CPS 590.01
LP and IP in Game theory
(Normal-form Games, Nash Equilibria and Stackelberg Games)

Joshua Letchford
Zero-sum game (Mini-max)

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L</strong></td>
<td>1, -1</td>
<td>-1, 1</td>
</tr>
<tr>
<td><strong>R</strong></td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
</tbody>
</table>

- Assume opponent knows our mixed strategy
- If we play L 50%, R 50%...
- ... opponent will be indifferent between R and L...
- ... we get \(0.5 \times (-1) + 0.5 \times (1) = 0\)
General-sum games

- You could still play a minimax strategy in general-sum games
  - i.e., pretend that the opponent is only trying to hurt you

- But this is not rational:

<table>
<thead>
<tr>
<th></th>
<th>0, 0</th>
<th>3, 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 0</td>
<td>2, 1</td>
<td></td>
</tr>
</tbody>
</table>

- If Column was trying to hurt Row, Column would play Left, so Row should play Down
- In reality, Column will play Right (strictly dominant), so Row should play Up
- Is there a better generalization of minimax strategies in zero-sum games to general-sum games?
Nash equilibrium

[Nash 50]

• A vector of strategies (one for each player) is called a strategy profile.
• A strategy profile \((\sigma_1, \sigma_2, \ldots, \sigma_n)\) is a Nash equilibrium if each \(\sigma_i\) is a best response to \(\sigma_{-i}\).
  – That is, for any \(i\), for any \(\sigma_i'\), \(u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma_i', \sigma_{-i})\).
• Note that this does not say anything about multiple agents changing their strategies at the same time.
• In any (finite) game, at least one Nash equilibrium (possibly using mixed strategies) exists [Nash 50].
• (Note - singular: equilibrium, plural: equilibria)
The presentation game

<table>
<thead>
<tr>
<th>Audience</th>
<th>Presenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pay attention (A)</td>
<td>Put effort into presentation (E)</td>
</tr>
<tr>
<td></td>
<td>4, 4</td>
</tr>
<tr>
<td>Do not pay</td>
<td>Do not put effort into presentation (NE)</td>
</tr>
<tr>
<td>attention (NA)</td>
<td>-16, -14</td>
</tr>
<tr>
<td></td>
<td>0, -2</td>
</tr>
<tr>
<td></td>
<td>0, 0</td>
</tr>
</tbody>
</table>

- Pure-strategy Nash equilibria: (A, E), (NA, NE)
- Mixed-strategy Nash equilibrium:
  
  
  \[ \frac{1}{10} A, \frac{9}{10} NA, \frac{4}{5} E, \frac{1}{5} NE \]

- Utility 0 for audience, -14/10 for presenter
- Can see that some equilibria are strictly better for both players than other equilibria
Some properties of Nash equilibria

- If you can eliminate a strategy using strict dominance or even iterated strict dominance, it will not occur (i.e., it will be played with probability 0) in every Nash equilibrium
  - Weakly dominated strategies may still be played in some Nash equilibrium

- In 2-player zero-sum games, a profile is a Nash equilibrium if and only if both players play minimax strategies
  - Hence, in such games, if \((\sigma_1, \sigma_2)\) and \((\sigma_1', \sigma_2')\) are Nash equilibria, then so are \((\sigma_1, \sigma_2')\) and \((\sigma_1', \sigma_2)\)
    - No equilibrium selection problem here!
Solving for a Nash equilibrium using MIP (2 players)

[Sandholm, Gilpin, Conitzer AAAI05]

- maximize *whatever you like* (e.g., social welfare)
- subject to
  - for both $i$, $\sum_{s_i} p_{s_i} = 1$
  - for both $i$, for all $s_i$, $\sum_{s\_i} p_{s\_i} u_i(s_i, s\_i) = u_{s_i}$
  - for both $i$, for all $s_i$, $u_i \geq u_{s_i}$ \quad (u_i = \text{max } u_{s_i})
  - for both $i$, for all $s_i$, $p_{s_i} \leq b_{s_i}$
  - for both $i$, for all $s_i$, $u_i - u_{s_i} \leq M(1 - b_{s_i})$

- $b_{s_i}$ is a binary variable indicating whether $s_i$ is in the support, $M$ is a large number
Stackelberg (commitment) games

```
<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>1, -1</td>
<td>3, 1</td>
</tr>
<tr>
<td>R</td>
<td>2, 1</td>
<td>4, -1</td>
</tr>
</tbody>
</table>
```

- Unique Nash equilibrium is (R,L)
  - This has a payoff of (2,1)
Commitment

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>(1,-1)</td>
<td>(3,1)</td>
</tr>
<tr>
<td>R</td>
<td>(2,1)</td>
<td>(4,-1)</td>
</tr>
</tbody>
</table>

- What if the officer has the option to (credibly) announce where he will be patrolling?
- This would give him the power to “commit” to being at one of the buildings
  - This would be a pure-strategy Stackelberg game
Commitment...

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>(1,-1)</td>
<td>(3,1)</td>
</tr>
</tbody>
</table>

- If the officer can commit to always being at the left building, then the vandal's best response is to go to the right building.
  - This leads to an outcome of (3,1)
Committing to mixed strategies

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>(1,-1)</td>
<td>(3,1)</td>
</tr>
<tr>
<td>R</td>
<td>(2,1)</td>
<td>(4,-1)</td>
</tr>
</tbody>
</table>

• What if we give the officer even more power: the ability to commit to a mixed strategy
  – This results in a mixed-strategy Stackelberg game
  – E.g., the officer commits to flip a weighted coin which decides where he patrols
Committing to mixed strategies is more powerful

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>(1,-1)</td>
<td>(3,1)</td>
</tr>
<tr>
<td>R</td>
<td>(2,1)</td>
<td>(4,-1)</td>
</tr>
</tbody>
</table>

- Suppose the officer commits to the following strategy: \{(.5+\epsilon)\text{L},(.5-\epsilon)\text{R}\}
  - The vandal’s best response is R
  - As $\epsilon$ goes to 0, this converges to a payoff of (3.5,0)
Stackelberg games in general

- One of the agents (the leader) has some advantage that allows her to commit to a strategy (pure or mixed)
- The other agent (the follower) then chooses his best response to this
Visualization

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>C</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>0,1</td>
<td>1,0</td>
<td>0,0</td>
</tr>
<tr>
<td>M</td>
<td>4,0</td>
<td>0,1</td>
<td>0,0</td>
</tr>
<tr>
<td>D</td>
<td>0,0</td>
<td>1,0</td>
<td>1,1</td>
</tr>
</tbody>
</table>

$(0,1,0) = M$

$(1,0,0) = U$

$(0,0,1) = D$
Easy polynomial-time algorithm for two players

[Conitzer & Sandholm EC’06, von Stengel & Zamir GEB’10]

• For every column $j$, we solve separately for the best mixed row strategy (defined by $z_i$) that induces player 2 to play $j$

\[
\forall j \max \sum_i z_i R_{ij}
\]

\[
\forall j', \sum_i z_i C_{ij} \geq \sum_i z_i C_{ij'}
\]

\[
\sum_i z_i = 1
\]

• (May be infeasible for some $j$)

• Pick the $j$ that is best for player 1

$I$ rows

$J$ columns

$R$ is the defenders payoff matrix

$C$ is the attackers payoff matrix

$z_i$ is the probability that row $i$ is played
Extensions

- A few extensions with LP or MIP formulations:
  - Bayesian setting (DOBSS [1])
    - Uses a MIP to avoid exponential size
  - Multiple Defense Resources (ERASER [2])
    - Assumes the structure is a “Security game”
    - Uses this structure to achieve a compact representation
  - Defense Costs [3]
    - Explicit costs for defense rather than limited defense resources

(a particular kind of) **Bayesian games**

<table>
<thead>
<tr>
<th>leader utilities</th>
<th>follower utilities (type 1)</th>
<th>follower utilities (type 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2    4</td>
<td>1   0</td>
<td>1   0</td>
</tr>
<tr>
<td>1    3</td>
<td>0   1</td>
<td>1   3</td>
</tr>
</tbody>
</table>

*probability*.6  
*probability*.4
Multiple types - visualization

Combined

(R,C)

(0,1,0)
DOBSS [Paruchuri et al. AAMAS‘08]
(MIP for the Bayesian setting)

\[
\max \sum_{l} \sum_{i} \sum_{j} p^l R^l_{i,j} z^l_{i,j}
\]

\[\forall l \sum_{i} \sum_{j} z^l_{i,j} = 1\]

\[\forall l, j \quad q^l_j \leq \sum_{i} z^l_{i,j} \leq 1\]

\[\forall l \sum_{j} q^l_j = 1\]

\[\forall l, j \quad 0 \leq (a^l - \sum_{i} C^l_{i,j} (\sum_{j'} z^l_{i,j'})) \leq (1 - q^l_j) M\]

\[\forall l, i, j \quad z^l_{i,j} \in [0, 1]\]

\[\forall l, j \quad q^l_j \in \{0, 1\}\]

I rows
J columns
\(p^l\) the probability that type \(l\) appears
\(R^l\) is the defenders payoff matrix
\(C^l\) is the attackers payoff matrix
\(z^l_{i,j}\) is the probability that row \(i\) and column \(j\) are played against type \(l\)
\(q^l_j = 1\) when type \(l\)’s best response is column \(j\)
\(M\) is a large number
(In)approximability of Bayesian games

[Letchford et al. SAGT’09]

• (# types)-approximation: optimize for each type separately using the LP method. Pick the solution that gives the best expected utility against the entire type distribution.

• Can’t do any better in polynomial time, unless P=NP
  – Reduction from INDEPENDENT-SET

• For adversarially chosen types, cannot decide in polynomial time whether it is possible to guarantee positive utility, unless P=NP
  – Again, a MIP formulation can be given
Reduction from independent set

leader utilities

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_l^1$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$a_l^2$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$a_l^3$</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

follower utilities (type 1)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_l^1$</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$a_l^2$</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>$a_l^3$</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

follower utilities (type 2)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_l^1$</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>$a_l^2$</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$a_l^3$</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

follower utilities (type 3)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_l^1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$a_l^2$</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>$a_l^3$</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Security games [Kiekintveld et al. AAMAS’09]

- Makes a simple assumption, namely that payoffs only depend on the identify of the target attacked and if that target is defended or not.
  - Often combined with an assumption that the defender is always better off when the attacked target is defended, and the attacker is always better off when the attacked target is undefended.

<table>
<thead>
<tr>
<th></th>
<th>Defended</th>
<th>Undefended</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>(5,-10)</td>
<td>(-20,30)</td>
</tr>
<tr>
<td>$T_2$</td>
<td>(10,-10)</td>
<td>(0,0)</td>
</tr>
</tbody>
</table>
**ERASER** [Kiekintveld et al. AAMAS’09]  
(MIP for multiple resources)

\[ \begin{align*}
\max & \quad d \\
\sum_{t} a_t &= 1 \\
\sum_{t} c_t &\leq m \\
\forall_t & \quad d - \sum_{t} (R_t^d c_t + R_t^u (1 - c_t)) \leq (1 - a_t)M \\
\forall_t & \quad 0 \leq k - \sum_{t} (C_t^d c_t + C_t^u (1 - c_t)) \leq (1 - a_t)M \\
& \quad a_t \in \{0, 1\} \\
& \quad c_t \in [0, 1]
\end{align*} \]

- **T** is the set of targets
- **R_t** is the defender’s payoff given *t*  
  (d – defended and u – undefended)
- **C_t** is the attacker’s payoff given *t*  
  (d – defended and u – undefended)
- **c_t** is the probability that the defender defends target *t*
- **m** is the number of defense resources the defender has
- **a_t = 1** for the target that is attacked
- **M** is a large number
Defense at a cost [Letchford & Vorobeychik UAI’12]

- Each target $t$ is assigned a cost $c_t$
- The defender can choose to pay $c_t$ to prevent an attack that originates at target $t$
  - Multiple options with different efficiency may be available
  - This cost is paid even when $t$ is not attacked
- Paying a fraction of $c_t$ will offer partial protection
  - This corresponds to playing a mixed strategy
- The existing MIP can be modified to solve this problem
  - Furthermore, using techniques similar to what are used in\(^1\) we are able to give an efficient linear programming formulation for the problem

\(^1\)Conitzer and Sandholm 2006
Goals

• Two conflicting goals for the defender:
  – Minimize expected loss from an attack
  – Minimize amount spent on defense
  – We define an optimal solution as one that minimizes the sum of these two values

• Simple goal for the attacker:
  – Given a defense strategy for the Defender, attack the target that gives the largest expected payoff
  – In the zero-sum case, this corresponds to hurting the Defender as much as possible
Linear program for defense costs

\[ \forall \hat{t} \text{ maximize } d^\hat{t} - \sum_t \sum_o p_{o,t} c_{o,t}^\hat{t} \]
\[ \text{s.t.} \]
\[ \forall o,t \quad c_{o,t}^\hat{t} \in [0, 1] \]
\[ \forall t \quad \sum_o c_{o,t}^\hat{t} = 1 \]
\[ d^\hat{t} - \sum_o \left( R_{o,\hat{t}} c_{o,\hat{t}}^\hat{t} \right) = 0 \]
\[ k^\hat{t} - \sum_o \left( C_{o,\hat{t}} c_{o,\hat{t}}^\hat{t} \right) = 0 \]
\[ \forall t \quad 0 \leq k^\hat{t} - \sum_o \left( C_{o,t} c_{o,t}^\hat{t} \right) \]

T is the set of targets
O is the set of defense options
R_{o,t} is the defenders payoff given o and t
C_{o,t} is the attackers payoff given o and t
p_{o,t} is the price (cost) of defense option o for target t
\( c_{o,t} \) is the probability that the defender defends target t with option o
A simple example
(car supply chain)

- Produces electronic components
- Produces engine components
- Produces engines
- Produces radios
- Produces Micro-car
- Produces SUV
Valuation

Produces Micro-car

Value = 2.168

Value = 1.227

Produces SUV
Attack model
(Independent cascade)

Spread chance: .5
Evaluating expected loss

Cascade model

Values

Expected loss = 0.8488

Expected loss = 0.6135

Expected loss = 1.227
Evaluating expected loss

• Related to the idea of maximizing influence
  – Which unfortunately is NP-hard$^1$

• However, extremely easy to approximate through simulation$^1$

• Fast algorithms for special cases
  – Two-pass algorithm for undirected trees (O(|T|))
  – Domain specific procedures (i.e. consequence analysis)

$^1$Kempe et al. 2003
Notation for two-pass algorithm for undirected trees

- $T$: set of targets
- $O$: set of defense options
- $z_{o,t}$: probability of an attack succeeding at target $t$ under defense option $o$
- $p_{t,t'}$: probability of a cascade from $t$ to $t'$
- $N_t$: set of the neighbors of $t$
- $w_t$: worth (value) of target $t$
- $P_t$: Parent of $t$
Expected loss in trees

• Expected loss due to cascading failure:

$$z_{o,t} \sum_{t' \neq t} w_{t'} p(\text{failure}(t') | t)$$

• $$p(\text{failure}(t') | t) = \text{product of probabilities of the edges on the path between } t \text{ and } t'$$

• By organizing these paths, we can express the expected loss of the contagion spreading across an edge $$(t, t')$$ as:

$$E[U_{(t, t')}] = p_{t,t'} \left( w_{t'} + \sum_{t'' \in N_{t'}, t'' \neq t} E[U_{(t', t'')}] \right)$$
Utility evaluation

Given expected losses over edges, we can calculate expected losses for a target $t$:

$$U_t = w_t + \sum_{t' \in N_t} E[U_{(t,t')}]$$
Two-pass algorithm for undirected trees

• Pick a random node to be the root

• Break each edge into two directed edges

• Upward pass
  – Calculate expected loss for each edge from parent to child

• Downward pass
  – Calculate expected loss for each edge from child to parent
Simple example

\[ P(\text{spread}) = 0.5 \]

\[ w = 0 \]

\[ w = 1 \]

\[ w = 1 \]
Upward pass

P(spread) = 0.5 

\[ P(spread) = 0.5 \]

\[ 0.5 \times 1 = 0.5 \]

\[ 0.5 = 0.5 \times 1 \]

w = 0

w = 1

w = 1
Downward pass

\[ \text{P(spread)} = 0.5 \]

\[ 0.5 \times 0.5 = 0.25 \]

\[ 25 = 0.5 \times 0.5 \]

w = 0

w = 1

w = 1

\[ 0.5 \times 0.5 = 0.25 \]

\[ 25 = 0.5 \times 0.5 \]
Expected loss calculation

\[ P(\text{spread}) = 0.5 \]

\[ w = 0 \]

\[ U = 1 \]

\[ w = 1 \]

\[ U = 1.25 \]

\[ 0.25 \]

\[ 0.5 \]

\[ 25 \]

\[ 25 \]

\[ 0.5 \]

\[ U = 1.25 \]

\[ w = 1 \]
Correctness

• We model this as a message passing algorithm
  – To calculate $E[U_{(t',t)}]$ requires the messages from $N_t \setminus t'$ to $t$

• Upward
  – Each node has only one parent
  – All children have previously passed messages to $t$
  – Thus, each node has $N_t \setminus P_t$ available when generating the message to its parent

• Downward pass
  – All children passed messages to $t$ in the upward pass
  – Parent has already passed message in this pass
  – Thus, each node has $N_t$ available
Achieving linear time

- As given, doesn’t achieve linear time
  - A node with $O(n)$ edges (star) requires $O(n^2)$ edge queries

- To get around this, need to store values at the nodes
  - We can reason that:
    \[
    U_t = w_t + \sum_{t \in N_t} p_{t,t'} (U_{t'} - E[U_{(t',t)}])
    \]

- Store a running total at each node
  - By the same reasoning as before the necessary calculations have been performed before they are needed as inputs
  - However, now need to show that we can recover the needed values from the stored value
Recovering the correct values

- **Upward pass**
  - Already processed all of the children but not the parent
  - Value stored is exactly what is needed

- **Downward pass**
  - Value stored is not correct
  - Children have not been updated yet
  - Can subtract out the value stored at each child to recover the needed values
• Visit each edge twice
  – Once on upward and once on downward pass

• Perform a constant amount of work each time
  – Need to query the source of the edge
  – On downward pass also need to query the target of the edge
  – Also need to update the target of the edge

• Since this is a tree, $|T| - 1 = |E|$
  – Thus, runtime is $O(|T|)$
Approximation through simulation

1: Take each edge with probability proportional to its spread chance

2: Propagate values from each node to every node that can reach it in the induced graph

3: Take the average
Formulating this as a game (zero sum)

If target 0 is attacked

<table>
<thead>
<tr>
<th></th>
<th>Defended</th>
<th>Not Defended</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(0,0)</td>
<td>(-0.8488,0.8488)</td>
</tr>
</tbody>
</table>
Optimal defense strategy (zero sum)

Expected loss

Uniform cost of .1428

P(Defense) = .28

P(Defense) = 0

P(Defense) = .5
Optimal defense strategy

(zero sum)

Expected loss

Uniform cost of \(0.0179\)
Optimal defense strategy
(zero sum)

Expected loss

Uniform cost of .5714

P(Defense) = 0

P(Defense) = .1