The Multilinear Minimax Relaxation of Bimatrix Games and Comparison with Nash Equilibria via Lemke-Howson

Chun Leung Lau (Jointly with B. Kalantari)

Department of Computer Science, Rutgers University, New Brunswick, New Jersey, USA

30th International Conference on Economics
July 16, 2019
• Von Neumann Minimax Theorem (Two Players)
• Von Neumann Minimax Theorem (Two Players)
• Nash Equilibria in the Bimatrix Case
Outline

- Von Neumann Minimax Theorem (Two Players)
- Nash Equilibria in the Bimatrix Case
- Explanation of MMR as an Approximation of Nash Equilibria
• Von Neumann Minimax Theorem (Two Players)
• Nash Equilibria in the Bimatrix Case
• Explanation of MMR as an Approximation of Nash Equilibria
• Experimental Results
Outline

- Von Neumann Minimax Theorem (Two Players)
- Nash Equilibria in the Bimatrix Case
- Explanation of MMR as an Approximation of Nash Equilibria
- Experimental Results
- Balanced Scaling
For a positive integer $k$ denote the unit simplex by

$$S_k = \{ u \in \mathbb{R}^k : \sum_{j=1}^{k} u_j = 1, \, u \geq 0 \}.$$  \hspace{1cm} (1)

**Figure:** The unit simplex $S_k$. 

(Rutgers)
Von Neumann Minimax Theorem - Zero-Sum Games

Theorem

Let \( A = (a_{ij}) \) be an \( m \times n \) real matrix.

\[
\max_{x \in S_m} \min_{y \in S_n} x^T Ay = \min_{y \in S_n} \max_{x \in S_m} x^T Ay,
\]

where \( S_k \) is the unit simplex in \( \mathbb{R}^k \), i.e.

\[
S_k = \{ x \in \mathbb{R}^k : \sum_{i=1}^{k} x_i = 1, \ x \geq 0 \}.
\]
Nash Equilibria - Non-Cooperative Games

Theorem

(Nash) Let $R = (R_{ij})$ and $C = (C_{ij})$ be $m \times n$ real payoff matrices. Given $p = (x, y) \in S_m \times S_n$, set

$$R[p] = x^T R y, \quad C[p] = x^T C y.$$ 

There exists a composite vector $p^* = (x^*, y^*)$ (called Nash Equilibrium) such that

$$\max \{ R[p] : p = (x, y^*), \quad x \in S_m \} = R[p^*]$$

and

$$\max \{ C[p] : p = (x^*, y), \quad y \in S_n \} = C[p^*].$$
Theorem

Let \( R = (R_{ij}) \) and \( C = (C_{ij}) \) be \( m \times n \) real payoff matrices. Given \( p = (x, y) \in S_m \times S_n \), set

\[
R[p] = x^T R y, \quad C[p] = x^T C y.
\]

There exists a composite vector \( p^* = (x^*, y^*) \) (called Nash Equilibrium) such that

\[
\max \{ R[p] : p = (x, y^*), \; x \in S_m \} = R[p^*]
\]

and

\[
\max \{ C[p] : p = (x^*, y), \; y \in S_n \} = C[p^*].
\]

It is not known if \( p^* \) can be computed efficiently.
The Multilinear Minimax Relaxation (MMR) Theorem

**Theorem**

(Kalantari) Set \( \Delta = S_m \times S_n \). Given \( \alpha = (\alpha_1, \alpha_2) \in S_2 \) and \( p = (x, y) \in \Delta \), let

\[
A_0[\alpha, p] = \alpha_1 R[p] + \alpha_2 C[p],
\]

(2)

Then,

\[
\min_{\alpha \in S_2} \max_{p \in \Delta} A_0[\alpha, p] = \max_{p \in \Delta} \min_{\alpha \in S_2} A_0[\alpha, p].
\]

(3)

Let \( \alpha^* \) and \( p^* = (x^*, y^*) \) be a minimax solution. Then, for any \((\alpha, p) \in S_2 \times \Delta\) we have

\[
A_0[\alpha^*, p] \leq A_0[\alpha^*, p^*] \leq A_0[\alpha, p^*]
\]

(4)

Furthermore, \((\alpha^*, p^*)\) is computable via a primal-dual pair of linear programming with \(O(m \times n)\) variables.
The game is played as a meta-player against Row and Column.

The meta-player chooses an $\alpha \in S_2$. Row chooses $x \in S_m$ and Column chooses $y \in S_n$.

Once $\alpha$ and $p = (x, y)$ are chosen, the meta-player’s expected loss is $A_0[\alpha, p]$, while Row and Column gains the same amount.

Under pure strategies, the meta-player plays against either Row and Column.
Corollary

Let $p_*$ be a Nash equilibrium and $p^*$ a solution to MMR. Then

$$\alpha_1^* R[p_*] + \alpha_2^* C[p_*] \leq \alpha_1^* R[p^*] + \alpha_2^* C[p^*].$$

In other words we can approximate a convex combination of $R$ and $C$ payoffs under Nash equilibrium.

Corollary

Either $R[p_*] \leq R[p^*]$, or $C[p_*] \leq C[p^*]$. In particular, if both inequalities are satisfied, then $p^*$ gives a better payoff than $p_*$ for both players.
Algorithmic Computation of MMR

The primal-dual LP formulation of MMR is the following:

\[
\begin{align*}
\text{(LP)} & \quad \min \delta \\
& \quad R_{ij} \alpha_1 + C_{ij} \alpha_2 \leq \delta, \quad \forall i \in [1, n], j \in [1, m], \\
& \quad \alpha_1 + \alpha_2 = 1, \\
& \quad \alpha_1 \geq 0, \\
& \quad \alpha_2 \geq 0.
\end{align*}
\]
Algorithmic Computation of MMR

\[ \text{max } \lambda \]

\[ \sum_{i=1}^{n} \sum_{j=1}^{m} R_{ij} q_{ij} \geq \lambda, \]

\[ \sum_{i=1}^{n} \sum_{j=1}^{m} C_{ij} q_{ij} \geq \lambda, \]

\[ \sum_{i=1}^{n} \sum_{j=1}^{m} q_{ij} = 1, \]

\[ q_{ij} \geq 0, \quad \forall i \in [1, n], j \in [1, m]. \]
Let \( Q \) be the \( m \times n \) matrix \((q_{ij})\). Given a positive integer \( k \) let \( e_k = (1, \ldots, 1)^T \in \mathbb{R}^k \). Then set

\[
x^* = Qe_n \quad y^* = Q^T e_m.
\]

Then \( x^* \) and \( y^* \) are the strategies for \( \text{Row} \) and \( \text{Column} \).
Using Gambit 15.1.1, the entries of payoff matrices are generated randomly between 0 and 1.
Using Gambit 15.1.1, the entries of payoff matrices are generated randomly between 0 and 1.

We compared Lemke-Howson algorithm with MMR algorithm in the LP formulation.
Using Gambit 15.1.1, the entries of payoff matrices are generated randomly between 0 and 1.

We compared Lemke-Howson algorithm with MMR algorithm in the LP formulation.

Programs run for a maximum time of 30 minutes before they are terminated and returned an error. For each comparison, we ran the algorithms 1000 times.
Figure: Speed comparisons for Lemke-Howson and MMR algorithms
## Computational Results

<table>
<thead>
<tr>
<th>Actions</th>
<th>Percentages</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>81.2%</td>
</tr>
<tr>
<td>30</td>
<td>80.3%</td>
</tr>
<tr>
<td>60</td>
<td>79.98%</td>
</tr>
<tr>
<td>100</td>
<td>82.0055%</td>
</tr>
<tr>
<td>150</td>
<td>80.023%</td>
</tr>
</tbody>
</table>

**Figure:** Percentages of times MMR algorithm had better payoffs for both players
### Computational Results

<table>
<thead>
<tr>
<th>Actions</th>
<th>Relative errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.129 ± 0.020</td>
</tr>
<tr>
<td>30</td>
<td>0.090 ± 0.012</td>
</tr>
<tr>
<td>60</td>
<td>0.110 ± 0.010</td>
</tr>
<tr>
<td>100</td>
<td>0.151 ± 0.016</td>
</tr>
<tr>
<td>150</td>
<td>0.141 ± 0.004</td>
</tr>
</tbody>
</table>

**Figure:** Relative errors of payoffs if one player’s MMR payoff did worse than the Nash equilibrium payoff.

As shown by the experimental results, MMR algorithm outperforms Lemke-Howson both in speed and the quality of payoffs.
As shown by the experimental results, MMR algorithm outperforms Lemke-Howson both in speed and the quality of payoffs.

Figure: Relative errors of payoffs if one player’s MMR payoff did worse than the Nash equilibrium payoff

<table>
<thead>
<tr>
<th>Actions</th>
<th>Relative errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.129 ± 0.020</td>
</tr>
<tr>
<td>30</td>
<td>0.090 ± 0.012</td>
</tr>
<tr>
<td>60</td>
<td>0.110 ± 0.010</td>
</tr>
<tr>
<td>100</td>
<td>0.151 ± 0.016</td>
</tr>
<tr>
<td>150</td>
<td>0.141 ± 0.004</td>
</tr>
</tbody>
</table>
Balanced Scaling Theorem

**Definition**

A profile is called *balanced* if both expected payoffs are equal.

**Theorem**

Given payoff matrices $R, C \in [0, 1]^{n \times n}$, for each $t \in [0, 1]$ let $p^*(t) = (p^*_1(t), p^*_2(t))$ be the MMR solution corresponding to the case when $R$ is replaced by $tR$ and $C$ by $(1 - t)C$. Then there exists $t^* \in [0, 1]$ that gives a balanced scaling, i.e.

$$t^* R[p^*(t^*)] = (1 - t^*) C[p^*(t^*)],$$

(5)

To compute a convex scaling that gives balanced MMR payoffs, we can use the Bisection Method to change the value of $t$. 

(Rutgers)
**Balanced Scaling Theorem**

**Definition**
A profile is called *balanced* if both expected payoffs are equal.

**Theorem**
Given payoff matrices $R, C \in [0, 1]^{n \times n}$, for each $t \in [0, 1]$ let $p^*(t) = (p^1(t), p^2(t))$ be the MMR solution corresponding to the case when $R$ is replaced by $tR$ and $C$ by $(1 - t)C$. Then there exists $t_\ast \in [0, 1]$ that gives a balanced scaling, i.e.

$$t_\ast R[p^*(t_\ast)] = (1 - t_\ast)C[p^*(t_\ast)],$$

(5)
**Balanced Scaling Theorem**

**Definition**

A profile is called *balanced* if both expected payoffs are equal.

**Theorem**

Given payoff matrices $R, C \in [0, 1]^{n \times n}$, for each $t \in [0, 1]$ let $p^*(t) = (p_1^*(t), p_2^*(t))$ be the MMR solution corresponding to the case when $R$ is replaced by $tR$ and $C$ by $(1 - t)C$. Then there exists $t_* \in [0, 1]$ that gives a balanced scaling, i.e.

$$t_* R[p^*(t_*)] = (1 - t_*) C[p^*(t_*)], \quad (5)$$

To compute a convex scaling that gives balanced MMR payoffs, we can use the Bisection Method to change the value of $t$. 
The advantage is that there exists a convex scaling pair $d = (t, 1 - t) \in S_2$ so that the MMR expected payoffs for Row and Column players are equal.

At least one player’s expected payoff is at least as good as the expected payoffs of any Nash equilibria.
Visualizations of Balanced Scalings

![Graph showing expected payoffs for Row Player and Column Player.](image)
Based on extensive computational results, MMR is an effective approximation to Nash Equilibria.
Based on extensive computational results, MMR is an effective approximation to Nash Equilibria.

In a good percentage of time, it out-performs Lemke-Howson in speed and in quality of solutions.
• Based on extensive computational results, MMR is an effective approximation to Nash Equilibria.

• In a good percentage of time, it out-performs Lemke-Howson in speed and in quality of solutions.

• Because Nash equilibria can be applied to machine-learning algorithms and robotics, we expect the applicability of MMR in such problems.


THANK YOU!