Computational Game Theory (CIS 620/OPIM 452)
A Sampling Argument for Approximate NE
M. Kearns Jan 21, 2002

Setting: 2-player, m-action games

Approach:
• First show that there always exists an approx. NE
  \((P, Q)\) s.t. \(P \neq Q\) have small support
• Then search small-support \(P \neq Q\)
  exhaustively
• Result will be a pseudopolynomial algorithm for computing approx. NE

Let \(P\) be a distribution, \(\text{support}(P) = \{i : P_i \neq 0\}\)
Note that if \((P, Q)\) is a NE, then
• \(i \in \text{support}(P) \implies i\) a best response to \(Q\)
• \(j \in \text{support}(Q) \implies j\) is a b.r. to \(P\)

So let \((P, Q)\) be a NE in which the expected payoffs are \((U, V)\) \((U, V, R)\).
Suppose we sample \(P \neq Q\) \(t\) times each, and create empirical (sampled) distributions \((P', Q')\).
Some notation:

- for any $i$, let
  \[ M_i(i, Q) = E_{j \in Q}[M_i(i, j)] \]

- for any $j$, let
  \[ M_2(P_j) = E_{i \in P}[M_i(i, j)] \]

- $M_i(i, Q)$

Now fix $i$, if sample size $t$ is large enough, $M_i(i, Q^t) = M_i(i, Q)$.

Why? The return that $i$ gets in a single round against $Q$ is a random variable with mean $M_i(i, Q)$. $M_i(i, Q^t)$ is the average of this r.v. over $t$ trials. So certainly

\[ \lim_{t \to \infty} M_i(i, Q^t) = M_i(i, Q) \]

But we need something stronger.

Claim: \( \exists \epsilon > 0: \)

\[ \Pr \left[ \left| M_i(i, Q^t) - M_i(i, Q) \right| > \epsilon \right] \leq e^{-\epsilon^2 t/3} \]

This prob. over the $t$ draws of $Q$!

(assumes $|M_i(i, j)| \leq 1$, but generalizes)

Large Deviation Bound
This was for a single, fixed $i$. But by $Pr[A \text{ or } B] \leq Pr[A] + Pr[B]$ (Union Bound) we have:

$$Pr \left[ |M_i(i, q') - M_i(i, q)| \geq \varepsilon \text{ for some } i \right] \leq m e^{-2t/3}$$

If again, over sample of $Q'$, and also

$$Pr \left[ |M_j(p', j) - M_j(p, j)| \geq \varepsilon \text{ for some } j \right] \leq m e^{-2t/3}$$

draw of $P'$

Note:

$\forall i \: |M_i(i, q') - M_i(i, q)| \leq \varepsilon \Rightarrow \forall p'' \: |M_i(p'' q') - M_i(p, q)| \leq \varepsilon$

$\forall j \: |M_j(p', j) - M_j(p, j)| \leq \varepsilon \Rightarrow \forall q'' \: |M_j(p' q'') - M_j(p, q'')| \leq \varepsilon$

So against $P'$, can't get better than $V + \varepsilon$

against $Q'$, can't get better than $U + \varepsilon$

Furthermore, since any $i$, $\text{support}(P)$ gets at least $U - \varepsilon$ against $Q'$, $P'$ gets at least $U - \varepsilon$ against $Q'$ (and $Q'$ gets at least $V - \varepsilon$ against $P'$).

Thus: $(P', Q')$ is a $2\varepsilon$-NE (and has support of size "only" $t$).
Now, to make sure such \((p', q')\) really exist, just need \(2m e^{-\varepsilon^2 t/3} \leq 1\), e.g.:

\[
2m e^{-\varepsilon^2 t/3} < \frac{1}{2}
\]

\[
e^{-\varepsilon^2 t/3} < \frac{1}{4m}
\]

\[-\varepsilon^2 t/3 < \log(4m)\]

\[\varepsilon^2 t/3 > \log(4m)\]

\[t > \frac{3}{\varepsilon^2} \log(4m) < \ll m\]

So if \(t\) is at least this large, we have

\(a 50\%\) chance of sampling \((p', q')\) that are \(2\varepsilon\)-NE \(\Rightarrow\) must exist such a \((p', q')\)!

The *Probabilistic Method*
OK... but how do we find \((P, Q)\)? Remember, we don't know \((P, Q)\) to begin!

Ugly but effective: discretized "exhaustive search":

We will search over all possible support sets of size \(t\). This gives

\[
\binom{m}{t} \times \binom{m}{t} = m^{2t}
\]

choices.

For each such support set, we will discretize the possible distributions—only considering probabilities that are multiples of some small value \(\Delta\). For any fixed support sets, the # of such distributions is

\[
\left(\frac{1}{\Delta}\right) \times \left(\frac{1}{\Delta}\right) \times \ldots \times \left(\frac{1}{\Delta}\right) = \left(\frac{1}{\Delta}\right)^t
\]

\(t\) times

or \(\left(\frac{1}{\Delta}\right)^{2t}\) for the pair of distributions.
Now claim that if a t-support set has a $2\varepsilon$-NE, then we will find a

$(2\varepsilon + 2t \Delta^t)$-NE among the $\Delta$-discretized distributions.

So if we want to end up with an $\varepsilon'$-NE, simply $2\varepsilon + 2t \Delta \leq \varepsilon'$

E.g. $\varepsilon = 3/4$, $\Delta = \frac{\varepsilon'}{4t}$ works.

Total computation time goes like:

$$m^{2t} \cdot \left(\frac{1}{\Delta^t}\right)^{2t} = \left(\frac{m}{\Delta}\right)^{2t} = \left(\frac{4tm}{\varepsilon'}\right)^{2t} = O\left(\frac{1}{(\varepsilon')^3 m \log(m)} \frac{4t^2 \log(4m)}{c \log(m)} \right).$$

Upshot; for fixed $\varepsilon'$, can compute $\varepsilon'$-NE in time $O\left(\frac{m \log(m)}{c \log(m)}\right)$ sub-exponential in m...

(But there must be a better idea...)