Date: Thursday, April 25, 2019

Issues with A/B testing

- A/B Testing for User-Interaction services
  - User-interaction: key goal is to get users to interact in some way.
- Traditional A/B testing allocation can lead to issues in contamination as well as frustration.
- Testing inference is no longer valid in this scenario.

Possibilities for fixing the problem?

1) Allocate over time: Version A is shown to everyone for 2 weeks and then version B is shown to everyone for the next 2 weeks.
2) Divide users into A and B based on geography/location.
3) Have feature to be an opt-in feature (user's choice to use new feature).

↳ Challenges of the above three strategies:

Choice One

1) Confounding factor of time (holidays, summer?, etc)
2) User behaviour and the actual base may change over time (e.g. lots of users successfully match during some parts of the test)
3) Bias on B from already seeing A.
4) Showing everyone a new feature is incredibly risky because it could be a failure. And, there is plenty of competition to scoop users.

Choice Two

1) May not be feasible to define geographic boundaries. (A user's willingness to "cross borders" is not well known.
2) People who travel — move between locations

3) Different user behaviour in each location. (Test may not be representative of the population.

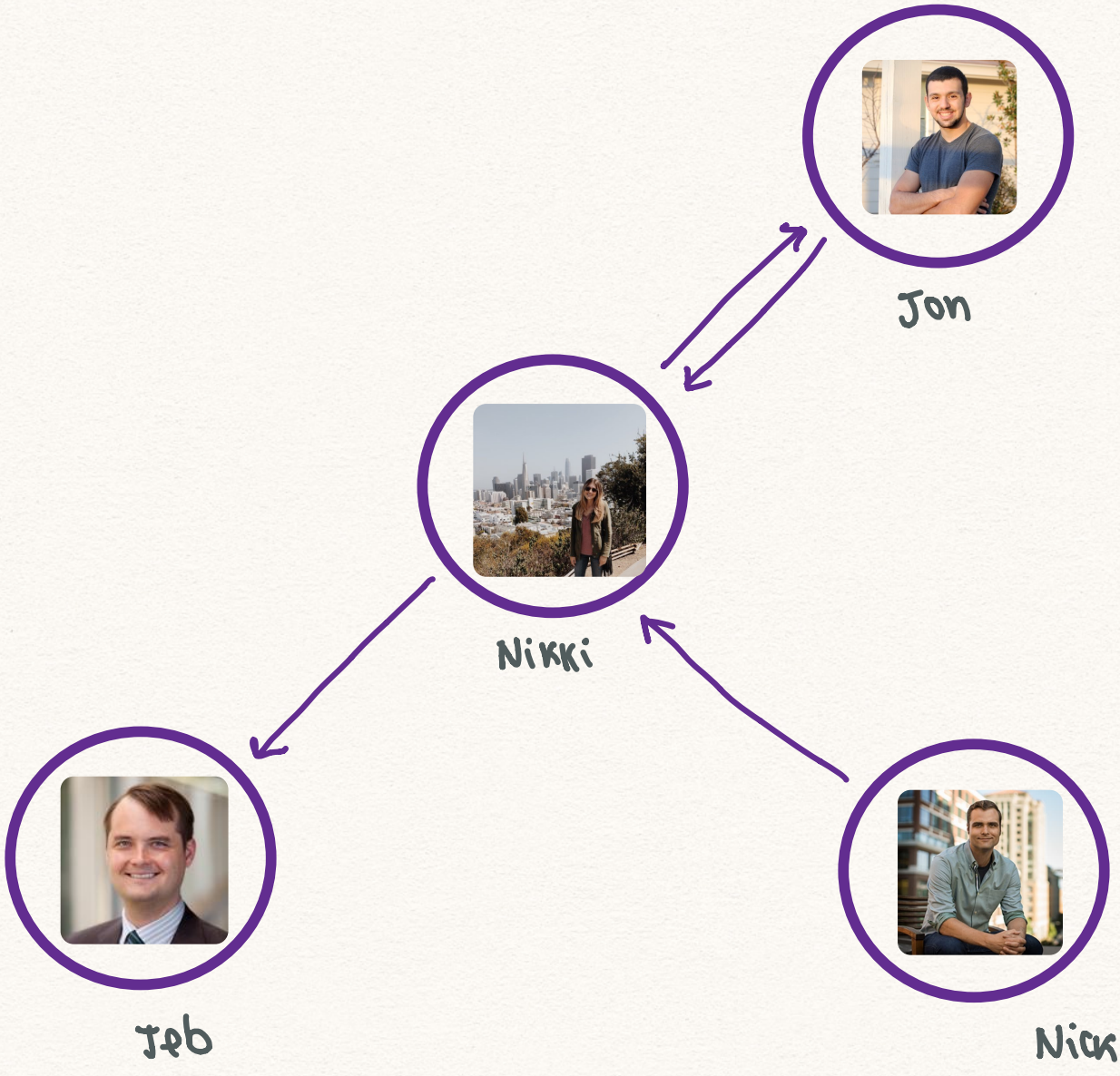
4) Not able to seperate by country b/c the service is not international — hard to get enough sample size.

Choice Three

1) Opt-in: Selection bias. Users who "opted-in" chose to do so. Thus, these users are different already than those who chose not to opt-in. Inference cannot be made on the entire population.

User-interaction services:

↳ Users can be described with a network(s)



In a network, users are represented by nodes/vertices and interaction are represented by edges.

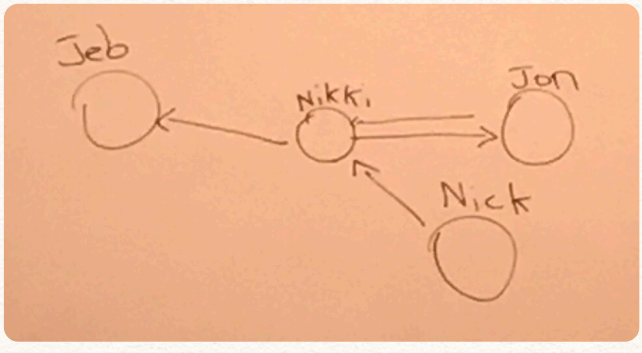
Often, there are many types of interaction describing the relationships between users, which lead to many networks describing the users.

Example: Dating app: 2 graphs

1) Edges are weighted by compatibility score.

2) Edges represent interactions on the site.

- Per-user random assignment does not work on a network! Problems of contamination and faulty inference.
  ↳ in networks, users are dependent!



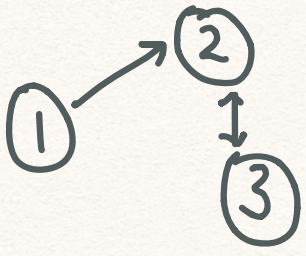
- The dependence of users in a network lead to faulty inferene when allocation is done in a per-user basis.
- Our aim then is to try to eliminate this dependence.

Per-community random assignment:

- Idea is to segment users into communities from the network. (Clustering on networks)
- then allocate users according to communities.

How to work with networks?

Visual structure



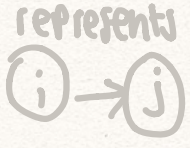
Adjacency matrix

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array}$$

Sparcity edge list

(i,j)

$$\begin{pmatrix} 1,2 \\ 2,3 \\ 3,2 \end{pmatrix}$$



The example of community slide:

- Users: users of an online dating app.
- Edges: measuring similarity of interests between users.
- Colors: Represent communities
- Nodes: Represent users.
  - ↳ Size of nodes: Represents overall # of edges coming from that node.