3.1 Introduction

Today we'll be talking about algorithms for computing Nash equilibria in general two-player games (not necessarily zero-sum). We know that a Nash equilibrium always exists, but how quickly can we find it? We'll see two related exponential-time algorithms today, which is somewhat slow but is basically the best we can really hope for. There are also some really nice structural consequences.

To start, recall that in a two-player game there are two players (the row player and the column player) with $S_1 = [N]$ and $S_2 = [M]$, and two matrices $A, B \in \mathbb{R}^{N \times M}$. Then $u_1(i,j) = A_{ij}$ and $u_2(i,j) = B_{ij}$, and hence if player 1 uses mixed strategy $x \in \Delta_N$ and player 2 uses mixed strategy $y \in \Delta_M$ then the expected of utility player 1 is $x^T Ay$ and the expected utility of player 2 is $x^T By$.

3.2 Best Response and Support Enumeration

3.2.1 Best Response

The first algorithm is based off of a simple but not necessarily obvious observation: even though Nash equilibria involve distributions and thus seem like continuous objects, they are "really" combinatorial. Se show this, we're going to have to give some easy definitions to formalize some obvious concepts, namely the idea of a "best response".

Definition 3.2.1 *$x \in \Delta_N$ is a* best response *to $y \in \Delta_M$ if $x^T Ay = \max_{x' \in \Delta_N} x'^T Ay$. Similarly, $y \in \Delta_M$ is a best response to $x \in \Delta_N$ if $x^T By = \max_{y' \in \Delta_M} x^T By'$*

So (x, y) is a Nash equilibrium if and only if x and y are best responses to each other.

We can use the same logic from last class (where we reasoned about what one player will do if the other player commits to some mixed strategy) to prove the following straightforward lemma.

Lemma 3.2.2 *Let $x \in \Delta_N$ and $y \in \Delta_M$. Then x is a best response to y if and only if for all $i \in [N]$,*

$$x_i > 0 \implies (Ay)_i = \max_{k \in [N]} (Ay)_k.$$

Similarly, y is a best response to x if and only if for all $j \in [M]$,

$$y_j > 0 \implies (x^T B)_j = \max_{k \in [M]} (x^T B)_k.$$

In other words, x is a best response to y if and only if x has nonzero probability only on pure strategies that max expected utility given y.

Proof: Suppose that x is a best response to y. Then $x^T Ay = \max_{x' \in \Delta_N} x'^T Ay = \max_{k \in [N]} (Ay)_k$ (by convexity). Thus x is a convex combination of entries of Ay which equals the maximum value, so it can only have support on the maximum values. More formally, if $x_i > 0$ but $(Ay)_i < \max_{k \in [N]} (Ay)_k$, then

$$\begin{aligned}
x^T Ay &= \sum_{k=1}^{N} x_k (Ay)_k = x_i (Ay)_i + \sum_{k \neq i} x_k (Ay)_k \\
&< x_i \max_{k \in [N]} (Ay)_k + \sum_{j \neq i} x_j \max_{k \in [N]} (Ay)_k \\
&= x_i \max_{k \in [N]} (Ay)_k + (1 - x_i) \max_{k \in [N]} (Ay)_k \\
&= \max_{k \in [N]} (Ay)_k,
\end{aligned}$$

which is a contradiction to x being a best response to y.

For the other direction, suppose that $x_i > 0 \implies (Ay)_i = \max_{k \in [N]} (Ay)_k$ for all $i \in [N]$. Then

$$x^T Ay = \sum_{i=1}^{N} x_i (Ay)_i = \sum_{i: x_i \neq 0} x_i (Ay)_i = \sum_{i: x_i \neq 0} x_i \max_{k \in [N]} (Ay)_k = \max_{k \in [N]} (Ay)_k \sum_{i: x_i \neq 0} x_i = \max_{k \in [N]} (Ay)_k.$$

An analogous argument works for the second part of the lemma. ■

3.2.2 Support Enumeration

So suppose that we have a guess for the support $I \subseteq [N]$ and $J \subseteq [M]$ of a Nash equilibrium. How can we check whether our guess is correct, i.e., there is some Nash with that support? We can just set up a system of linear equations!

$$\begin{aligned}
\sum_{i \in I} x_i &= 1 \\
\sum_{j \in J} y_j &= 1 \\
\sum_{i \in I} x_i B_{ij} &= v && \forall j \in J \\
\sum_{j \in J} A_{ij} y_j &= u && \forall i \in I
\end{aligned}$$

This system of linear equations has $|I| + |J| + 2$ equations and $|I| + |J| + 2$ variables, so there is a unique solution unless it is degenerate. So we can find the unique solution (using, e.g., Gaussian elimination). Once we have a solution, we can check whether it corresponds to a Nash: we check whether all variables a nonnegative and whether $u = \max_{k \in [N]} (Ay)_k$ and whether $v = \max_{k \in [M]} (x^T B)_k$. If all of these checks pass, then we've found a Nash (by Lemma 3.2.2). And if any of them fail, then I, J cannot be the support of a Nash.

Thus we can find a Nash equilibrium (in fact, all Nash equilibria) by "support enumeration": we just try all possible $I \subseteq [N]$ and $J \subseteq [M]$ and check (using the above procedure) whether they give a Nash.

3.3 Lemke-Howson

For the rest of today we're going to focus on a more involved algorithm known as Lemke-Howson. Making this fully formal requires a bit more convex geometry than is required for this course, so I'm going to be a little bit handwavy. But it's really nice despite the fact that it's exponential time because it combines game theory, convex geometry, and graph theory – it's a whole bunch of cool math in one place.

Let's first set up some notation and basic definitions. For a vector $x \in \mathbb{R}^k$, let $S(x) = \{i : x_i \neq 0\}$ be its support.

Definition 3.3.1 *A bimatrix game is* non-degenerate *if and only if:*

1. *For all $x \in \Delta_N$, $|S(x)| \geq \#$ pure best responses to x, and*
2. *for all $y \in \Delta_M$, $|S(y)| \geq \#$ pure best responses to y.*

This is a bit hard to interpret, so let's try to give a little bit of intuition for this condition. It says that for any $x \in \Delta_N$, the number of coordinate of $x^T B$ with the maximum value is at most $|S(x)|$, and similarly for Ay and $|S(y)|$. So, for example, if x is a pure strategy then $|S(x)| = 1$, and thus non-degeneracy requires that there is at most 1 pure best response to x. Since this is true for all pure strategies x, this means that every row of B has a unique max. Similarly, every column of A has a unique maximum.

It's also worth noting that slightly perturbing A, B is enough to guarantee non-degeneracy, so it's not too much of a restriction.

Lemke-Howson is based off of thinking about some geometric/algebraic objects, which we're now going to define. We first define two convex polyhedra:

$$\bar{P} = \left\{(x, u) : x \in \Delta_N, \sum_{i=1}^{N} x_i B_{ij} \leq u \ \ \forall j \in [M]\right\}$$
$$\bar{Q} = \left\{(y, v) : y \in \Delta_M, \sum_{j=1}^{M} y_j A_{ij} \leq v \ \ \forall i \in [N]\right\}$$

These polyhedra basically correspond to mixed strategies together with an upper bound on the best response of the *other* player. These are polyhedra (which are not bounded), and we would instead like to make them bounded (i.e., polytopes). To do this, we'll "normalize" the last coordinate to 1 (and thus get rid of it) by getting rid of the upper bound by "dividing through" the other coordinates to get the following polytopes.

$$P = \left\{x : x_i \geq 0 \ \ \forall i \in [N], \ \sum_{i=1}^{N} x_i B_{ij} \leq 1 \ \ \forall j \in [M]\right\}$$
$$Q = \left\{y : y_j \geq 0 \ \ \forall j \in [M], \ \sum_{j=1}^{M} y_j A_{ij} \leq 1 \ \ \forall i \in [N]\right\}$$

Recall from optimization / convex geometry that a *vertex* of a polyhedron is an extreme point (there is no direction z such that we can move in both z and $-z$ and stay in the polytope). Because the game is non-degenerate all of these polyhedra are also non-degenerate, so every vertex of any of the polyhedra corresponds to a point where the number of tight constraints is exactly equal to the number of variables.

It is not hard to see that from the perspective of their vertices, our polyhedra and polytopes are basically the same.

Lemma 3.3.2 *There is a bijection between the vertices of P and $\bar{P}$, except for $0 \in P$. There is also a bijection between the vertices of Q and $\bar{Q}$, except for $0 \in Q$.*

Proof: Suppose that (x, u) is a vertex of $\bar{P}$. Then (x/u) is a vertex of P. Similarly, suppose that $x \neq 0$ is a vertex of P. Then $\left(\frac{x}{\sum_{i=1}^N x_i}, \frac{1}{\sum_{i=1}^N x_i}\right)$ is a vertex of $\bar{P}$.

A similar argument works for Q and $\bar{Q}$. ∎

So I'm going to back and forth between vertices of P and $\bar{P}$ and Q and $\bar{Q}$ pretty interchangeably, just to simplify notation.

We're going to show a relationship between vertices of these objects and Nash equilibria. To get some intuition before we formalize things, suppose that x is a vertex of P. Then there are N tight constraints at x. Some of them (say k) are nonnegativity constraints: i such that $x_i = 0$. This means that there are $N - k$ values of $j \in [M]$ so that $\sum_{i=1}^N x_i B_{ij} = 1$, which means that each such j is a best response to x (or really to the scaled up x which is a vertex in $\bar{P}$)!

Definition 3.3.3 *The* label set *is $L = [N] \bigsqcup [M]$ (where $\sqcup$ is the disjoint union).*

We're going to label vertices of the polytopes using L: the labels of a vertex will correspond to the tight constraints at that vertex. So for any vertex x of P we let

$$L(x) = \{i : x_i = 0\} \bigcup \left\{j : \sum_{i=1}^N x_i B_{ij} = 1\right\}$$

be the N tight constraints at x, and for any vertex y of Q we let

$$L(y) = \{j : y_j = 0\} \bigcup \left\{i : \sum_{j=1}^M y_j A_{ij} = 1\right\}$$

be the M tight constraints at y.

Let's relate all of this convex geometry back to the game setting.

Theorem 3.3.4 *(x, y) is a Nash equilibrium if and only if $L(x) \cup L(y) = L$ and $x, y \neq 0$.*

Proof: For the if direction, suppose that $L(x) \cup L(y) = L$. Then since $|L(x)| = N$ and $|L(y)| = M$, every label appears *exactly* once, either in $L(x)$ or $L(y)$. This allows us to partition $[N]$ into two sets:

$$N_1 = \{i \in [N] : x_i = 0\} \qquad (i \in L(x))$$
$$N_2 = \left\{i \in [N] : \sum_{j=1}^M y_j A_{ij} = 1\right\} \qquad (i \in L(y))$$

(note that this is a partition since every label in $[N]$ appears in either $L(x)$ or $L(y)$ but not both). Thus $S(x) = N_2$, which means that x is a best response to y (both appropriately scaled up to be in $\bar{P}$ and $\bar{Q}$) since x has support only in entries that give the maximum value in Ay.

Similarly, we can partition $[M]$ into

$$M_1 = \{j \in [M] : y_j = 0\} \qquad (j \in L(y))$$
$$M_2 = \left\{ j \in [M] : \sum_{i=1}^{N} x_i B_{ij} = 1 \right\} \qquad (j \in L(x))$$

Then $S(y) = M_2$, and so y is a best response to x. Thus (x, y) is a Nash equilibrium.

For the only if direction, suppose that (x, y) is a Nash equilibrium. If we can show that every label appears in $L(x) \cup L(y)$ *at least* once then this will imply that $L(x) \cup L(y) = L$. So consider some label $i \in [N]$. If $x_i = 0$ then $i \in L(x)$. Otherwise, $x_i > 0$ which by Lemma 3.2.2 implies that $(Ay)_i = 1$ and thus $i \in L(y)$. Similarly, consider some $j \in [M]$. If $y_j = 0$ then $j \in L(y)$, and otherwise by Lemma 3.2.2 we get that $j \in L(x)$. Thus $L \subseteq L(x) \cup L(y)$ and so $L = L(x) \cup L(y)$. ■

This is great – now in order to find a Nash, we just have to find vertices of two polytopes that together contain all of the labels. To do this, we're going to actually build some graphs that are related to these two polytopes. These graphs will be based on the graphs of the polytopes themselves. Consider the vertices of P (points where N constraints are tight). We can say that two vertices are *adjacent* in P is there is an "edge" between them: there is an $(N-1)$-dimensional face that includes both of them. More combinatorially, let x, x' be two vertices of P. Then there is an edge between them if $|L(x) \cap L(x')| = N - 1$. Note that this means we can think of "dropping" a label from x and then "adding" a label to get from x to x', since the labels are exactly the tight constraints. Similarly, for two vertices $y, y' \in Q$ we say that there is an edge between them if $|L(y) \cap L(y')| = M - 1$.

Now we have two graphs: one for P and one for Q. We're going to combine these into one graph. Let V_1 be the vertices of P, and let V_2 be the vertices of Q. Our new graph G will have vertex set $V = V_1 \times V_2$. We'll add an edge between (x, y) and (x', y') is either

- x is adjacent to x' in P and $y = y'$, or
- y is adjacent to y' in Q and $x = x'$.

Instead of reasoning about G itself, we're going to reason about some subgraphs of G. Let $k \in L$ be an arbitrary label (in either $[N]$ or $[M]$). Let

$$U_k = \{(x, y) \in V : L(x) \cup L(y) \supseteq L \setminus \{k\}\}$$

be vertices of the graph that contain all labels except possibly k (these are known as k-*almost completely labeled*). Let H_k be the subgraph of G induced by U_k. The following theorem gives the key structure about H_k.

Theorem 3.3.5 *For all $k \in L$:*

1. $(0, 0)$ *and all Nash equilibria are in* U_k*, and their degree in* H_k *is* 1*, and*

2. *Every other vertex in* U_k *has degree* 2 *in* H_k*.*

Proof: We know from Theorem 3.3.4 that every Nash equilibrium (x, y) has all labels, and so is clearly in U_k. And obviously $(0, 0)$ has all labels and so is in U_k.

Let (x, y) be either a Nash or $(0, 0)$, and without loss of generality let's assume that $k \in L(x)$. Then by making the constraint for k *not* tight we are on a one-dimensional face of P (i.e., an edge of P), so we can move to the other end of this edge to be at some vertex x' of P. In H_k, this corresponds to an edge between (x, y) and (x', y), since the only label we lost by going from x to x' is k. Thus (x, y) has degree at least 1 in H_k.

Now suppose that there is some other edge incident on (x, y) in H_k. Then this corresponds to either moving from x to some $x'' \neq x'$ in P, or moving from y to some y' in Q. But this means moving along an edge of either P or Q which corresponds to dropping a label *other* than k, and thus would involve removing a label other than k. But that means that the other endpoint of this edge would not be in U_k. Thus (x, y) has degree exactly one in H_k.

For part two of the theorem, let (x, y) be any other vertex in U_k. Since $|L(x)| + |L(y)| = N + M$ but $k \notin L(x) \cup L(y)$ (or else (x, y) would be a Nash), there must be some label ℓ which is in both $L(x)$ and $L(y)$. This means that we can "drop" ℓ from x (make the constraint corresponding to ℓ no longer tight in P and move along the corresponding edge of P) and move to some (x', y) which will still be in U_k, or can drop ℓ from y and move to some $(x, y') \in U_k$. Thus (x, y) has degree at least two. And it has degree at most two because we cannot drop any other label from it (there is only one duplicated label since every label other than k must be in $L(x) \cup L(y)$. ∎

This theorem implies that H_k is a collection of paths and cycles, since that's the only way to make all vertices have degree one or two. Now the Lemke-Howson algorithm is very simple:

1. Choose $k \in L$ arbitrarily.

2. Start at $(0, 0)$ (which has degree 1 in H_k) and walk along the path in H_k until we reach a node of degree 1.

3. Return it.

Thanks to Theorem 3.3.5, it is obvious that this algorithm does in fact return a Nash. And note that it also gives the following interesting corollary:

Corollary 3.3.6 *In any non-degenerate bimatrix game, there are an odd number of Nash equilibria.*