - Why would candidate A beat candidate B for *all* users?
    - Presumably, we are better off picking A or B *based* on the user.
    - Sites have new users all the time. So cannot just run the experiment per user, even ignoring noise issues.
    - Need features that are identifiable for *every* user. Need an scheme to pick candidate based on features.
    - Example of features: browser, region
    - Is user_id a valid feature?
    - Suppose we have already seen 10 users from 5 regions and 2 browsers with every possible combination of user and region. And we build a model on browser, region. There is no signal! Signals must repeat!
    - Suppose that we pick between using only browser and only region.
    - If we had a lot of data, we could see which classifies better
    - Without a lot of data, we divide set into training and test sets
        - Build model on training set, evaluate on test set. Prevents overfitting.
    - Clearly, helps to combine region and browser too:
        - Use regularization to constrain model complexity, and improve generalization.

Fun puzzle: Can you identify which part of the lecture this applies to?

Feynman puzzle: Given n dishes (rated 1 .. n) and m <= n (meals to be eaten at the restaurant), how many new dishes d should you try before switching to ordering the best of them for all the remaining (m–d) meals, in order to maximize the average total ratings of the dishes consumed?

Point out when lecture arrives at a similar issue!

Solution:

Key is to find expected value of the best solution given that you explore for ‘d’ meals.

1. $P[i\ is\ best] = Binom(i-1,\ d-1)/Binom(n.d)\ if\ d\ <=\ i$
2. $E[value\ of\ best\ dish] = \sum_{i=d}^{n} i * Binom(i-1,\ d-1)/Binom(n,d)$
3. $E[value\ of\ best\ dish] = (n+1) * d/(d+1)$

Now to pick ‘d’ optimally:

$(n+1) * d/2 \ + \ (m-d) \ * \ (n+1) * d/(d+1)$

Differentiate wrt d, and set to zero:

$d \ = \ \sqrt{2 * (m+1)} - 1$