Incentives and Machine Learning (CSC 482A/581A)
Lectures 11–13

Nishant Mehta

1 Recap and Beyond

Let's quickly review what we have covered in this course so far. In information elicitation, we saw proper scoring rules, which are basic examples of mechanisms that incentivize agents to truthfully reveal their information. Next, in auctions, we focused on DSIC mechanisms that maximize social welfare (and ideally satisfy some additional properties as well). Designing an auction mechanism that is DSIC can be viewed as incentivizing agents to be truthful while also considering a numbering of other objectives as well.

We are now moving into the third major topic for this course: how to learn equilibria. Up until now, we have focused on dominant strategy equilibria. For information elicitation, we only considered a single agent, in which case any equilibrium must be a dominant strategy equilibrium. For auctions, recall the "DS" in DSIC stands for dominant strategy. In both of these examples, the equilibria were truthful, and we designed games keeping in mind that we want to ensure that truthful equilibria exist. But what if we instead are *given* a game? How can we determine if it has various types of equilibria? Moreover, if a game has a certain type of equilibrium (like a mixed strategy Nash equilibrium), is it possible to approximately find such an equilibrium?

2 Congestion Games

Having spent most of the course so far with dominated strategy equilibria, we'll now switch focus to pure strategy Nash equilibria. *Congestion games* are an excellent class of games when it comes to learning about pure strategy Nash equilibria. Informally, in a congestion game, multiple agents use a common set of resources. Each resource induces a cost that depends on how many agents are using the resource, and finally, each agent suffers a cost that is the sum of the costs of the resources they are using. You might notice that we've switched from talking about agents having utilities (which should be maximized) to agents having costs (which should be minimized). This switch is because many typical examples of congestion games work better when agents have cost functions rather than utility functions. We will continue using cost functions when we shift our attention to online learning. Let's formally define congestion games.

Definition 1 (Congestion game)**.** In a *congestion game*,

- there are n agents (n is finite);
- there is a set R of m resources;
- each agent i has a set of strategies $A_i \subseteq 2^R$, so each strategy $a_i \in A_i$ is a subset of R (we may think of agent i as using each resource in a_i);
- each resource $r \in R$ is associated with a cost[1] function $\ell_r : \mathbb{N} \to \mathbb{R}$;

As usual, $A = A_1 \times \ldots A_n$. Each agent i has a cost function

$$c_i(a) = \sum_{r \in a_i} \ell_r(n_r(a)),$$

where $n_r(a) := \left|\{j \in [n] \colon r \in a_i\}\right|$ is the number of agents using resource r.

Let's consider a first example of a congestion game.

Example 1. This example is a variant of Pigou's example.[2] Let $n \geq 2$, and let the set of resources be equal to the set of edges in the graph below. Each agent wishes to go from the source vertex s to the sink vertex t. The edge costs are a function of how many agents use each edge.

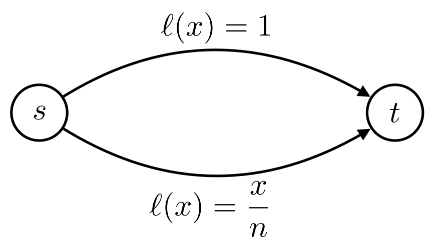

Figure 1: Variant of Pigou's example

Let's consider the example when $n = 10$. This game has many pure strategy Nash equilibria: in all the pure strategy Nash equilibria, at least 9 agents take the bottom route. Indeed, if only 8 agents take the bottom route, then an agent taking the top route would be better off by switching to the bottom route. If at least 9 agents take the bottom route, the agent taking the top route would not be better off by taking the bottom route. For this game, pure strategy Nash equilibria are undesirable in a certain sense. Why? First, let's define the *social cost* $C(a)$ of a strategy profile a as the sum of the agents' respective costs:

$$C(a) = \sum_{i=1}^{n} c_i(a).$$

[1] If it helps you remember the notation, you may think of ℓ as a loss function, or as a load function.

[2] In the original version of Pigou's example, we assume that there are infinitely many agents and encode any strategy profile as the proportion of agents using each edge. In this lecture, we look at a variation with finite n to ensure that our example is a congestion game.

Social cost is the analogue of social welfare, adapted to cost functions. Now, observe that any pure strategy Nash equilibrium has average (averaged over agents) social cost of at most

$$\frac{(n-1)\cdot\frac{n-1}{n}+1\cdot 1}{n}=1-\frac{1}{n}+\frac{1}{n^2},$$

which for $n = 10$ gives 0.91. Moreover, as we increase n, the average social cost approaches 1. If instead half the agents took the top route and the other half took the bottom route, then the average social cost would be $\frac{3}{4}$. This discrepancy motivates a new definition.

Definition 2 (Price of Anarchy[3]). The *price of anarchy* is

$$\frac{\max_{a\in \text{PNE}} C(a)}{\min_{a\in A} C(a)},$$

where PNE is the set of pure strategy Nash equilibria.

We want the price of anarchy (PoA) to be as low as possible; 1 is ideal.

What is the price of anarchy (PoA) in Example 1 as $n \to \infty$? Since there is only one pure strategy Nash equilibrium, the price of anarchy is $\frac{1}{3/4} = \frac{4}{3}$.

Let's look at another example, again keeping in mind what happens as $n \to \infty$. Consider the left panel of Figure 2 below.

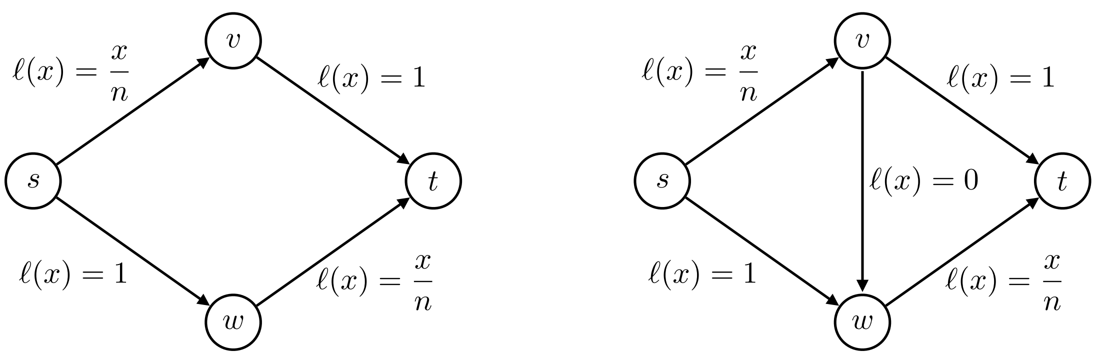

Figure 2: Original network and modified network

Does this game have any pure strategy Nash equilibria? Yes, and there are actually many of them, but they all share a common characteristic: as $n \to \infty$, in every pure strategy Nash equilibrium, half the agents take the top path and the other half take the bottom path. Note that these pure strategy Nash equilibria all have an average social cost of 1.5, which is the same as the optimal average social cost. Hence, the PoA is 1.

Braess's paradox is the curious phenomenon in selfish routing games that introducing an extra, lower cost edge can sometimes actually lead an equilibrium that is much worse. Consider modifying

[3]This definition is the right one when agents have cost functions. When agents have utility functions $u_1, \ldots, u_n$, price of anarchy is defined in terms of social welfare $W(a) = \sum_{i=1}^n u_i(a)$ as $\frac{\max_{a\in A} W(a)}{\min_{a\in \text{PNE}} W(a)}$. Lower is still better, and 1 is still ideal.

the above network by introducing a new, zero-cost edge as shown in the right panel of Figure 2 above.

In the new game, there is precisely one pure strategy Nash equilibrium: all agents take start by taking the top left edge, then taking the "free" edge down, and finally taking the bottom right edge. Indeed, if there exists a positive fraction of agents that do not take this "zig-zag" path, then any one of these agents would be better off by unilaterally deviating to take the zig-zag path. Consequently, for any pure strategy Nash equilibrium, the social cost is 2. Yet, any strategy profile that existed in the original game is still valid in this game (agents are free to ignore the new edge); in particular, the optimal average social cost is still 1.5. Thus, the price of anarchy is $\frac{4}{3}$.

A generalization. As $n \to \infty$, we saw that the PoA is 4/3 in both Pigou's example and example in Figure 2. Is this just a coincidence? Well, not exactly. For a general class of games called *selfish routing games*, when the cost function for any resource is linear, it turns out that the PoA can be at most $4/3 \approx 1.33$ (and this is tight, as witnessed by the examples we saw). If one allows cost functions to be quadratic, the worst case PoA is roughly 1.6. For cubic, it is roughly 1.9. In general, for a polynomial of degree p, we get roughly $\frac{p}{\log p}$, where log is the natural logarithm. For more details about what exactly are selfish routing games, as well as proofs of these claims, see Lecture 11 of Tim Roughgarden's Twenty Lectures on Algorithmic Game Theory.

3 Congestion Games and Best-Response Dynamics

We've now seen a few examples of congestion games. In each example, there was at least one pure strategy Nash equilibrium. Do all congestion games have pure strategy Nash equilibria? Remember that Nash's Theorem only tells us that every finite game has a *mixed strategy* Nash equilibrium (recall the game matching pennies, which had no pure strategy Nash equilibria). Could it be that some congestion games has pure strategy Nash equilibria while others don't? The answer is no.

Theorem 1. *Every congestion game has a pure strategy Nash equilibrium.*

We'll see a constructive proof that makes use of a method called *Best-Response Dynamics* which, in any congestion game, returns a pure strategy Nash equilibrium. Best-Response Dynamics is an iterative procedure for updating a strategy profile a that works as follows. Whenever there exists an agent i that could reduce its cost by best-responding to a_{-i}, the agent's current strategy a_i is updated to a best response. Recal that a best response to a_{-i} is some $a_i' \in A_i$ that minimizes $c_i(a_i', a_{-i})$. The formal procedure is shown below.

Algorithm 1: Best-Response Dynamics

Initialize $a = (a_1, \ldots, a_n)$ arbitrarily;
while *there exists an agent i such that* $\min_{a_i' \in A_i} c_i(a_i', a_{-i}) < c_i(a_i, a_{-i})$ **do**
  Select any a_i^* satisfying $c_i(a_i^*, a_{-i}) = \min_{a_i' \in A_i} c_i(a_i', a_{-i})$;
  $a_i \leftarrow a_i^*$;
end
return a;

The following lemma is immediate from the definition of pure strategy Nash equilibrium and the halting condition of Best-Response Dynamics.

Lemma 1. *If Best-Response Dynamics halts, then it returns a pure strategy Nash equilibrium (so, a pure strategy Nash equilibrium exists).*

Theorem 2. *In every congestion game, Best-Response Dynamics halts.*

This theorem, in combination with the previous lemma, implies the following corollary.

Corollary 1. *Every congestion game has a pure strategy Nash equilibrium.*

Let's prove the theorem.

Proof (of Theorem 2). The idea of the proof is to introduce a *potential function* $\Phi : A \to \mathbb{R}$ with the property that whenever some agent i reduces its cost by best-responding to a_{-i}, the potential function also gets reduced. Formally, we will show that for all agents $i \in [n]$, strategies $a_i, a_i' \in A_i$, and $a_{-i} \in A_{-i}$, it holds that

$$c_i(a_i', a_{-i}) - c_i(a_i, a_{-i}) = \Phi(a_i', a_{-i}) - \Phi(a_i, a_{-i}). \tag{1}$$

Hence, any iteration of Best-Response Dynamics also reduces the potential function, implying that the algorithm never revisits any strategy profile a. Since there are only finitely many strategy profiles, the algorithm must terminate.

It remains to establish (1). Observe that

$$\begin{aligned}
&\Phi(a_i', a_{-i}) - \Phi(a_i, a_{-i}) \\
&= \sum_{r \in R} \sum_{j=1}^{n_r(a_i', a_{-i})} \ell_r(j) - \sum_{r \in R} \sum_{j=1}^{n_r(a_i, a_{-i})} \ell_r(j) \\
&= \sum_{r \in a_i' \setminus a_i} \left(\sum_{j=1}^{n_r(a_i', a_{-i})} \ell_r(j) - \sum_{j=1}^{n_r(a_i, a_{-i})} \ell_r(j) \right) + \sum_{r \in a_i \setminus a_i'} \left(\sum_{j=1}^{n_r(a_i', a_{-i})} \ell_r(j) - \sum_{j=1}^{n_r(a_i, a_{-i})} \ell_r(j) \right) \\
&= \sum_{r \in a_i' \setminus a_i} \ell_r(n_r(a_i', a_{-i})) - \sum_{r \in a_i \setminus a_i'} \ell_r(n_r(a_i, a_{-i})),
\end{aligned}$$

which is the same as $c_i(a_i', a_{-i}) - c_i(a_i, a_{-i})$. □

Let's reflect on what the proof did. The proof showed that the existence of an *exact potential function*, i.e., a potential function for which equality (1) holds, implies that Best-Response Dynamics halts. If you reflect on the proof, you might notice that it didn't really matter that whenever an agent's cost decreases, the potential decreases by exactly the same amount. All that we really needed was that if an agent's cost decreases, the potential function also decreases. With this insight in mind, we can define a large class of games for which the core argument of the above proof still goes through.

4 Potential Games

Definition 3 (Potential game)**.** We say a game is an *(ordinal) potential game* if there exists a potential function $\Phi : A \to \mathbb{R}$ such that for all agents $i \in [n]$, all strategies $a_i, a_i' \in A_i$, and all $a_{-i} \in A_{-i}$, we have $c_i(a_i, a_{-i}) - c_i(a_i', a_{-i}) < 0$ implies $\Phi(a_i, a_{-i}) - \Phi(a_i', a_{-i}) < 0$.

We will simply write "potential game" to mean ordinal potential game.

It is not hard to see that for every finite potential game, Best-Response Dynamics converges. Thus, we have the following proposition.

Proposition 1. *Let G be a finite game. If G is a potential game, then Best-Response Dynamics is guaranteed to converge.*

If we zoom out quite a bit, we can phrase this result as "for every game that has structural property X, a good thing Y happens". We saw a previous such result in the case of proper scoring rules: for every scoring rule that has the structural property that it can be expressed in terms of some convex function G, the scoring rule is proper (so truthful reporting is optimal). For proper scoring rules, we also saw that every proper scoring rule *must* have this structural property. With this perspective in mind, it is natural to ask:

> For every finite game in which Best-Response Dynamics is guaranteed to converge ("a good thing Y happens"), is the game a potential game ("does the game have structural property X")?

It turns out the answer is yes, so that Proposition 1 can be extended to an if and only if.

Theorem 3. *Let G be a finite game. Best-Response Dynamics is guaranteed to converge if and only a G is a potential game.*

Proof.

($\Leftarrow$) Suppose a game is a potential game. It is easy to see that our proof that Best-Response Dynamics converges in congestion games still goes through for potential games.

($\Rightarrow$) Next, suppose Best-Response Dynamics is guaranteed to converge in G. We form a directed graph from the game G as follows. Create a vertex for each strategy profile. From any strategy profile $a \in A$, we create an edge to each strategy profile a' that results from some agent best-responding and lowering their cost; more formally, for every pair of strategy profiles a and a' such that, for some agent i, we have $a' = (a'_i, a_{-i})$ and $c_i(a') < c_i(a)$, create an edge from a to a'.

Since Best-Response Dynamics is guaranteed to converge, the graph does not have any cycles. Hence, there must exist at least one vertex (a sink) with no outgoing edges, and any such vertex is a pure strategy Nash equilibrium. We define the potential function as follows: for any $a \in A$, set $\Phi(a)$ to be the longest path distance[4] to a pure strategy Nash equilibrium (PNE). Finally, observe that for any edge (a, a'), the longest path distance from a to a PNE is at least as long as the longest path distance from a' to a PNE (since, starting at a, we can first go to a' and then follow any longest path from a' to a PNE). Hence, we have $\Phi(a) \geq \Phi(a') + 1$, which gives $\Phi(a') < \Phi(a)$, as desired. □

We have now seen a characterization of those finite games for which Best-Response Dynamics converges. For any potential game, we can be assured that in finite time, Best-Response Dynamics will give us a pure strategy Nash equliibrium. What more could we ask for? We can ask for quite a bit more! In general, Best-Response Dynamics might take a number of iterations to converge that is exponential in the number of agents. That's bad. The good news is that if one is OK with an *approximate* pure strategy Nash equilibrium, then the core algorithmic idea (of "best-response moves") with a small tweak can give a reasonable approximation in a reasonable amount of time.

Let's first see a notion of an approximate pure strategy Nash equilibrium.

Definition 4. We say that $a \in A$ is an ε-approximate pure strategy Nash equilibrium if, for all $i \in [n]$ and $a'_i \in A_i$, it holds that

$$c_i(a'_i, a_{-i}) > c_i(a_i, a_{-i}) - \varepsilon.$$

[4] Each edge has a cost of 1.

The above definition is for costs. It is straightforward to adapt the definition to the case of payoffs (utilities). One could instead consider multiplicative approximate (relative approximation)

Now, let's consider a simple modification of Best-Response Dynamics for learning an ε-approximate pure strategy Nash equilibrium.

Algorithm 2: ε-Best-Response Dynamics

Initialize $a = (a_1, \ldots, a_n)$ arbitrarily;
while *there exists an agent i such that* $\min_{a'_i \in A_i} c_i(a'_i, a_{-i}) \leq c_i(a_i, a_{-i}) - \varepsilon$ **do**
  Select any a_i^* satisfying $c_i(a_i^*, a_{-i}) = \min_{a'_i \in A_i} c_i(a'_i, a_{-i})$;
  $a_i \leftarrow a_i^*$;
end
return a;

You may have noticed that in the original Best-Response Dynamics algorithm, it was not important for our analysis that the deviating agent i chooses a best response. It would have sufficed for the agent to choose *any* response that strictly reduces her cost. Something similar is true for ε-Best-Response Dynamics: it suffices for a deviating agent i to select any a'_i that reduces her cost by at least ε.

The next result is more or less immediate from the definition of *exact potential games*, i.e., games for which an exact potential function exists.

Theorem 4. *Let G be an exact potential game with a potential function Φ taking values in a finite range $[\phi_{\min}, \phi_{\max}]$. Then ε-Best-Response Dynamics is guaranteed to converge after at most $\frac{\phi_{\max} - \phi_{\min}}{\varepsilon}$ iterations.*

For congestion games in particular (which we saw are exact potential games), if we assume that for all $r \in R$ and $j \in \{0, 1, \ldots, n\}$, it holds that $\ell_r(j) \in [0, L]$, then $\phi_{\min} = 0$ and $\phi_{\max} = m \cdot n \cdot L$. Hence, for such congestion games, ε-Best-Response Dynamics converges after at most $\frac{m \cdot n \cdot L}{\varepsilon}$ iterations.

Bibliographical notes

1. These notes draw heavily from Bo Waggoner's Lectures 2 and 3 of his 2018 Algorithmic Game Theory course at UPenn, which can be found here: Lecture 2 Lecture 3.

2. For a different notion of ε-approximate Nash equilibrium as well as more advanced results for a slightly different notion of ε-Best Response Dynamics , see Lecture 16 of Tim Roughgarden's Twenty Lectures on Algorithmic Game Theory: try this link or this one.