Online Labor Markets III: Mechanism Design in Online Platforms

March 29, 2017

Guest Speaker: Yash Kanoria, Columbia Business School

1 Intro and Overview

These notes cover three different working papers:

- (with Daniela Saban) "Facilitating the search for partners on matching platforms: Restricting Agent Actions"
- (with Itai Ashlagi, Mark Braverman, and Peng Shi) "Communication Requirements and Informative Signaling in Matching Markets"
- (with Ramesh Johari and Vijay Kamble) "Matching while Learning"

At the time these notes were written, all three papers are available on Yash's website. To obtain good outcomes in matching markets, it is important to know agents' preferences over potential matches. These papers all analyze situations in which acquiring this information is somehow costly or difficult.

2 Facilitating search for partners on matching platforms

2.1 Model

This paper considers a model of dynamic matching where agents on both sides of the market arrive online. Agent i's utility for matching with agent j, denoted u_{ij}, is iid for each i and j. Match utilities are unknown to everyone in the system initially, and must be discovered by paying a screening cost which is uniform across each side of the market (we'll let c_e denote employers' cost for screening workers, and c_w the cost for workers to screen employers). Each agent has a Poisson clock, which determines when they are allowed to act in the market. When an agent i's clock ticks:

- They decide whether to (costlessly) ask to view an agent on the other side of the market, selected uniformly at random (which is without loss, as agents are ex ante indistinguishable). Call this agent j.
- i decides whether or not to pay their cost to screen j.
- Regardless of i's screening decision, they decide whether or not to propose to j.

- j then views i and decides whether or not to pay their cost to screen i.
- Regardless of j's screening decision, they decide whether or not to accept i's proposal.
- If j accepts a proposal by i, both agents leave the market. Otherwise, both agents stay.

The space of interventions considered in the above market is:

- Prohibiting one or both sides of the market from screening.
- Prohibiting one side of the market from proposing.

Importantly, the platform cannot influence the supply of individuals on either side of the market, or observe utilities to suggest matches. There are no payments. The objects of study are steady-state equilibria in the above markets.

2.2 Examples

- Airbnb: no screening prohibited (both renters and hosts can read each others' profiles) but hosts cannot propose to renters.
- Airbnb Instant Book: only renters may screen (hosts automatically accept renters), and hosts cannot propose.
- Uber/Lyft: screening prohibited (riders and drivers are matched without browsing potential matches), riders propose.
- Tinder (for heterosexual matches): no screening prohibited (men and women can view each others' profiles) and both sides can propose.
- Bumble (for heterosexual matches): no screening prohibited, but only women can propose.
- Upwork: both sides can screen, both sides can propose.

Several examples above are similar platforms (or the same platform) experimenting with interventions of the form considered in this paper. When do these interventions yield better equilibria?

2.3 Warmup Results

Allowing both sides to screen and propose and increasing the screening costs yields the following forms for the equilibria:

- With low screening costs, equilibrium takes the form where one side of the market screens and proposes, and the other side screens and accepts (with proposals and acceptance subject to some threshold policy on match utilities).
- With high screening costs, proposers cease to screen.
- With extremely high screening costs, both sides cease to screen.

2.4 Unbalanced Markets

Assume the arrival rate of workers is much higher than the arrival rate of employers. This market is now *unbalanced*. The paper observes the following:

- The workers tend to be less selective.
- There exist bad equilibria where the workers are also doing the proposing. The market ends up losing a lot of welfare from the inability of these workers to profitably screen.
- The solution to these bad equilibria is to prevent the less selective agents from proposing. This is an intuitive explanation for the success of Bumble.

A similar phenomenon occurs with unbalanced search costs: there exist bad equilibria where the agents with high costs propose, which can be eliminated with a similar intervention. Intuitively, these interventions serve to reduce screening costs incurred by agents.

The paper also considers a modification to the model where one side is heterogeneous in the sense that there are two different distributions for match qualities (e.g. "good" workers and "bad" workers).

3 Communication Requirements and Informative Signaling in Matching Markets

This paper considers matching markets with nonstrategic agents. It considers the communication requirements for computing a stable matching. How much information about agents' preference orderings (in the form of choice queries) does the mechanism need to extract before it can reach a stable matching?

3.1 Previous Work

The model for this paper is the standard stable matching model: each side of the market consists of n agents with strict preference orderings over the other side. In is model, Gonczarowski et al. (2015) prove:

Theorem 1. *(Informal) There exists a distribution over preferences such that every matching protocol which finds a stable matching with probability at least 2/3 must on average make* $\Omega(n)$ *choice queries per agent.*

This is a notable result because simply asking each agent for their full preference ordering can be done with $O(n \log n)$ queries. In a sense, Deferred Acceptance is optimal up to a logarithmic factor if you allow arbitrary distributions over preferences.

Real-life markets are so big that $\Omega(n)$ queries is prohibitive. What can be done?

3.2 Separable Markets

This paper considers a model where workers have arbitrary preferences over firms, but firm j has a random utility for work i drawn iid from the following distribution:

$$u_{ji} = a_{ji} + \epsilon_{ji}, \tag{1}$$

Where a_{ji} is public information, known to the algorithm, and ϵ_{ji} is private noise, drawn from a distribution with bounded hazard rate, and with a_{ji}s sufficiently "spread out." The authors reason that this is a closer approximation to real markets, where some candidates are "obviously" better than others. The paper gives an algorithm which requires $\tilde{O}(\sqrt{n})$ choice queries per agents, and gives a matching lower bound instance.

3.3 Communication-Efficient Deferred Acceptance

The high-level idea for the algorithm is to simulate deferred acceptance, but to keep workers from proposals to employers that they "obviously" cannot get. Slightly more formally, the algorithm works as follows:

1. Each firm j signals which agents i have high idiosyncratic utility ϵ_{ji}. (i.e. in the top $1/\sqrt{n}$ quantile).
2. Workers propose to firms as in DA, but to propose to a firm, they must satisfy one of the following requirements:
   - They are one of the agents signaled to be desirable in the previous step.
   - Their public term a_{ji} meets a qualification threshold q_j maintained by each firm j, and updated as j's tentative match is updated.

The restrictions on eligible proposals by workers, combined with the bounded hazard rate assumption, guarantees that whenever a firm switches tentative match to a different worker, they are probably increasing their utility by a large amount, which will eventually cause their utility to "max out" and conclude the process.

3.4 Tiered Model

The paper also considers a model where each side has multiple tiers, where each tier's members are preferable to all members of lower tiers. The protocol for this problem roughly has the following form:

- Each tier has a corresponding "target" tier on the other side of the market. Agents in each tier signal their preferred agents in the target tier.
- Run DA with the partial preference lists given by the signals.

4 Matching While Learning

Yash briefly covered this third paper, which deals the following rough model:

- Workers arrive online. Their defining characteristic or "type" is not known.
- Jobs arrive online. Their type is known to the designer.
- Each worker survives in the system for N time periods, and may work on a different job chosen by the algorithm in each period.

- Each type of worker has a different success probability on each type of job.

The algorithm's task is to maximize the number of completed jobs, which must be done by learning workers' types as the results of their matchings are realized. The authors give an algorithm which achieves near-optimal regret.

[Note: one often sees in matching systems that new users are often given preferential treatment so they don't leave the platform. This seems in tension with the need of the platform to experiment on users and learn about their type.]