Two-player game

Two-player game \((A, B)\) given by matrices \(A, B \in \mathbb{R}^{m \times n}\).
Two-player game

Two-player game \((A, B)\) given by matrices \(A, B \in \mathbb{R}^{m \times n}\).
- Alice plays mixed strategy in \(\Delta_A\) over the \(m\) rows.

**Example**

Alice has \(S_A = \{a_1, a_2\}\) and \(S_B = \{b_1, b_2, b_3\}\).

\[A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix}\]

\[B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}\]

Suppose that \(x = (1, 0)\) and \(y = (0.5, 0, 0.5)\), then

\[C_{\text{Bob}}(x, y) = x^T By = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0 \\ 0.5 \end{pmatrix} = 2.2\]
Two-player game

Two-player game \((A, B)\) given by matrices \(A, B \in \mathbb{R}^{m \times n}\).

- Alice plays mixed strategy in \(\Delta_A\) over the \(m\) rows.
- Bob plays mixed strategy in \(\Delta_B\) over \(n\) columns.

Example

Alice has \(S_A = \{a_1, a_2\}\) and \(S_B = \{b_1, b_2, b_3\}\).

\(A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix}\) and \(B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}\).

Suppose that \(x = (1, 0)\) and \(y = (0.5, 0, 0.5)\), then

\[
C_{Bob}(x, y) = x^T B y = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0 \\ 0.5 \end{pmatrix} = 2.2.
\]
Two-player game

Two-player game $(A, B)$ given by matrices $A, B \in \mathbb{R}^{m \times n}$.

- Alice plays mixed strategy in $\Delta_A$ over the $m$ rows.
- Bob plays mixed strategy in $\Delta_B$ over $n$ columns.

For $x \in \Delta_A$ and $y \in \Delta_B$, (expected) cost given by

$$C_{\text{Alice}}(x, y) = x^T Ay, \quad C_{\text{Bob}}(x, y) = x^T By.$$
Two-player game

Two-player game \((A, B)\) given by matrices \(A, B \in \mathbb{R}^{m \times n}\).

- Alice plays mixed strategy in \(\Delta_A\) over the \(m\) rows.
- Bob plays mixed strategy in \(\Delta_B\) over \(n\) columns.

For \(x \in \Delta_A\) and \(y \in \Delta_B\), (expected) cost given by

\[
C_{\text{Alice}}(x, y) = x^T Ay, \quad C_{\text{Bob}}(x, y) = x^T By.
\]

Example

Alice has \(S_A = \{a_1, a_2\}\) and \(S_B = \{b_1, b_2, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]
Two-player game

Two-player game \((A, B)\) given by matrices \(A, B \in \mathbb{R}^{m \times n}\).
- Alice plays mixed strategy in \(\Delta_A\) over the \(m\) rows.
- Bob plays mixed strategy in \(\Delta_B\) over \(n\) columns.

For \(x \in \Delta_A\) and \(y \in \Delta_B\), (expected) cost given by
\[
C_{\text{Alice}}(x, y) = x^T A y, \quad C_{\text{Bob}}(x, y) = x^T B y.
\]

Example

Alice has \(S_A = \{a_1, a_2\}\) and \(S_B = \{b_1, b_2, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

Suppose that \(x = (1, 0)\) and \(y = (0.5, 0, 0.5)\), then
\[
C_{\text{Bob}}(x, y) = x^T B y = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0 \\ 0.5 \end{pmatrix} = 2.
\]
Two-player game

Two-player game \((A, B)\) given by matrices \(A, B \in \mathbb{R}^{m \times n}\).

- Alice plays mixed strategy in \(\Delta_A\) over the \(m\) rows.
- Bob plays mixed strategy in \(\Delta_B\) over \(n\) columns.

For \(x \in \Delta_A\) and \(y \in \Delta_B\), (expected) cost given by
\[
C_{\text{Alice}}(x, y) = x^T Ay, \quad C_{\text{Bob}}(x, y) = x^T By.
\]

Example

Alice has \(S_A = \{a_1, a_2\}\) and \(S_B = \{b_1, b_2, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

Suppose that \(x = (1, 0)\) and \(y = (0.5, 0, 0.5)\), then
\[
C_{\text{Bob}}(x, y) = x^T By = (1 \ 0) \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0 \\ 0.5 \end{pmatrix} = 2.
\]
Mixed Nash equilibrium (MNE)

We will use the “best response” version of the MNE definition.
Mixed Nash equilibrium (MNE)

We will use the “best response” version of the MNE definition.

Definition

Column $b_j$ is **best response against** $x$ for Bob if $(x^T B)_j = \min_k (x^T B)_k$. 

For $(x^T B) = (2, 4, 2)^T$, we have $(x^T B)_1 = 2$, $(x^T B)_2 = 4$ and $(x^T B)_3 = 2$. 

Mixed Nash equilibrium (MNE)

We will use the “best response” version of the MNE definition.

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column $b_j$ is <strong>best response against</strong> $x$ for Bob if $(x^T B)_j = \min_k (x^T B)_k$.</td>
</tr>
<tr>
<td>Row $a_i$ is <strong>best response against</strong> $y$ for Alice if $(A y)_i = \min_k (A y)_k$.</td>
</tr>
</tbody>
</table>
Mixed Nash equilibrium (MNE)

We will use the “best response” version of the MNE definition.

**Definition**

Column $b_j$ is best response against $x$ for Bob if $(x^T B)_j = \min_k (x^T B)_k$.

Row $a_i$ is best response against $y$ for Alice if $(Ay)_i = \min_k (Ay)_k$.

- For $(x^T B) = (2, 4, 2)^T$, we have $(x^T B)_1 = 2$, $(x^T B)_2 = 4$ and $(x^T B)_3 = 2$. 
Mixed Nash equilibrium (MNE)

We will use the “best response” version of the MNE definition.

**Definition**

Column $b_j$ is **best response** against $x$ for Bob if $(x^T B)_j = \min_k (x^T B)_k$.

Row $a_i$ is **best response** against $y$ for Alice if $(Ay)_i = \min_k (Ay)_k$.

- For $(x^T B) = (2, 4, 2)^T$, we have $(x^T B)_1 = 2$, $(x^T B)_2 = 4$ and $(x^T B)_3 = 2$.

**Definition (MNE, best response version)**

Pair $(x^*, y^*)$ is MNE if and only if

$$
\begin{align*}
x^*_i > 0 & \implies (Ay^*)_i = \min_k (Ay^*)_k, \quad \forall i = 1, \ldots, m, \\
y^*_j > 0 & \implies ((x^*)^T B)_j = \min_k ((x^*)^T B)_k, \quad \forall j = 1, \ldots, n.
\end{align*}
$$
Mixed Nash equilibrium (MNE)

We will use the “best response” version of the MNE definition.

Definition

Column $b_j$ is best response against $x$ for Bob if $(x^T B)_j = \min_k(x^T B)_k$. Row $a_i$ is best response against $y$ for Alice if $(A y)_i = \min_k(A y)_k$.

For $(x^T B) = (2, 4, 2)^T$, we have $(x^T B)_1 = 2$, $(x^T B)_2 = 4$ and $(x^T B)_3 = 2$.

Definition (MNE, best response version)

Pair $(x^*, y^*)$ is $\epsilon$-MNE if and only if

\[
x_i^* > 0 \quad \Rightarrow \quad (A y^*)_i \leq \min_k(A y^*)_k + \epsilon \quad \forall i = 1, \ldots, m,
\]

\[
y_j^* > 0 \quad \Rightarrow \quad ((x^*)^T B)_j \leq \min_k((x^*)^T B)_k + \epsilon \quad \forall j = 1, \ldots, n.
\]

That is, players only assign positive probability to best responses.
Mixed Nash equilibrium (MNE)

We will use the “best response” version of the MNE definition.

**Definition**

Column $b_j$ is best response against $x$ for Bob if $(x^T B)_j = \min_k (x^T B)_k$. Row $a_i$ is best response against $y$ for Alice if $(Ay)_i = \min_k (Ay)_k$.

For $(x^T B) = (2, 4, 2)^T$, we have $(x^T B)_1 = 2$, $(x^T B)_2 = 4$ and $(x^T B)_3 = 2$.

**Definition (MNE, best response version)**

Pair $(x^*, y^*)$ is $\epsilon$-MNE if and only if

$x^*_i > 0 \Rightarrow (Ay^*)_i \leq \min_k (Ay^*)_k + \epsilon \quad \forall \ i = 1, \ldots, m,$

$y^*_j > 0 \Rightarrow ((x^*)^T B)_j \leq \min_k ((x^*)^T B)_k + \epsilon \quad \forall \ j = 1, \ldots, n.$

That is, players only assign positive probability to best responses.

Strategies that get positive probability assigned to them play special role.
Recap from Lecture 4

Theorem (Nash's theorem, 1950)
Any finite game $\Gamma$ has a mixed Nash equilibrium. Probably no polynomial time algorithm exists for computing one. PPAD-hardness.

In two-player zero-sum games $(A, B)$, where $A + B = 0$, computing an MNE can be reduced to solving a linear program.

We also saw fictitious play, where empirical beliefs of other player's mixed strategy "converge" to MNE. Today, the goal is to give a "quasi-polynomial" time algorithm that computes an $\epsilon$-approximate mixed Nash equilibrium. Supports of mixed strategies play an important role here.
Recap from Lecture 4

Theorem (Nash’s theorem, 1950)

Any finite game $\Gamma$ has a mixed Nash equilibrium.
Theorem (Nash’s theorem, 1950)

Any finite game $\Gamma$ has a mixed Nash equilibrium.

- Probably no polynomial time algorithm exists for computing one.
Theorem (Nash’s theorem, 1950)

Any finite game $\Gamma$ has a mixed Nash equilibrium.

- Probably no polynomial time algorithm exists for computing one.
  - PPAD-hardness.
Recap from Lecture 4

Theorem (Nash’s theorem, 1950)

Any finite game $\Gamma$ has a mixed Nash equilibrium.

- Probably no polynomial time algorithm exists for computing one.
  - PPAD-hardness.

In two-player zero-sum games $(A, B)$,
Recap from Lecture 4

**Theorem (Nash’s theorem, 1950)**

Any finite game \( \Gamma \) has a mixed Nash equilibrium.

- Probably no polynomial time algorithm exists for computing one.
  - PPAD-hardness.

In two-player zero-sum games \((A, B)\), where \(A + B = 0\),
Recap from Lecture 4

Theorem (Nash’s theorem, 1950)

Any finite game $\Gamma$ has a mixed Nash equilibrium.

- Probably no polynomial time algorithm exists for computing one.
  - PPAD-hardness.

In two-player zero-sum games $(A, B)$, where $A + B = 0$, computing an MNE can be reduced to solving a linear program.
Theorem (Nash’s theorem, 1950)

Any finite game $\Gamma$ has a mixed Nash equilibrium.

- Probably no polynomial time algorithm exists for computing one.
- PPAD-hardness.

In two-player zero-sum games $(A, B)$, where $A + B = 0$, computing an MNE can be reduced to solving a linear program.

- We also saw fictitious play,
Theorem (Nash’s theorem, 1950)

Any finite game \( \Gamma \) has a mixed Nash equilibrium.

- Probably no polynomial time algorithm exists for computing one.
  - PPAD-hardness.

In two-player zero-sum games \((A, B)\), where \( A + B = 0 \), computing an MNE can be reduced to solving a linear program.

- We also saw fictitious play, where empirical beliefs of other player’s mixed strategy “converge” to MNE.
Theorem (Nash’s theorem, 1950)

Any finite game $\Gamma$ has a mixed Nash equilibrium.

- Probably no polynomial time algorithm exists for computing one.
  - PPAD-hardness.

In two-player zero-sum games $(A, B)$, where $A + B = 0$, computing an MNE can be reduced to solving a linear program.

- We also saw fictitious play, where empirical beliefs of other player’s mixed strategy “converge” to MNE.

Today, the goal is to give a “quasi-polynomial” time algorithm that computes an $\epsilon$-approximate mixed Nash equilibrium.
Recap from Lecture 4

Theorem (Nash’s theorem, 1950)

Any finite game \( \Gamma \) has a mixed Nash equilibrium.

- Probably no polynomial time algorithm exists for computing one.
  - PPAD-hardness.

In two-player zero-sum games \((A, B)\), where \( A + B = 0 \), computing an MNE can be reduced to solving a linear program.

- We also saw fictitious play, where empirical beliefs of other player’s mixed strategy “converge” to MNE.

Today, the goal is to give a “quasi-polynomial” time algorithm that computes an \( \epsilon \)-approximate mixed Nash equilibrium.

Supports of mixed strategies play an important role here.
Support of mixed strategies
Support of mixed strategy

The support of a mixed strategy $x \in \Delta_A$ is

Similarly, for $y \in \Delta_B$ it is

Example (cont'd)

Suppose again that $x = (1, 0)$ and $y = (0.5, 0, 0.5)$. Then

$\text{Supp}(x) = \{a_1\}$ and $\text{Supp}(y) = \{b_1, b_3\}$.

Does it help if one knows the supports of an equilibrium? Yes!

Remark

Informally speaking, knowing the support of an ($\epsilon$-)MNE is enough to be able to efficiently compute one.

Once the support is fixed, the computation of an equilibrium (with that support) reduces to solving a linear program.
Support of mixed strategy

The support of a mixed strategy \( x \in \Delta_A \) is

\[
\text{Supp}(x) = \{ a_i : x_i > 0 \text{ for } i = 1, \ldots, m \} \subseteq S_A.
\]
The support of a mixed strategy $x \in \Delta_A$ is
$$\text{Supp}(x) = \{ a_i : x_i > 0 \text{ for } i = 1, \ldots, m \} \subseteq S_A.$$ 
Similarly, for $y \in \Delta_B$ it is
$$\text{Supp}(y) = \{ b_j : y_j > 0 \text{ for } j = 1, \ldots, n \} \subseteq S_B.$$
Support of mixed strategy

The **support** of a mixed strategy $x \in \Delta_A$ is

$$\text{Supp}(x) = \{ a_i : x_i > 0 \text{ for } i = 1, \ldots, m \} \subseteq S_A.$$ 

Similarly, for $y \in \Delta_B$ it is

$$\text{Supp}(y) = \{ b_j : y_j > 0 \text{ for } j = 1, \ldots, n \} \subseteq S_B.$$ 

**Example (cont’d)**

Suppose again that $x = (1, 0)$ and $y = (0.5, 0, 0.5)$. Then $\text{Supp}(x) = \{ a_1 \}$ and $\text{Supp}(y) = \{ b_1, b_3 \}$. 

Yes! Informally speaking, knowing the support of an (\(\epsilon\)-)MNE is enough to be able to efficiently compute one. Once the support is fixed, the computation of an equilibrium (with that support) reduces to solving a linear program.
Support of mixed strategy

The support of a mixed strategy $x \in \Delta_A$ is

$$\text{Supp}(x) = \{ a_i : x_i > 0 \text{ for } i = 1, \ldots, m \} \subseteq S_A.$$ 

Similarly, for $y \in \Delta_B$ it is

$$\text{Supp}(y) = \{ b_j : y_j > 0 \text{ for } j = 1, \ldots, n \} \subseteq S_B.$$ 

Example (cont’d)

Suppose again that $x = (1, 0)$ and $y = (0.5, 0, 0.5)$. Then
$$\text{Supp}(x) = \{ a_1 \} \text{ and } \text{Supp}(y) = \{ b_1, b_3 \}.$$ 

Does it help if one knows the supports of an equilibrium?
The support of a mixed strategy $x \in \Delta_A$ is
\[
\text{Supp}(x) = \{ a_i : x_i > 0 \text{ for } i = 1, \ldots, m \} \subseteq S_A.
\]
Similarly, for $y \in \Delta_B$ it is
\[
\text{Supp}(y) = \{ b_j : y_j > 0 \text{ for } j = 1, \ldots, n \} \subseteq S_B.
\]

Example (cont’d)
Suppose again that $x = (1, 0)$ and $y = (0.5, 0, 0.5)$. Then
\[
\text{Supp}(x) = \{ a_1 \} \text{ and Supp}(y) = \{ b_1, b_3 \}.
\]

Does it help if one knows the supports of an equilibrium? Yes!
Support of mixed strategy

The support of a mixed strategy \( x \in \Delta_A \) is
\[
\text{Supp}(x) = \{ a_i : x_i > 0 \text{ for } i = 1, \ldots, m \} \subseteq S_A.
\]

Similarly, for \( y \in \Delta_B \) it is
\[
\text{Supp}(y) = \{ b_j : y_j > 0 \text{ for } j = 1, \ldots, n \} \subseteq S_B.
\]

Example (cont’d)
Suppose again that \( x = (1, 0) \) and \( y = (0.5, 0, 0.5) \). Then
\[
\text{Supp}(x) = \{ a_1 \} \text{ and } \text{Supp}(y) = \{ b_1, b_3 \}.
\]

Does it help if one knows the supports of an equilibrium? Yes!

Remark
Informally speaking, knowing the support of an \((\epsilon-)\)MNE is enough to be able to efficiently compute one.
The support of a mixed strategy \( x \in \Delta_A \) is
\[
\text{Supp}(x) = \{ a_i : x_i > 0 \text{ for } i = 1, \ldots, m \} \subseteq S_A.
\]
Similarly, for \( y \in \Delta_B \) it is
\[
\text{Supp}(y) = \{ b_j : y_j > 0 \text{ for } j = 1, \ldots, n \} \subseteq S_B.
\]

**Example (cont’d)**
Suppose again that \( x = (1, 0) \) and \( y = (0.5, 0, 0.5) \). Then
\[
\text{Supp}(x) = \{ a_1 \} \text{ and } \text{Supp}(y) = \{ b_1, b_3 \}.
\]

**Remark**
Informally speaking, knowing the support of an \((\epsilon-\)MNE is enough to be able to efficiently compute one. Once the support is fixed, the computation of an equilibrium (with that support)
Support of mixed strategy

The support of a mixed strategy \( x \in \Delta_A \) is
\[
\text{Supp}(x) = \{ a_i : x_i > 0 \text{ for } i = 1, \ldots, m \} \subseteq S_A.
\]
Similarly, for \( y \in \Delta_B \) it is
\[
\text{Supp}(y) = \{ b_j : y_j > 0 \text{ for } j = 1, \ldots, n \} \subseteq S_B.
\]

Example (cont’d)
Suppose again that \( x = (1, 0) \) and \( y = (0.5, 0, 0.5) \). Then
\[
\text{Supp}(x) = \{ a_1 \} \text{ and } \text{Supp}(y) = \{ b_1, b_3 \}.
\]

Does it help if one knows the supports of an equilibrium? Yes!

Remark
Informally speaking, knowing the support of an \( (\epsilon-) \)MNE is enough to be able to efficiently compute one. Once the support is fixed, the computation of an equilibrium (with that support) reduces to solving a linear program.
Somewhat more technical, if we know supports Supp\(x^∗\) and Supp\(y^∗\) of an \((\epsilon-)\text{MNE}\) \((x^∗,y^∗)\), but not \(x^∗\) and \(y^∗\) themselves, then there is a linear program to compute MNE with supports Supp\(x^∗\) and Supp\(y^∗\). The linear program does not necessarily return \((x^∗,y^∗)\), but possibly another equilibrium with the same supports. Linear program comes from (best response) MNE definition.

Definition (MNE, best response version) Pair \((x^∗,y^∗)\) is MNE if and only if
\[
x^∗_i > 0 \Rightarrow (\mathbf{A}y^∗)_i = \min_k (\mathbf{A}y^∗)_k \forall i = 1,\ldots,m,
\]
\[
y^∗_j > 0 \Rightarrow ((x^∗)^\text{T}\mathbf{B})_j = \min_k ((x^∗)^\text{T}\mathbf{B})_k \forall j = 1,\ldots,n.
\]
For Alice, expected costs for rows in support should be equal, and minimal compared to rows not in support. For Bob, expected costs for columns in support should be equal, and minimal compared to columns not in support.
Remark (cont’d)

Somewhat more technical, if we know supports \( \text{Supp}(x^*) \) and \( \text{Supp}(y^*) \) of an \((\epsilon-)\text{MNE} \ (x^*, y^*)\),
Remark (cont’d)

Somewhat more technical, if we know supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$ of an $(\epsilon)$MNE $(x^*, y^*)$, but not $x^*$ and $y^*$ themselves,
Remark (cont’d)

Somewhat more technical, if we know supports Supp($x^*$) and Supp($y^*$) of an ($\epsilon$-)MNE ($x^*$, $y^*$), but not $x^*$ and $y^*$ themselves, then there is a linear program to compute MNE with supports Supp($x^*$) and Supp($y^*$).
Remark (cont’d)

Somewhat more technical, if we know supports \( \text{Supp}(x^*) \) and \( \text{Supp}(y^*) \) of an \((\varepsilon-\)MNE \((x^*, y^*)\), but not \( x^* \) and \( y^* \) themselves, then there is a linear program to compute MNE with supports \( \text{Supp}(x^*) \) and \( \text{Supp}(y^*) \).

- The linear program does not necessarily return \((x^*, y^*)\),
Remark (cont’d)

Somewhat more technical, if we know supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$ of an $(\epsilon)$MNE $(x^*, y^*)$, but not $x^*$ and $y^*$ themselves, then there is a linear program to compute MNE with supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$.

- The linear program does not necessarily return $(x^*, y^*)$, but possibly another equilibrium with the same supports.
Somewhat more technical, if we know supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$ of an $(\epsilon)$MNE $(x^*, y^*)$, but not $x^*$ and $y^*$ themselves, then there is a linear program to compute MNE with supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$.

- The linear program does not necessarily return $(x^*, y^*)$, but possibly another equilibrium with the same supports.
- **Linear program comes from (best response) MNE definition.**
Remark (cont’d)

Somewhat more technical, if we know supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$ of an $(\epsilon\text{-})\text{MNE} (x^*, y^*)$, but not $x^*$ and $y^*$ themselves, then there is a linear program to compute MNE with supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$.

- The linear program does not necessarily return $(x^*, y^*)$, but possibly another equilibrium with the same supports.
- **Linear program comes from (best response) MNE definition.**

Definition (MNE, best response version)

Pair $(x^*, y^*)$ is MNE if and only if

$$x^*_i > 0 \implies (Ay^*)_i = \min_k (Ay^*)_k \quad \forall i = 1, \ldots, m,$$

$$y^*_j > 0 \implies ((x^*)^T B)_j = \min_k ((x^*)^T B)_k \quad \forall j = 1, \ldots, n.$$
Remark (cont’d)

Somewhat more technical, if we know supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$ of an $(\epsilon)$-MNE $(x^*, y^*)$, but not $x^*$ and $y^*$ themselves, then there is a linear program to compute MNE with supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$.

- The linear program does not necessarily return $(x^*, y^*)$, but possibly another equilibrium with the same supports.
- **Linear program comes from (best response) MNE definition.**

Definition (MNE, best response version)

Pair $(x^*, y^*)$ is MNE if and only if

$$x_i^* > 0 \Rightarrow (Ay^*)_i = \min_k (Ay^*)_k \quad \forall i = 1, \ldots, m,$$

$$y_j^* > 0 \Rightarrow ((x^*)^T B)_j = \min_k ((x^*)^T B)_k \quad \forall j = 1, \ldots, n.$$  

- For Alice, expected costs for rows in support should be equal,
Remark (cont’d)

Somewhat more technical, if we know supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$ of an $(\epsilon)$-MNE $(x^*, y^*)$, but not $x^*$ and $y^*$ themselves, then there is a linear program to compute MNE with supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$.

- The linear program does not necessarily return $(x^*, y^*)$, but possibly another equilibrium with the same supports.
- Linear program comes from (best response) MNE definition.

Definition (MNE, best response version)

Pair $(x^*, y^*)$ is MNE if and only if

\[
x^*_i > 0 \implies (Ay^*)_i = \min_k (Ay^*)_k \quad \forall i = 1, \ldots, m,
\]

\[
y^*_j > 0 \implies ((x^*)^T B)_j = \min_k ((x^*)^T B)_k \quad \forall j = 1, \ldots, n.
\]

- For Alice, expected costs for rows in support should be equal, and minimal compared to rows not in support.
Remark (cont’d)

Somewhat more technical, if we know supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$ of an $(\epsilon)$MNE $(x^*, y^*)$, but not $x^*$ and $y^*$ themselves, then there is a linear program to compute MNE with supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$.

- The linear program does not necessarily return $(x^*, y^*)$, but possibly another equilibrium with the same supports.
- **Linear program comes from (best response) MNE definition.**

Definition (MNE, best response version)

Pair $(x^*, y^*)$ is MNE if and only if

$$x^*_i > 0 \Rightarrow (Ay^*)_i = \min_k (Ay^*)_k \quad \forall i = 1, \ldots, m,$$

$$y^*_j > 0 \Rightarrow ((x^*)^TB)_j = \min_k ((x^*)^TB)_k \quad \forall j = 1, \ldots, n.$$

- For Alice, expected costs for rows in support should be equal, and minimal compared to rows not in support.
- For Bob, expected costs for columns in support should be equal,
Remark (cont’d)

Somewhat more technical, if we know supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$ of an $(\epsilon)$MNE $(x^*, y^*)$, but not $x^*$ and $y^*$ themselves, then there is a linear program to compute MNE with supports $\text{Supp}(x^*)$ and $\text{Supp}(y^*)$.

- The linear program does not necessarily return $(x^*, y^*)$, but possibly another equilibrium with the same supports.
- **Linear program comes from (best response) MNE definition.**

Definition (MNE, best response version)

Pair $(x^*, y^*)$ is MNE if and only if

\[
\begin{align*}
x_i^* > 0 & \implies (Ay^*)_i = \min_k (Ay^*)_k & \forall i = 1, \ldots, m, \\
y_j^* > 0 & \implies ((x^*)^T B)_j = \min_k ((x^*)^T B)_k & \forall j = 1, \ldots, n.
\end{align*}
\]

- For Alice, expected costs for rows in support should be **equal**, and **minimal** compared to rows not in support.
- For Bob, expected costs for columns in support should be **equal**, and **minimal** compared to columns not in support.
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

For Bob:

...
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

For Bob:
- Expected cost for Bob, given Alice’s strategy \(x\), on \(b_1\) and \(b_3\) are equal:
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

For Bob:

- Expected cost for Bob, given Alice’s strategy \(x\), on \(b_1\) and \(b_3\) are equal:
  \[
  2x_1 + 2x_2 = (x^T B)_1 = (x^T B)_3 = 2x_1 + 4x_2
  \]
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

For Bob:

- Expected cost for Bob, given Alice’s strategy \(x\), on \(b_1\) and \(b_3\) are equal:
  \[
  2x_1 + 2x_2 = (x^TB)_1 = (x^TB)_3 = 2x_1 + 4x_2
  \]
- Expected cost of \(b_1, b_3\) are minimal compared to that of \(b_2\):
Sketch of how to get linear program

Suppose for MNE $(x, y)$ we have $\text{Supp}(x) = \{a_1\}$, $\text{Supp}(y) = \{b_1, b_3\}$.

$$A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.$$  

For Bob:

- Expected cost for Bob, given Alice’s strategy $x$, on $b_1$ and $b_3$ are equal:
  $$2x_1 + 2x_2 = (x^T B)_1 = (x^T B)_3 = 2x_1 + 4x_2$$

- Expected cost of $b_1, b_3$ are minimal compared to that of $b_2$:
  $$2x_1 + 2x_2 = (x^T B)_1 (\text{or } 3) \leq (x^T B)_2 = 4x_1 + 0x_2.$$
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

For Bob:

- Expected cost for Bob, given Alice’s strategy \(x\), on \(b_1\) and \(b_3\) are equal:
  \[
  2x_1 + 2x_2 = (x^T B)_1 = (x^T B)_3 = 2x_1 + 4x_2
  \]

- Expected cost of \(b_1, b_3\) are minimal compared to that of \(b_2\):
  \[
  2x_1 + 2x_2 = (x^T B)_1 \ (\text{or } 3) \leq (x^T B)_2 = 4x_1 + 0x_2.
  \]

- Non-support columns have zero probability: \(y_2 = 0\).
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

For Bob:

- Expected cost for Bob, given Alice’s strategy \(x\), on \(b_1\) and \(b_3\) are equal:
  \[
  2x_1 + 2x_2 = (x^T B)_1 = (x^T B)_3 = 2x_1 + 4x_2
  \]
- Expected cost of \(b_1, b_3\) are minimal compared to that of \(b_2\):
  \[
  2x_1 + 2x_2 = (x^T B)_{(or\ 3)} \leq (x^T B)_2 = 4x_1 + 0x_2.
  \]
- Non-support columns have zero probability: \(y_2 = 0\).
- Support columns have positive probability: \(y_1, y_3 > 0\).
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}, \text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

For Bob:

- Expected cost for Bob, given Alice’s strategy \(x\), on \(b_1\) and \(b_3\) are equal:
  \[
  2x_1 + 2x_2 = (x^T B)_1 = (x^T B)_3 = 2x_1 + 4x_2
  \]

- Expected cost of \(b_1, b_3\) are minimal compared to that of \(b_2\):
  \[
  2x_1 + 2x_2 = (x^T B)_1 \ (\text{or } 3) \leq (x^T B)_2 = 4x_1 + 0x_2.
  \]

- Non-support columns have zero probability: \(y_2 = 0\).

- Support columns have positive probability: \(y_1, y_3 > 0\).

For Alice:
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

For Bob:
- Expected cost for Bob, given Alice’s strategy \(x\), on \(b_1\) and \(b_3\) are equal:
  \[
  2x_1 + 2x_2 = (x^T B)_1 = (x^T B)_3 = 2x_1 + 4x_2
  \]
- Expected cost of \(b_1, b_3\) are minimal compared to that of \(b_2\):
  \[
  2x_1 + 2x_2 = (x^T B)_1 \ \text{or}_3 \leq (x^T B)_2 = 4x_1 + 0x_2.
  \]
- Non-support columns have zero probability: \(y_2 = 0\).
- Support columns have positive probability: \(y_1, y_3 > 0\).

For Alice:
- For Alice, minimality of expected cost on \(a_1\) gives
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

For Bob:
- Expected cost for Bob, given Alice’s strategy \(x\), on \(b_1\) and \(b_3\) are equal:
  \[
  2x_1 + 2x_2 = (x^T B)_1 = (x^T B)_3 = 2x_1 + 4x_2
  \]
- Expected cost of \(b_1, b_3\) are minimal compared to that of \(b_2\):
  \[
  2x_1 + 2x_2 = (x^T B)_1 \text{ (or 3)} \leq (x^T B)_2 = 4x_1 + 0x_2.
  \]
- Non-support columns have zero probability: \(y_2 = 0\).
- Support columns have positive probability: \(y_1, y_3 > 0\).

For Alice:
- For Alice, minimality of expected cost on \(a_1\) gives
  \[
  2y_1 + y_2 + 2y_3 = (Ay)_1 \leq (Ay)_2 = 3y_1 + 3y_2 + 2y_3.
  \]
Sketch of how to get linear program

Suppose for MNE \((x, y)\) we have \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\).

\[
A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 3 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 4 & 2 \\ 2 & 0 & 4 \end{pmatrix}.
\]

For Bob:

- Expected cost for Bob, given Alice’s strategy \(x\), on \(b_1\) and \(b_3\) are equal:
  \[2x_1 + 2x_2 = (x^T B)_1 = (x^T B)_3 = 2x_1 + 4x_2\]

- Expected cost of \(b_1, b_3\) are minimal compared to that of \(b_2\):
  \[2x_1 + 2x_2 = (x^T B)_1 \ (\text{or} \ 3) \leq (x^T B)_2 = 4x_1 + 0x_2.\]

- Non-support columns have zero probability: \(y_2 = 0\).

- Support columns have positive probability: \(y_1, y_3 > 0\).

For Alice:

- For Alice, minimality of expected cost on \(a_1\) gives
  \[2y_1 + y_2 + 2y_3 = (Ay)_1 \leq (Ay)_2 = 3y_1 + 3y_2 + 2y_3.\]

- Similarly as for Bob, we get \(x_2 = 0\) and \(x_1 > 0\).
That is, \((x, y)\), with \(\text{Supp}(x) = \{ a_1 \}\), \(\text{Supp}(y) = \{ b_1, b_3 \}\), should satisfy
That is, \((x, y)\), with \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\), should satisfy

\[
\begin{align*}
2x_1 + 2x_2 & = 2x_1 + 4x_2 \\
2x_1 + 2x_2 & \leq 4x_1 + 0x_2 \\
2y_1 + y_2 + 2y_3 & \leq 3y_1 + 3y_2 + 2y_3 \\
x_1 + x_2 & = 1 \\
y_1 + y_2 + y_3 & = 1 \\
x_2 = y_2 = 0 \\
x_1, y_1, y_3 & > 0 \quad \text{(not linear constraint)}
\end{align*}
\]
That is, \((x, y)\), with \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\), should satisfy

\[
\begin{align*}
2x_1 + 2x_2 &= 2x_1 + 4x_2 \\
2x_1 + 2x_2 &\leq 4x_1 + 0x_2 \\
2y_1 + y_2 + 2y_3 &\leq 3y_1 + 3y_2 + 2y_3 \\
x_1 + x_2 &= 1 \\
y_1 + y_2 + y_3 &= 1 \\
x_2 &= y_2 = 0 \\
x_1, y_1, y_3 &> 0 \quad \text{(not linear constraint)}
\end{align*}
\]

To turn the last constraint into a linear one, we consider the program
That is, \((x, y)\), with \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\), should satisfy

\[
\begin{align*}
2x_1 + 2x_2 &= 2x_1 + 4x_2 \\
2x_1 + 2x_2 &\leq 4x_1 + 0x_2 \\
2y_1 + y_2 + 2y_3 &\leq 3y_1 + 3y_2 + 2y_3 \\
x_1 + x_2 &= 1 \\
y_1 + y_2 + y_3 &= 1 \\
x_2 = y_2 = 0 \\
x_1, y_1, y_3 &> 0 \quad \text{(not linear constraint)}
\end{align*}
\]

To turn the last constraint into a linear one, we consider the program

\[
\begin{align*}
\text{max} & \quad \delta \\
\text{subject to} & \quad 2x_1 + 2x_2 = 2x_1 + 4x_2 \\
& \quad 2x_1 + 2x_2 \leq 4x_1 + 0x_2 \\
& \quad 2y_1 + y_2 + 2y_3 \leq 3y_1 + 3y_2 + 2y_3 \\
& \quad x_2 = y_2 = 0, \quad x_1 + x_2 = y_1 + y_2 + y_3 = 1 \\
& \quad x_1 \geq \delta \\
& \quad y_1 \geq \delta \\
& \quad y_3 \geq \delta
\end{align*}
\]
That is, \((x, y)\), with \(\text{Supp}(x) = \{a_1\}\), \(\text{Supp}(y) = \{b_1, b_3\}\), should satisfy

\[
\begin{align*}
2x_1 + 2x_2 &= 2x_1 + 4x_2 \\
2x_1 + 2x_2 &\leq 4x_1 + 0x_2 \\
2y_1 + y_2 + 2y_3 &\leq 3y_1 + 3y_2 + 2y_3 \\
x_1 + x_2 &= 1 \\
y_1 + y_2 + y_3 &= 1 \\
x_2 = y_2 &= 0 \\
x_1, y_1, y_3 &> 0 \quad \text{(not linear constraint)}
\end{align*}
\]

To turn the last constraint into a linear one, we consider the program

\[
\begin{align*}
\max \quad & \delta \\
\text{subject to} \quad & 2x_1 + 2x_2 = 2x_1 + 4x_2 \\
& 2x_1 + 2x_2 \leq 4x_1 + 0x_2 \\
& 2y_1 + y_2 + 2y_3 \leq 3y_1 + 3y_2 + 2y_3 \\
& x_1 + x_2 = y_1 + y_2 + y_3 = 1 \\
& x_1 \geq \delta \\
& y_1 \geq \delta \\
& y_3 \geq \delta
\end{align*}
\]

\((A, B)\) has MNE with given supports iff LP returns feasible solution with \(\delta > 0\).
Computing MNE by support enumeration

Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$. 

Theorem: There is a polynomial time algorithm $A$ to decide if there exists an MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$.

An MNE will be computed by $A$ in polynomial time if the answer is YES. Algorithm $A$ consists of solving a linear program (given later).

Corollary (Support enumeration): There exists a $2^{n+m}\text{poly}(n, m, |A|, |B|)$ algorithm that computes an MNE of a two-player game $(A, B)$ with $A, B \in \mathbb{Q}^{m \times n}$.

Proof (of corollary): We have $2^m$ choices for $T_A$, and $2^n$ choices of $T_B$.

For fixed $(T_A, T_B)$, we can compute an MNE with those supports in polynomial time with $A$ (or decide that none exists). Nash's theorem guarantees that at least one MNE $(x^*, y^*)$ exists.

For $T_A = \text{Supp}(x^*)$ and $T_B = \text{Supp}(y^*)$, $A$ will return an MNE.
Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.

**Theorem**

There is a polynomial time algorithm $A$ to decide if there exists an MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$.

Proof (of corollary): We have $2^n$ choices for $T_A$, and $2^m$ choices of $T_B$. For fixed $(T_A, T_B)$, we can compute an MNE with those supports in polynomial time with $A$ (or decide that none exists). Nash’s theorem guarantees that at least one MNE $(x^*, y^*)$ exists.
Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.

**Theorem**

There is a polynomial time algorithm $A$ to decide if there exists an MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$. An MNE will be computed by $A$ in polynomial time in case the answer is YES.
Computing MNE by support enumeration

Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.

**Theorem**

There is a polynomial time algorithm $A$ to decide if there exists an MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$. An MNE will be computed by $A$ in polynomial time in case the answer is YES.

- Algorithm $A$ consists of solving linear program (given later).
Computing MNE by support enumeration

Let \( T_A \subseteq \{a_1, \ldots, a_m\} \) and \( T_B \subseteq \{b_1, \ldots, b_n\} \).

**Theorem**

There is a polynomial time algorithm \( A \) to decide if there exists an MNE \((x^*, y^*)\) with \( \text{Supp}(x^*) = T_A \) and \( \text{Supp}(y^*) = T_B \). An MNE will be computed by \( A \) in polynomial time in case the answer is YES.

- Algorithm \( A \) consists of solving linear program (given later).

**Corollary (Support enumeration)**

There exists an \( 2^{n+m} \text{poly}(n, m, |A|, |B|) \) algorithm that computes an MNE of a two-player game \((A, B)\) with \( A, B \in \mathbb{Q}^{m \times n} \).
Computing MNE by support enumeration

Let \( T_A \subseteq \{a_1, \ldots, a_m\} \) and \( T_B \subseteq \{b_1, \ldots, b_n\} \).

**Theorem**

There is a polynomial time algorithm \( A \) to decide if there exists an MNE \((x^*, y^*)\) with \( \text{Supp}(x^*) = T_A \) and \( \text{Supp}(y^*) = T_B \). An MNE will be computed by \( A \) in polynomial time in case the answer is YES.

- Algorithm \( A \) consists of solving linear program (given later).

**Corollary (Support enumeration)**

There exists an \( 2^{n+m} \text{poly}(n, m, |A|, |B|) \) algorithm that computes an MNE of a two-player game \((A, B)\) with \( A, B \in \mathbb{Q}^{m \times n} \).

**Proof (of corollary):**
Computing MNE by support enumeration

Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.

**Theorem**

There is a polynomial time algorithm $A$ to decide if there exists an MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$. An MNE will be computed by $A$ in polynomial time in case the answer is YES.

- Algorithm $A$ consists of solving linear program (given later).

**Corollary (Support enumeration)**

There exists an $2^{n+m}\text{poly}(n, m, |A|, |B|)$ algorithm that computes an MNE of a two-player game $(A, B)$ with $A, B \in \mathbb{Q}^{m \times n}$.

Proof (of corollary): We have $2^m$ choices for $T_A$, ...
Computing MNE by support enumeration

Let \( T_A \subseteq \{a_1, \ldots, a_m\} \) and \( T_B \subseteq \{b_1, \ldots, b_n\} \).

**Theorem**

There is a polynomial time algorithm \( \mathcal{A} \) to decide if there exists an MNE \((x^*, y^*)\) with \( \text{Supp}(x^*) = T_A \) and \( \text{Supp}(y^*) = T_B \). An MNE will be computed by \( \mathcal{A} \) in polynomial time in case the answer is YES.

- Algorithm \( \mathcal{A} \) consists of solving linear program (given later).

**Corollary (Support enumeration)**

There exists an \( 2^{n+m}\text{poly}(n, m, |A|, |B|) \) algorithm that computes an MNE of a two-player game \((A, B)\) with \( A, B \in \mathbb{Q}^{m \times n} \).

Proof (of corollary): We have \( 2^m \) choices for \( T_A \), and \( 2^n \) choices of \( T_B \).
Computing MNE by support enumeration

Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.

**Theorem**

There is a polynomial time algorithm $A$ to decide if there exists an MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$. An MNE will be computed by $A$ in polynomial time in case the answer is YES.

- Algorithm $A$ consists of solving linear program (given later).

**Corollary (Support enumeration)**

There exists an $2^{n+m}\text{poly}(n, m, |A|, |B|)$ algorithm that computes an MNE of a two-player game $(A, B)$ with $A, B \in \mathbb{Q}^{m \times n}$.

**Proof (of corollary):** We have $2^m$ choices for $T_A$, and $2^n$ choices of $T_B$.

- For fixed $(T_A, T_B)$, we can compute an MNE with those supports in polynomial time with $A$ (or decide that none exists).
Computing MNE by support enumeration

Let \( T_A \subseteq \{ a_1, \ldots, a_m \} \) and \( T_B \subseteq \{ b_1, \ldots, b_n \} \).

**Theorem**

There is a polynomial time algorithm \( A \) to decide if there exists an MNE \((x^*, y^*)\) with \( \text{Supp}(x^*) = T_A \) and \( \text{Supp}(y^*) = T_B \). An MNE will be computed by \( A \) in polynomial time in case the answer is YES.

- Algorithm \( A \) consists of solving linear program (given later).

**Corollary (Support enumeration)**

There exists an \( 2^{n+m} \text{poly}(n, m, |A|, |B|) \) algorithm that computes an MNE of a two-player game \((A, B)\) with \( A, B \in \mathbb{Q}^{m \times n} \).

**Proof (of corollary):** We have \( 2^m \) choices for \( T_A \), and \( 2^n \) choices of \( T_B \).

- For fixed \((T_A, T_B)\), we can compute an MNE with those supports in polynomial time with \( A \) (or decide that none exists).

Nash’s theorem guarantees that at least one MNE \((x^*, y^*)\) exists.
Computing MNE by support enumeration

Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$. 

**Theorem**

There is a polynomial time algorithm $A$ to decide if there exists an MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$. An MNE will be computed by $A$ in polynomial time in case the answer is YES.

- Algorithm $A$ consists of solving linear program (given later).

**Corollary (Support enumeration)**

There exists an $2^{n+m} \text{poly}(n, m, |A|, |B|)$ algorithm that computes an MNE of a two-player game $(A, B)$ with $A, B \in \mathbb{Q}^{m \times n}$.

**Proof (of corollary):** We have $2^m$ choices for $T_A$, and $2^n$ choices of $T_B$.

- For fixed $(T_A, T_B)$, we can compute an MNE with those supports in polynomial time with $A$ (or decide that none exists).

Nash's theorem guarantees that at least one MNE $(x^*, y^*)$ exists.

- For $T_A = \text{Supp}(x^*)$ and $T_B = \text{Supp}(y^*)$, $A$ will return an MNE.
Computing MNE by support enumeration

Let \( T_A \subseteq \{ a_1, \ldots, a_m \} \) and \( T_B \subseteq \{ b_1, \ldots, b_n \} \).

**Theorem**

There is a polynomial time algorithm \( \mathcal{A} \) to decide if there exists an MNE \((x^*, y^*)\) with \( \text{Supp}(x^*) = T_A \) and \( \text{Supp}(y^*) = T_B \). An MNE will be computed by \( \mathcal{A} \) in polynomial time in case the answer is YES.

- Algorithm \( \mathcal{A} \) consists of solving linear program (given later).

**Corollary (Support enumeration)**

There exists an \( 2^{n+m} \text{poly}(n, m, |A|, |B|) \) algorithm that computes an MNE of a two-player game \((A, B)\) with \( A, B \in \mathbb{Q}^{m \times n} \).

**Proof (of corollary):** We have \( 2^m \) choices for \( T_A \), and \( 2^n \) choices of \( T_B \).

- For fixed \((T_A, T_B)\), we can compute an MNE with those supports in polynomial time with \( \mathcal{A} \) (or decide that none exists).

Nash’s theorem guarantees that at least one MNE \((x^*, y^*)\) exists.

- For \( T_A = \text{Supp}(x^*) \) and \( T_B = \text{Supp}(y^*) \), \( \mathcal{A} \) will return an MNE.
The algorithm $\mathcal{A}$ (linear program)

Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$ be “candidate” supports.

\[
\begin{align*}
&\max \delta \\
\text{subject to} \quad & (Ay)_i = U \quad a_i \in T_A \\
& (x^T B)_j = V \quad b_j \in T_B \\
& x_i \geq \delta \quad a_i \in T_A \\
& y_j \geq \delta \quad b_j \in T_B \\
& (Ay)_i \geq U \quad a_i \in T_A \\
& (x^T B)_j \geq V \quad b_j \in T_B \\
& x_i = 0 \quad a_i \in T_A \\
& y_j = 0 \quad b_j \in T_B \\
& \sum_{i=1}^m x_i = 1 \\
& \sum_{j=1}^n y_j = 1 \\
& U, x_1, \ldots, x_m, \delta \in \mathbb{R} \\
& V, y_1, \ldots, y_n \in \mathbb{R}
\end{align*}
\]

Note that $(Ay)_i = \sum_j A_{ij} y_j$ and $(x^T B)_j = \sum_i x_i B_{ij}$.

Theorem

There exists an MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$ if and only if linear program above returns optimal solution with $\delta > 0$.

Exercise: Prove this theorem (using best response definition of MNE).
The algorithm $\mathcal{A}$ (linear program)

Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$ be “candidate” supports.

<table>
<thead>
<tr>
<th>max $\delta$</th>
<th>$(Ay)_i = U$ if $a_i \in T_A$</th>
<th>$(Ay)_i \geq U$ if $a_i \notin T_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject to</td>
<td>$x_i \geq \delta$ if $a_i \in T_A$</td>
<td>$x_i = 0$ if $a_i \notin T_A$</td>
</tr>
<tr>
<td></td>
<td>$\sum_{i=1}^m x_i = 1$</td>
<td>$\sum_{i=1}^m x_i = 1$</td>
</tr>
<tr>
<td></td>
<td>$U, x_1, \ldots, x_m, \delta \in \mathbb{R}$</td>
<td>$V, y_1, \ldots, y_n \in \mathbb{R}$</td>
</tr>
</tbody>
</table>

Note that

$(Ay)_i = \sum_{j=1}^n A_{ij} y_j$ and $(x^T B)_j = \sum_{i=1}^m x_i B_{ij}$.

Theorem

There exists an MNE $(x^\ast, y^\ast)$ with $\text{Supp}(x^\ast) = T_A$ and $\text{Supp}(y^\ast) = T_B$ if and only if linear program above returns optimal solution with $\delta > 0$.

Exercise: Prove this theorem (using best response definition of MNE).
The algorithm $\mathcal{A}$ (linear program)

Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$ be “candidate” supports.

\[
\begin{align*}
\text{max} & \quad \delta \\
\text{subject to} & \quad (Ay)_i = U & a_i \in T_A & (x^T B)_j = V & b_j \in T_B \\
& \quad x_i \geq \delta & a_i \in T_A & y_j \geq \delta & b_j \in T_B \\
& \quad (Ay)_i \geq U & a_i \notin T_A & (x^T B)_j \geq V & b_j \notin T_B \\
& \quad x_i = 0 & a_i \notin T_A & y_j = 0 & b_j \notin T_B \\
& \quad \sum_{i=1}^{m} x_i = 1 & U, x_1, \ldots, x_m, \delta \in \mathbb{R} & \sum_{j=1}^{n} y_j = 1 & V, y_1, \ldots, y_n \in \mathbb{R}
\end{align*}
\]

Note that $(Ay)_i = \sum_j A_{ij} y_j$ and $(x^T B)_j = \sum_i x_i B_{ij}$.
The algorithm \( \mathcal{A} \) (linear program)

Let \( T_A \subseteq \{a_1, \ldots, a_m\} \) and \( T_B \subseteq \{b_1, \ldots, b_n\} \) be “candidate” supports.

\[
\begin{align*}
\text{max} & \quad \delta \\
\text{subject to} & \quad (Ay)_i = U, \quad a_i \in T_A \\
& \quad x_i \geq \delta, \quad a_i \in T_A \\
& \quad (Ay)_i \geq U, \quad a_i \notin T_A \\
& \quad x_i = 0, \quad a_i \notin T_A \\
& \quad \sum_{i=1}^{m} x_i = 1 \\
& \quad U, x_1, \ldots, x_m, \delta \in \mathbb{R}
\end{align*}
\]

\[
\begin{align*}
& \quad (x^T B)_j = V, \quad b_j \in T_B \\
& \quad y_j \geq \delta, \quad b_j \in T_B \\
& \quad (x^T B)_j \geq V, \quad b_j \notin T_B \\
& \quad y_j = 0, \quad b_j \notin T_B \\
& \quad \sum_{j=1}^{n} y_j = 1 \\
& \quad V, y_1, \ldots, y_n \in \mathbb{R}
\end{align*}
\]

Note that \((Ay)_i = \sum_j A_{ij} y_j\) and \((x^T B)_j = \sum_i x_i B_{ij}\).

**Theorem**

There exists an MNE \((x^*, y^*)\) with \(\text{Supp}(x^*) = T_A\) and \(\text{Supp}(y^*) = T_B\) if and only if linear program above returns optimal solution with \(\delta > 0\).
The algorithm $\mathcal{A}$ (linear program)

Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$ be “candidate” supports.

\[
\begin{align*}
\text{max} & \quad \delta \\
\text{subject to} & \quad (Ay)_i = U \quad a_i \in T_A \\
& \quad x_i \geq \delta \quad a_i \in T_A \\
& \quad (Ay)_i \geq U \quad a_i \notin T_A \\
& \quad x_i = 0 \quad a_i \notin T_A \\
& \quad \sum_{i=1}^m x_i = 1 \\
& \quad U, x_1, \ldots, x_m, \delta \in \mathbb{R}
\end{align*}
\]

\[
\begin{align*}
& \quad (x^TB)_j = V \quad b_j \in T_B \\
& \quad y_j \geq \delta \quad b_j \in T_B \\
& \quad (x^TB)_j \geq V \quad b_j \notin T_B \\
& \quad y_j = 0 \quad b_j \notin T_B \\
& \quad \sum_{j=1}^n y_j = 1 \\
& \quad V, y_1, \ldots, y_n \in \mathbb{R}
\end{align*}
\]

Note that $(Ay)_i = \sum_j A_{ij}y_j$ and $(x^TB)_j = \sum_i x_iB_{ij}$.

**Theorem**

There exists an MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$ if and only if linear program above returns optimal solution with $\delta > 0$.

Exercise: Prove this theorem (using best response definition of MNE).
Computing MNE with sparse supports

$\text{MNE}(x^*, y^*)$ is $k$-sparse if $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

Players assign positive probability to at most $k$ strategies.

Game $(A, B)$ is called $k$-sparse if it has $k$-sparse MNE.

Theorem (Computation of sparse MNE)

There exists $\sum_{k} q^k \leq m^k + 1$ choices of support of Alice that are $k$-sparse, and $\sum_{k} q^k \leq n^k + 1$ for Bob. Remainder is similar to proof of corollary on Slide 10.

Remark

There exist games with unique MNE $(x^*, y^*)$ having $|\text{Supp}(x^*)| = m$ and $|\text{Supp}(y^*)| = n$.

Theorem useful for computation of approximate Nash equilibrium.
Computing MNE with sparse supports

MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).
MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).

- Players assign positive probability to at most \(k\) strategies.
MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).

- Players assign positive probability to at most \(k\) strategies.
- Game \((A, B)\) is called \(k\)-sparse if it has \(k\)-sparse MNE.
MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).

- Players assign positive probability to at most \(k\) strategies.
- Game \((A, B)\) is called \(k\)-sparse if it has \(k\)-sparse MNE.

**Theorem (Computation of sparse MNE)**

There exists \((nm)^k \text{poly}(n, m, |A|, |B|)\)-time algorithm to decide whether \(k\)-sparse MNE exists.
Computing MNE with sparse supports

MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).
- Players assign positive probability to at most \(k\) strategies.
- Game \((A, B)\) is called \(k\)-sparse if it has \(k\)-sparse MNE.

**Theorem (Computation of sparse MNE)**

There exists \((nm)^k\text{poly}(n, m, |A|, |B|)\)-time algorithm to decide whether \(k\)-sparse MNE exists (and that outputs one if answer is YES)
Computing MNE with sparse supports

MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).

- Players assign positive probability to at most \(k\) strategies.
- Game \((A, B)\) is called \(k\)-sparse if it has \(k\)-sparse MNE.

**Theorem (Computation of sparse MNE)**

There exists \((nm)^k \text{poly}(n, m, |A|, |B|)\)-time algorithm to decide whether \(k\)-sparse MNE exists (and that outputs one if answer is YES) in games \((A, B)\) with \(A, B \in \mathbb{Q}^{m \times n}\).
Computing MNE with sparse supports

MNE \((x^*, y^*)\) is **k-sparse** if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).

- Players assign positive probability to at most \(k\) strategies.
- Game \((A, B)\) is called \(k\)-sparse if it has \(k\)-sparse MNE.

**Theorem (Computation of sparse MNE)**

There exists \((nm)^k \text{poly}(n, m, |A|, |B|)\)-time algorithm to decide whether \(k\)-sparse MNE exists (and that outputs one if answer is YES) in games \((A, B)\) with \(A, B \in \mathbb{Q}^{m \times n}\).

**Proof:** There are \(\sum_{q=1}^{k} \binom{m}{q} \leq m^{k+1}\) choices of support of Alice that are \(k\)-sparse,
MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).

- Players assign positive probability to at most \(k\) strategies.
- Game \((A, B)\) is called \(k\)-sparse if it has \(k\)-sparse MNE.

**Theorem (Computation of sparse MNE)**

There exists \((nm)^k \text{poly}(n, m, |A|, |B|)\)-time algorithm to decide whether \(k\)-sparse MNE exists (and that outputs one if answer is YES) in games \((A, B)\) with \(A, B \in \mathbb{Q}^{m \times n}\).

**Proof:** There are \(\sum_{q=1}^{k} \binom{m}{q} \leq m^{k+1}\) choices of support of Alice that are \(k\)-sparse, and \(\sum_{q=1}^{k} \binom{n}{q} \leq n^{k+1}\) for Bob.
Computing MNE with sparse supports

MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).
- Players assign positive probability to at most \(k\) strategies.
- Game \((A, B)\) is called \(k\)-sparse if it has \(k\)-sparse MNE.

**Theorem (Computation of sparse MNE)**

There exists \((nm)^k \text{poly}(n, m, |A|, |B|)\)-time algorithm to decide whether \(k\)-sparse MNE exists (and that outputs one if answer is YES) in games \((A, B)\) with \(A, B \in \mathbb{Q}^{m \times n}\).

**Proof:** There are \(\sum_{q=1}^{k} \binom{m}{q} \leq m^{k+1}\) choices of support of Alice that are \(k\)-sparse, and \(\sum_{q=1}^{k} \binom{n}{q} \leq n^{k+1}\) for Bob. Remainder is similar to proof of corollary on Slide 10.
Computing MNE with sparse supports

MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).

- Players assign positive probability to at most \(k\) strategies.
- Game \((A, B)\) is called \(k\)-sparse if it has \(k\)-sparse MNE.

**Theorem (Computation of sparse MNE)**

There exists \((nm)^k\ poly(n, m, |A|, |B|)\)-time algorithm to decide whether \(k\)-sparse MNE exists (and that outputs one if answer is YES) in games \((A, B)\) with \(A, B \in \mathbb{Q}^{m \times n}\).

**Proof:** There are \(\sum_{q=1}^{k} \binom{m}{q} \leq m^{k+1}\) choices of support of Alice that are \(k\)-sparse, and \(\sum_{q=1}^{k} \binom{n}{q} \leq n^{k+1}\) for Bob. Remainder is similar to proof of corollary on Slide 10. 

\(\square\)
Computing MNE with sparse supports

MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).

- Players assign positive probability to at most \(k\) strategies.
- Game \((A, B)\) is called \(k\)-sparse if it has \(k\)-sparse MNE.

**Theorem (Computation of sparse MNE)**

There exists \((nm)^k \text{poly}(n, m, |A|, |B|)\)-time algorithm to decide whether \(k\)-sparse MNE exists (and that outputs one if answer is YES) in games \((A, B)\) with \(A, B \in \mathbb{Q}^{m \times n}\).

**Proof:** There are \(\sum_{q=1}^{k} \binom{m}{q} \leq m^{k+1}\) choices of support of Alice that are \(k\)-sparse, and \(\sum_{q=1}^{k} \binom{n}{q} \leq n^{k+1}\) for Bob. Remainder is similar to proof of corollary on Slide 10.

**Remark**

There exist games with unique MNE \((x^*, y^*)\) having \(|\text{Supp}(x^*)| = m\) and \(|\text{Supp}(y^*)| = n\).
Computing MNE with sparse supports

MNE \((x^*, y^*)\) is \(k\)-sparse if \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).
- Players assign positive probability to at most \(k\) strategies.
- Game \((A, B)\) is called \(k\)-sparse if it has \(k\)-sparse MNE.

**Theorem (Computation of sparse MNE)**

There exists \((nm)^k \text{poly}(n, m, |A|, |B|)\)-time algorithm to decide whether \(k\)-sparse MNE exists (and that outputs one if answer is YES) in games \((A, B)\) with \(A, B \in \mathbb{Q}^{m \times n}\).

**Proof:** There are \(\sum_{q=1}^{k} \binom{m}{q} \leq m^{k+1}\) choices of support of Alice that are \(k\)-sparse, and \(\sum_{q=1}^{k} \binom{n}{q} \leq n^{k+1}\) for Bob. Remainder is similar to proof of corollary on Slide 10.

**Remark**

There exist games with unique MNE \((x^*, y^*)\) having \(|\text{Supp}(x^*)| = m\) and \(|\text{Supp}(y^*)| = n\).

*Theorem useful for computation of approximate Nash equilibrium.*
Computation of approximate MNE
Consider two-player game \((A, B)\) played by Alice and Bob.

**Definition (Approximate MNE, pure strategy formulation)**

For \(\epsilon > 0\), mixed strategies \((x^*, y^*)\) form \(\epsilon\)-MNE iff

\[
(x^*)^T Ay^* \leq (e_i^*)^T Ay^* + \epsilon \quad i = 1, \ldots, m,
\]

\[
(x^*)^T By^* \leq (x^*)^T Be_j + \epsilon \quad j = 1, \ldots, n.
\]

That is, players both have no improving move to pure strategy.

Captures idea that mixed strategies are "almost" an equilibrium.

Example: \(x = (1, 0)\), \(y = (1, 0)\) is a \(0.1\)-approximate equilibrium for game

\[
A = \begin{pmatrix} 1 & 1 \\ 0 & .9 & 2 \end{pmatrix}
\]

and

\[
B = \begin{pmatrix} 1 & 2 \\ 2 & 2 \end{pmatrix}.
\]
Consider two-player game \((A, B)\) played by Alice and Bob.

- For \(x \in \Delta_A\) and \(y \in \Delta_B\), (expected) cost given by
  \[
  C_{\text{Alice}}(x, y) = x^T A y, \quad C_{\text{Bob}}(x, y) = x^T B y.
  \]
Consider two-player game \((A, B)\) played by Alice and Bob.

- For \(x \in \Delta_A\) and \(y \in \Delta_B\), (expected) cost given by
  \[C_{\text{Alice}}(x, y) = x^T Ay, \quad C_{\text{Bob}}(x, y) = x^T By.\]

**Definition (Approximate MNE, pure strategy formulation)**

For \(\epsilon > 0\), mixed strategies \((x^*, y^*)\) form \(\epsilon\)-MNE iff

\[
(x^*)^T Ay^* \leq (e^i)^T Ay^* + \epsilon \quad i = 1, \ldots, m,
\]

\[
(x^*)^T By^* \leq (x^*)^T Be^j + \epsilon \quad j = 1, \ldots, n.
\]

That is, players both have no improving move to pure strategy.
Consider two-player game \((A, B)\) played by Alice and Bob.

For \(x \in \Delta_A\) and \(y \in \Delta_B\), (expected) cost given by
\[
C_{\text{Alice}}(x, y) = x^T Ay, \quad C_{\text{Bob}}(x, y) = x^T By.
\]

**Definition (Approximate MNE, pure strategy formulation)**

For \(\epsilon > 0\), mixed strategies \((x^*, y^*)\) form \(\epsilon\)-MNE iff
\[
(x^*)^T Ay^* \leq (e_i)^T Ay^* + \epsilon \quad i = 1, \ldots, m,
(x^*)^T By^* \leq (x^*)^T Be_j + \epsilon \quad j = 1, \ldots, n.
\]

That is, players both have no improving move to pure strategy.

Captures idea that mixed strategies are “almost” an equilibrium.
Consider two-player game \((A, B)\) played by Alice and Bob.

- For \(x \in \Delta_A\) and \(y \in \Delta_B\), (expected) cost given by
  \[
  C_{\text{Alice}}(x, y) = x^T Ay, \quad C_{\text{Bob}}(x, y) = x^T By.
  \]

Definition (Approximate MNE, pure strategy formulation)

For \(\epsilon > 0\), mixed strategies \((x^*, y^*)\) form \(\epsilon\-MNE\) iff

\[
(x^*)^T Ay^* \leq (e^i)^T Ay^* + \epsilon \quad i = 1, \ldots, m,
\]

\[
(x^*)^T By^* \leq (x^*)^T Be^j + \epsilon \quad j = 1, \ldots, n.
\]

That is, players both have no improving move to pure strategy.

Captures idea that mixed strategies are “almost” an equilibrium.

- Players might be able to improve cost, but at most by term \(\epsilon\).
Approximate equilibrium

Consider two-player game \((A, B)\) played by Alice and Bob.

- For \(x \in \Delta_A\) and \(y \in \Delta_B\), (expected) cost given by
  \[ C_{\text{Alice}}(x, y) = x^T Ay, \quad C_{\text{Bob}}(x, y) = x^T By. \]

**Definition (Approximate MNE, pure strategy formulation)**

For \(\epsilon > 0\), mixed strategies \((x^*, y^*)\) form \(\epsilon\)-MNE iff

\[
(x^*)^T A y^* \leq (e^i)^T A y^* + \epsilon \quad i = 1, \ldots, m, \\
(x^*)^T B y^* \leq (x^*)^T B e^j + \epsilon \quad j = 1, \ldots, n.
\]

That is, players both have no improving move to pure strategy.

Captures idea that mixed strategies are “almost” an equilibrium.

- Players might be able to improve cost, but at most by term \(\epsilon\).

**Example**

\(x = (1, 0), y = (1, 0)\) is 0.1-approximate equilibrium for game

\[
A = \begin{pmatrix} 1 & 1 \\ 0.9 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 & 2 \\ 2 & 2 \end{pmatrix}.
\]
Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.
Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.

**Theorem**

*There is a polynomial time algorithm $A$ to decide if there exists an $\epsilon$-approximate MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$.***
Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.

**Theorem**

There is a polynomial time algorithm $A$ to decide if there exists an $\epsilon$-approximate MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$. An $\epsilon$-approximate MNE will be computed by $A$ in polynomial time in case the answer is YES.
Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.

**Theorem**

There is a polynomial time algorithm $A$ to decide if there exists an $\epsilon$-approximate MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$. An $\epsilon$-approximate MNE will be computed by $A$ in polynomial time in case the answer is YES.

- Modify the linear program from the case $\epsilon = 0$ on Slide 11.
Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.

**Theorem**

There is a polynomial time algorithm $A$ to decide if there exists an $\epsilon$-approximate MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$. An $\epsilon$-approximate MNE will be computed by $A$ in polynomial time in case the answer is YES.

- Modify the linear program from the case $\epsilon = 0$ on Slide 11.

\[
\begin{align*}
\text{max} & \quad \delta \\
\text{subject to} & \quad (Ay)_i \leq U + \epsilon & a_i \in T_A & (x^T B)_j \leq V + \epsilon & b_j \in T_B \\
& \quad x_i \geq \delta & a_i \in T_A & y_j \geq \delta & b_j \in T_B \\
& \quad (Ay)_i \geq U & a_i \notin T_A & (x^T B)_j \geq V & b_j \notin T_B \\
& \quad x_i = 0 & a_i \notin T_A & y_j = 0 & b_j \notin T_B \\
& \quad \sum_{i=1}^m x_i = 1 & & \sum_{j=1}^n y_j = 1 \\
& \quad U, x_1, \ldots, x_m, \delta \in \mathbb{R} & & V, y_1, \ldots, y_n \in \mathbb{R}
\end{align*}
\]
Let $T_A \subseteq \{a_1, \ldots, a_m\}$ and $T_B \subseteq \{b_1, \ldots, b_n\}$.

**Theorem**

There is a polynomial time algorithm $A$ to decide if there exists an $\epsilon$-approximate MNE $(x^*, y^*)$ with $\text{Supp}(x^*) = T_A$ and $\text{Supp}(y^*) = T_B$. An $\epsilon$-approximate MNE will be computed by $A$ in polynomial time in case the answer is YES.

- Modify the linear program from the case $\epsilon = 0$ on Slide 11.

\[
\begin{align*}
\max \quad & \delta \\
\text{subject to} \quad & (Ay)_i \leq U + \epsilon \quad a_i \in T_A \\
& x_i \geq \delta \quad a_i \in T_A \\
& (Ay)_i \geq U \quad a_i \notin T_A \\
& x_i = 0 \quad a_i \notin T_A \\
& \sum_{i=1}^m x_i = 1 \\
& U, x_1, \ldots, x_m, \delta \in \mathbb{R} \\
& (x^T B)_j \leq V + \epsilon \quad b_j \in T_B \\
& y_j \geq \delta \quad b_j \in T_B \\
& (x^T B)_j \geq V \quad b_j \notin T_B \\
& y_j = 0 \quad b_j \notin T_B \\
& \sum_{j=1}^n y_j = 1 \\
& V, y_1, \ldots, y_n \in \mathbb{R}
\end{align*}
\]

- “Support enumeration” corollary on Slide 10 also holds for $\epsilon$-MNE.
Small support approximate equilibria

An $\epsilon$-MNE $(x^*, y^*)$ is $k$-sparse if $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$. 
Small support approximate equilibria

An $\epsilon$-MNE $(x^*, y^*)$ is $k$-sparse if $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Same as for MNE (since definition does not involve $\epsilon$).
Small support approximate equilibria

An $\epsilon$-MNE $(x^*, y^*)$ is $k$-sparse if $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Same as for MNE (since definition does not involve $\epsilon$).

**Theorem (Computation of sparse approximate MNE)**

Suppose game $(A, B)$, with $A, B \in \mathbb{Q}^{m \times n}$, has $k$-sparse $\epsilon$-MNE. Then there is an $(nm)^k \text{poly}(n, m, |A|, |B|)$-time algorithm to compute one.
Small support approximate equilibria

An $\epsilon$-MNE $(x^*, y^*)$ is $k$-sparse if $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Same as for MNE (since definition does not involve $\epsilon$).

**Theorem (Computation of sparse approximate MNE)**

Suppose game $(A, B)$, with $A, B \in \mathbb{Q}^{m \times n}$, has $k$-sparse $\epsilon$-MNE. Then there is an $(nm)^k \text{poly}(n, m, |A|, |B|)$-time algorithm to compute one.

**Lemma (Lipton, Markakis and Mehta (LMM), 2003)**

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with $|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2)$ and $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$. 
Small support approximate equilibria

An $\epsilon$-MNE $(x^*, y^*)$ is $k$-sparse if $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Same as for MNE (since definition does not involve $\epsilon$).

**Theorem (Computation of sparse approximate MNE)**

Suppose game $(A, B)$, with $A, B \in \mathbb{Q}^{m \times n}$, has $k$-sparse $\epsilon$-MNE. Then there is an $(nm)^k \text{poly}(n, m, |A|, |B|)$-time algorithm to compute one.

**Lemma (Lipton, Markakis and Mehta (LMM), 2003)**

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$$|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2).$$

**Corollary**

There exists $(nm)^{O(\log(\max\{m,n\})/\epsilon^2)} \text{poly}(n, m, |A|, |B|)$ time algorithm for computing $\epsilon$-MNE in game $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$. 

16 / 28
Small support approximate equilibria

An $\epsilon$-MNE $(x^*, y^*)$ is $k$-sparse if $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Same as for MNE (since definition does not involve $\epsilon$).

**Theorem (Computation of sparse approximate MNE)**

Suppose game $(A, B)$, with $A, B \in \mathbb{Q}^{m \times n}$, has $k$-sparse $\epsilon$-MNE. Then there is an $(nm)^k \text{poly}(n, m, |A|, |B|)$-time algorithm to compute one.

**Lemma (Lipton, Markakis and Mehta (LMM), 2003)**

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$$|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2).$$

**Corollary**

There exists $(nm)^O(\log(\max\{m, n\})/\epsilon^2) \text{poly}(n, m, |A|, |B|)$ time algorithm for computing $\epsilon$-MNE in game $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$.

- Assuming $m \geq n$, running time reduces to $m^O(\log(m)/\epsilon^2) \text{poly}(n, m, |A|, |B|)$. 

---

For constant $\epsilon > 0$, $m^O(\log(m))$ dependence is much better than $2^O(m)$. 

---
Small support approximate equilibria

An $\epsilon$-MNE $(x^*, y^*)$ is $k$-sparse if $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Same as for MNE (since definition does not involve $\epsilon$).

**Theorem (Computation of sparse approximate MNE)**

Suppose game $(A, B)$, with $A, B \in \mathbb{Q}^{m \times n}$, has $k$-sparse $\epsilon$-MNE. Then there is an $(nm)^k \text{poly}(n, m, |A|, |B|)$-time algorithm to compute one.

**Lemma (Lipton, Markakis and Mehta (LMM), 2003)**

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^{\epsilon}, y^{\epsilon})$ with

$$|\text{Supp}(x^{\epsilon})| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^{\epsilon})| = O(\log(m)/\epsilon^2).$$

**Corollary**

There exists $(nm)^{O(\log(\max\{m,n\})/\epsilon^2)} \text{poly}(n, m, |A|, |B|)$ time algorithm for computing $\epsilon$-MNE in game $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$.

- Assuming $m \geq n$, running time reduces to $m^{O(\log(m)/\epsilon^2)} \text{poly}(n, m, |A|, |B|)$.
- For constant $\epsilon > 0$, $m^{O(\log(m))}$ dependence is much better than $2^{O(m)}$. 
Computation of approximate MNE

Proof of LMM lemma
Recap (computation of approximate MNE)

Suppose there is an \( \epsilon \)-MNE \((x^*, y^*)\) with \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).
Suppose there is an $\epsilon$-MNE $(x^*, y^*)$ with $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Enumerate over all $(nm)^k$ possible supports $(T_A, T_B)$.
Recap (computation of approximate MNE)

Suppose there is an $\epsilon$-MNE $(x^*, y^*)$ with $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Enumerate over all $(nm)^k$ possible supports $(T_A, T_B)$.
- Solve linear program for every fixed $(T_A, T_B)$.

For exact MNE ($\epsilon = 0$), there is no non-trivial bound known for $k$.

There exist games for which $k$ is as large as $m$ (or $n$) for all MNE.

For $\epsilon$-MNE, with $\epsilon$ constant, there does exist a non-trivial bound on $k$.

Lemma (Lipton, Markakis and Mehta, 2003):

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x_\epsilon, y_\epsilon)$ with $|\text{Supp}(x_\epsilon)| = O(\log(n)/\epsilon^2)$ and $|\text{Supp}(y_\epsilon)| = O(\log(m)/\epsilon^2)$.

The normalization of $A$ and $B$ is not without loss of generality! Just like Nash's theorem, proof is non-constructive!
Recap (computation of approximate MNE)

Suppose there is an $\epsilon$-MNE $(x^*, y^*)$ with $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.
- Enumerate over all $(nm)^k$ possible supports $(T_A, T_B)$.
- Solve linear program for every fixed $(T_A, T_B)$.

For exact MNE ($\epsilon = 0$), there is no non-trivial bound known for $k$.
There exist games for which $k$ is as large as $m$ (or $n$) for all MNE.
For $\epsilon$-MNE, with $\epsilon$ constant, there does exist a non-trivial bound on $k$.

Lemma (Lipton, Markakis and Mehta, 2003)
For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x_\epsilon, y_\epsilon)$ with $|\text{Supp}(x_\epsilon)| = O(\log(n)/\epsilon^2)$ and $|\text{Supp}(y_\epsilon)| = O(\log(m)/\epsilon^2)$.

The normalization of $A$ and $B$ is not without loss of generality!

Just like Nash's theorem, proof is non-constructive!
Recap (computation of approximate MNE)

Suppose there is an $\epsilon$-MNE $(x^*, y^*)$ with $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Enumerate over all $(nm)^k$ possible supports $(T_A, T_B)$.
  - Solve linear program for every fixed $(T_A, T_B)$.

For exact MNE ($\epsilon = 0$), there is no non-trivial bound known for $k$. 

The normalization of $A$ and $B$ is not without loss of generality! 

Just like Nash's theorem, proof is non-constructive!
Recap (computation of approximate MNE)

Suppose there is an \( \epsilon \)-MNE \((x^*, y^*)\) with \(|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k\).

- Enumerate over all \((nm)^k\) possible supports \((T_A, T_B)\).
  - Solve linear program for every fixed \((T_A, T_B)\).

For exact MNE \((\epsilon = 0)\), there is no non-trivial bound known for \(k\).

- There exist games for which \(k\) is as large as \(m\) (or \(n\)) for all MNE.
Recap (computation of approximate MNE)

Suppose there is an $\epsilon$-MNE $(x^*, y^*)$ with $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Enumerate over all $(nm)^k$ possible supports $(T_A, T_B)$.
- Solve linear program for every fixed $(T_A, T_B)$.

For exact MNE ($\epsilon = 0$), there is no non-trivial bound known for $k$.

- There exist games for which $k$ is as large as $m$ (or $n$) for all MNE.

For $\epsilon$-MNE, with $\epsilon$ constant, there does exist a non-trivial bound on $k$. 
Recap (computation of approximate MNE)

Suppose there is an $\epsilon$-MNE $(x^*, y^*)$ with $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Enumerate over all $(nm)^k$ possible supports $(T_A, T_B)$.
- Solve linear program for every fixed $(T_A, T_B)$.

For exact MNE ($\epsilon = 0$), there is no non-trivial bound known for $k$.

- There exist games for which $k$ is as large as $m$ (or $n$) for all MNE.

For $\epsilon$-MNE, with $\epsilon$ constant, there does exist a non-trivial bound on $k$.

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with $|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2)$ and $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$. 
Recap (computation of approximate MNE)

Suppose there is an $\epsilon$-MNE $(x^*, y^*)$ with $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.

- Enumerate over all $(nm)^k$ possible supports $(T_A, T_B)$.
- Solve linear program for every fixed $(T_A, T_B)$.

For exact MNE ($\epsilon = 0$), there is no non-trivial bound known for $k$.

- There exist games for which $k$ is as large as $m$ (or $n$) for all MNE.

For $\epsilon$-MNE, with $\epsilon$ constant, there does exist a non-trivial bound on $k$.

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$$|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2).$$

- The normalization of $A$ and $B$ is not without loss of generality!
Recap (computation of approximate MNE)

Suppose there is an $\epsilon$-MNE $(x^*, y^*)$ with $|\text{Supp}(x^*)|, |\text{Supp}(y^*)| \leq k$.
- Enumerate over all $(nm)^k$ possible supports $(T_A, T_B)$.
- Solve linear program for every fixed $(T_A, T_B)$.

For exact MNE ($\epsilon = 0$), there is no non-trivial bound known for $k$.
- There exist games for which $k$ is as large as $m$ (or $n$) for all MNE.

For $\epsilon$-MNE, with $\epsilon$ constant, there does exist a non-trivial bound on $k$.

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with $|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2)$ and $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$.

- The normalization of $A$ and $B$ is not without loss of generality!
- Just like Nash’s theorem, proof is non-constructive!
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)

For any \( \epsilon > 0 \), \((A, B) \) with \( A, B \in [-1, 1]^{m \times n} \) has \( \epsilon \)-MNE \((x^\epsilon, y^\epsilon)\) with

\[
|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2).
\]
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with 

$$|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2).$$

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$. 
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$$|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2).$$

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$.

- Always exists at least one because of Nash’s theorem.
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$$|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2).$$

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$.

- Always exists at least one because of Nash’s theorem.

High-level idea
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2)$ and $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$.

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$.

- Always exists at least one because of Nash’s theorem.

High-level idea

- First, replace $y$ by mixed strategy $y^\epsilon$ with properties:
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2)$ \quad and \quad $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$.

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$.

- Always exists at least one because of Nash’s theorem.

High-level idea

- First, replace $y$ by mixed strategy $y^\epsilon$ with properties:
  - $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$,
Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$$|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2) \text{ and } |\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2).$$

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$.

- Always exists at least one because of Nash’s theorem.

High-level idea

- First, replace $y$ by mixed strategy $y^\epsilon$ with properties:
  - $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$,
  - $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-approximate MNE.
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2)$ and $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$.

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$.

- Always exists at least one because of Nash’s theorem.

High-level idea

- First, replace $y$ by mixed strategy $y^\epsilon$ with properties:
  - $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$,
  - $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-approximate MNE.
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with
\[ |\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2). \]

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$.
- Always exists at least one because of Nash’s theorem.

High-level idea
- First, replace $y$ by mixed strategy $y^\epsilon$ with properties:
  - $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$,
  - $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-approximate MNE.
- Then, replace $x$ by mixed strategy $x^\epsilon$ with properties:
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)

For any \( \epsilon > 0 \), \((A, B)\) with \( A, B \in [-1, 1]^{m \times n} \) has \( \epsilon \)-MNE \((x^\epsilon, y^\epsilon)\) with \(|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2)\) and \(|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)\).

Proof: We start with an exact \((\epsilon = 0)\) mixed Nash equilibrium \((x, y)\).

- Always exists at least one because of Nash’s theorem.

High-level idea

- First, replace \( y \) by mixed strategy \( y^\epsilon \) with properties:
  - \(|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)\),
  - \((x, y^\epsilon)\) is \( \frac{\epsilon}{2} \)-approximate MNE.

- Then, replace \( x \) by mixed strategy \( x^\epsilon \) with properties:
  - \(|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2)\),
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)
For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$|\text{Supp}(x^\epsilon)| = O\left(\frac{\log(n)}{\epsilon^2}\right)$ and $|\text{Supp}(y^\epsilon)| = O\left(\frac{\log(m)}{\epsilon^2}\right)$.

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$.
- Always exists at least one because of Nash’s theorem.

High-level idea
- First, replace $y$ by mixed strategy $y^\epsilon$ with properties:
  - $|\text{Supp}(y^\epsilon)| = O\left(\frac{\log(m)}{\epsilon^2}\right)$,
  - $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-approximate MNE.
- Then, replace $x$ by mixed strategy $x^\epsilon$ with properties:
  - $|\text{Supp}(x^\epsilon)| = O\left(\frac{\log(n)}{\epsilon^2}\right)$,
  - $(x^\epsilon, y^\epsilon)$ is $\epsilon$-approximate MNE.
Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^{\epsilon}, y^{\epsilon})$ with

$$|\text{Supp}(x^{\epsilon})| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^{\epsilon})| = O(\log(m)/\epsilon^2).$$

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$.

- Always exists at least one because of Nash’s theorem.

High-level idea

- First, replace $y$ by mixed strategy $y^{\epsilon}$ with properties:
  - $|\text{Supp}(y^{\epsilon})| = O(\log(m)/\epsilon^2)$,
  - $(x, y^{\epsilon})$ is $\frac{\epsilon}{2}$-approximate MNE.

- Then, replace $x$ by mixed strategy $x^{\epsilon}$ with properties:
  - $|\text{Supp}(x^{\epsilon})| = O(\log(n)/\epsilon^2)$,
  - $(x^{\epsilon}, y^{\epsilon})$ is $\epsilon$-approximate MNE.

Both sparsification steps can be proved in a similar way.
Proof of LMM lemma

Lemma (Lipton, Markakis and Mehta, 2003)

For any $\epsilon > 0$, $(A, B)$ with $A, B \in [-1, 1]^{m \times n}$ has $\epsilon$-MNE $(x^\epsilon, y^\epsilon)$ with

$$|	ext{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2) \quad \text{and} \quad |\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2).$$

Proof: We start with an exact ($\epsilon = 0$) mixed Nash equilibrium $(x, y)$.

- Always exists at least one because of Nash’s theorem.

High-level idea

- First, replace $y$ by mixed strategy $y^\epsilon$ with properties:
  - $|\text{Supp}(y^\epsilon)| = O(\log(m)/\epsilon^2)$,
  - $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-approximate MNE.

- Then, replace $x$ by mixed strategy $x^\epsilon$ with properties:
  - $|\text{Supp}(x^\epsilon)| = O(\log(n)/\epsilon^2)$,
  - $(x^\epsilon, y^\epsilon)$ is $\epsilon$-approximate MNE.

Both sparsification steps can be proved in a similar way

(Of course, one may also first sparsify $x$, and then $y$.)
Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE?
Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE?

(Note that mixed strategy $x$ is fixed throughout sparsification of $y$.)
Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE?

(Note that mixed strategy $x$ is fixed throughout sparsification of $y$.)

**Definition ($\epsilon$-MNE, pure strategy version)**

Pair $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-MNE if

$$x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2} \quad i = 1, \ldots, m,$$

$$x^T By^\epsilon \leq x^T Be^j + \frac{\epsilon}{2} \quad j = 1, \ldots, n.$$

That is, players both have no improving move to pure strategy.
Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE? 
(Note that mixed strategy $x$ is fixed throughout sparsification of $y$.)

**Definition ($\epsilon$-MNE, pure strategy version)**

Pair $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-MNE if

$$x^T A y^\epsilon \leq (e^i)^T A y^\epsilon + \frac{\epsilon}{2} \quad i = 1, \ldots, m,$$

$$x^T B y^\epsilon \leq x^T B e^j + \frac{\epsilon}{2} \quad j = 1, \ldots, n.$$ 

That is, players both have no **improving move** to pure strategy.

For Bob, we want $x^T B y^\epsilon \leq x^T B e^j + \epsilon/2$ for $j = 1, \ldots, n.$
Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE? 
(Note that mixed strategy $x$ is fixed throughout sparsification of $y$.)

**Definition ($\epsilon$-MNE, pure strategy version)**

Pair $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-MNE if

\[
x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2} \quad i = 1, \ldots, m, \\
x^T By^\epsilon \leq x^T Be^i + \frac{\epsilon}{2} \quad j = 1, \ldots, n.
\]

That is, players both have no improving move to pure strategy.

For Bob, we want $x^T By^\epsilon \leq x^T Be^i + \epsilon/2$ for $j = 1, \ldots, n$.

- If expected cost of Bob does not change much, i.e.,
Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE? (Note that mixed strategy $x$ is fixed throughout sparsification of $y$.)

**Definition ($\epsilon$-MNE, pure strategy version)**

Pair $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-MNE if

$$
x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2} \quad i = 1, \ldots, m,
$$

$$
x^T By^\epsilon \leq x^T Be^i + \frac{\epsilon}{2} \quad j = 1, \ldots, n.
$$

That is, players both have no improving move to pure strategy.

For Bob, we want $x^T By^\epsilon \leq x^T Be^i + \epsilon/2$ for $j = 1, \ldots, n$.

- If expected cost of Bob does not change much, i.e.,
  $$
  \left| x^T By - x^T By^\epsilon \right| \leq \epsilon/2,
  $$(1)
Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE?

(Note that mixed strategy $x$ is fixed throughout sparsification of $y$.)

Definition ($\epsilon$-MNE, pure strategy version)

Pair $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-MNE if

$$x^T Ay^\epsilon \leq (e_i^j)^T Ay^\epsilon + \frac{\epsilon}{2} \quad i = 1, \ldots, m,$$

$$x^T By^\epsilon \leq x^T Be^j + \frac{\epsilon}{2} \quad j = 1, \ldots, n.$$  

That is, players both have no improving move to pure strategy.

For Bob, we want $x^T By^\epsilon \leq x^T Be^j + \epsilon/2$ for $j = 1, \ldots, n$.

- If expected cost of Bob does not change much, i.e.,
  $$|x^T By - x^T By^\epsilon| \leq \epsilon/2,$$
  (1)
  then, for any pure strategy $e^j$ with $j = 1, \ldots, m$, 

20 / 28
Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE?

(Note that mixed strategy $x$ is fixed throughout sparsification of $y$.)

**Definition ($\epsilon$-MNE, pure strategy version)**

Pair $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-MNE if

$$x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2} \quad i = 1, \ldots, m,$$

$$x^T By^\epsilon \leq x^T Be^j + \frac{\epsilon}{2} \quad j = 1, \ldots, n.$$ 

That is, players both have no improving move to pure strategy.

For Bob, we want $x^T By^\epsilon \leq x^T Be^j + \frac{\epsilon}{2}$ for $j = 1, \ldots, n$.

If expected cost of Bob does not change much, i.e.,

$$|x^T By - x^T By^\epsilon| \leq \frac{\epsilon}{2},$$

then, for any pure strategy $e^j$ with $j = 1, \ldots, m$,

$$x^T By^\epsilon$$
Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE? (Note that mixed strategy $x$ is fixed throughout sparsification of $y$.)

**Definition ($\epsilon$-MNE, pure strategy version)**

Pair $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-MNE if

\[
\begin{align*}
    x^T Ay^\epsilon &\leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2} & i = 1, \ldots, m, \\
    x^T By^\epsilon &\leq x^T Be^j + \frac{\epsilon}{2} & j = 1, \ldots, n.
\end{align*}
\]

That is, players both have no improving move to pure strategy.

For Bob, we want $x^T By^\epsilon \leq x^T Be^j + \epsilon/2$ for $j = 1, \ldots, n$.

- If expected cost of Bob does not change much, i.e.,
  \[
  \left| x^T By - x^T By^\epsilon \right| \leq \epsilon/2,
  \]
  then, for any pure strategy $e^j$ with $j = 1, \ldots, m$,
  \[
  x^T By^\epsilon \leq x^T By + \epsilon/2
  \]
Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE?

(Note that mixed strategy $x$ is fixed throughout sparsification of $y$.)

**Definition ($\epsilon$-MNE, pure strategy version)**

Pair $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-MNE if

$$x^TAy^\epsilon \leq (e^i)^TAy^\epsilon + \frac{\epsilon}{2}, \quad i = 1, \ldots, m,$$

$$x^TBy^\epsilon \leq x^TB(e^j) + \frac{\epsilon}{2}, \quad j = 1, \ldots, n.$$

That is, players both have no improving move to pure strategy.

For Bob, we want $x^TBy^\epsilon \leq x^TB(e^j) + \epsilon/2$ for $j = 1, \ldots, n$.

If expected cost of Bob does not change much, i.e.,

$$|x^TBy - x^TBy^\epsilon| \leq \epsilon/2,$$

then, for any pure strategy $e^j$ with $j = 1, \ldots, m$,

$$x^TBy^\epsilon \leq x^TBy + \epsilon/2 \leq x^TB(e^j) + \epsilon/2.$$


Sparsifying mixed strategy $y$

What should the mixed strategy $y^\epsilon$ satisfy for $(x, y^\epsilon)$ to be $\frac{\epsilon}{2}$-MNE?

(Note that mixed strategy $x$ is fixed throughout sparsification of $y$.)

**Definition ($\epsilon$-MNE, pure strategy version)**

Pair $(x, y^\epsilon)$ is $\frac{\epsilon}{2}$-MNE if

\[
x^T A y^\epsilon \leq (e^i)^T A y^\epsilon + \frac{\epsilon}{2} \quad i = 1, \ldots, m,
\]

\[
x^T B y^\epsilon \leq x^T B e^j + \frac{\epsilon}{2} \quad j = 1, \ldots, n.
\]

That is, players both have no improving move to pure strategy.

For Bob, we want $x^T B y^\epsilon \leq x^T B e^j + \epsilon/2$ for $j = 1, \ldots, n$.

- If expected cost of Bob does not change much, i.e.,
  \[
  |x^T B y - x^T B y^\epsilon| \leq \epsilon/2,
  \]
  then, for any pure strategy $e^j$ with $j = 1, \ldots, m$,

  \[
x^T B y^\epsilon \leq x^T B y + \epsilon/2 \leq x^T B(e^j) + \epsilon/2.
  \]

- Second inequality holds because $(x, y)$ is MNE.
For Alice, we want $x^T A y^c \leq (e^i)^T A y^c + \frac{\epsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.
For Alice, we want $x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have
  \[ |(Ay)_i - (Ay^\epsilon)_i| \leq \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m. \] (2)
For Alice, we want $x^T Ay^c \leq (e^i)^T Ay^c + \frac{\varepsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have
  $$|(Ay)_i - (Ay^c)_i| \leq \frac{\varepsilon}{4} \text{ for } i = 1, \ldots, m. \quad (2)$$

- This is the same as saying $||Ay - Ay^c||_\infty \leq \frac{\varepsilon}{4}$.
For Alice, we want \( x^T Ay^\epsilon \leq (e_i)^T Ay^\epsilon + \frac{\epsilon}{2} \) for \( i = 1, \ldots, m \).

The **expected cost per row for Alice** should not change much.

- Suffices to have
  \[
  |(Ay)_i - (Ay^\epsilon)_i| \leq \frac{\epsilon}{4} \quad \text{for } i = 1, \ldots, m. 
  \]  

- This is the same as saying \( \|Ay - Ay^\epsilon\|_\infty \leq \frac{\epsilon}{4} \).
  
  (Infinity norm: \( \|z\|_\infty = \max_i |z_i| \) for \( z = (z_1, \ldots, z_n) \in \mathbb{R}^n \).)
For Alice, we want $x^T Ay^e \leq (e^i)^T Ay^e + \frac{\varepsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have

$$|(Ay)_i - (Ay^e)_i| \leq \frac{\varepsilon}{4} \text{ for } i = 1, \ldots, m. \tag{2}$$

- This is the same as saying $||Ay - Ay^e||_{\infty} \leq \frac{\varepsilon}{4}$.
  - (Infinity norm: $||z||_{\infty} = \max_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Why?
For Alice, we want \( x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2} \) for \( i = 1, \ldots, m \).

The expected cost per row for Alice should not change much.

- Suffices to have

\[
|(Ay)_i - (Ay^\epsilon)_i| \leq \frac{\epsilon}{4} \quad \text{for } i = 1, \ldots, m. \tag{2}
\]

- This is the same as saying \( \|Ay - Ay^\epsilon\|_\infty \leq \frac{\epsilon}{4} \).

  (Infinity norm: \( \|z\|_\infty = \max_i |z_i| \) for \( z = (z_1, \ldots, z_n) \in \mathbb{R}^n \).)

**Why?** Inequality (2) implies

\[
|x^T Ay - x^T Ay^\epsilon| \leq \|x\|_1 \|Ay - Ay^\epsilon\|_\infty \leq \epsilon/4 \tag{3}
\]
For Alice, we want $x^T Ay^c \leq (e_i)^T Ay^c + \frac{\varepsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have

  $$|(Ay)_i - (Ay^c)_i| \leq \frac{\varepsilon}{4} \text{ for } i = 1, \ldots, m. \quad (2)$$

- This is the same as saying $||Ay - Ay^c||_\infty \leq \frac{\varepsilon}{4}$.

  (Infinity norm: $||z||_\infty = \max_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

**Why?** Inequality (2) implies

$$|x^T Ay - x^T Ay^c| \leq ||x||_1 ||Ay - Ay^c||_\infty \leq \varepsilon / 4 \quad (3)$$

- (1-norm: $||z||_1 = \sum_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)
For Alice, we want $x^T Ay^c \leq (e^i)^T Ay^c + \frac{\epsilon}{2}$ for $i = 1, \ldots, m$. 

The expected cost per row for Alice should not change much.

- Suffices to have
  
  $$ |(Ay)_i - (Ay^c)_i| \leq \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m. \quad (2) $$

  - This is the same as saying $||Ay - Ay^c||_\infty \leq \frac{\epsilon}{4}$.
  - (Infinity norm: $||z||_\infty = \max_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

**Why?** Inequality (2) implies

$$ |x^T Ay - x^T Ay^c| \leq ||x||_1 ||Ay - Ay^c||_\infty \leq \epsilon/4 \quad (3) $$

- (1-norm: $||z||_1 = \sum_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Now, for pure strategy $e^i$ for $i = 1, \ldots, m$ of Alice,
For Alice, we want $x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have

$$|(Ay)_i - (Ay^\epsilon)_i| \leq \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m. \quad (2)$$

- This is the same as saying $||Ay - Ay^\epsilon||_\infty \leq \frac{\epsilon}{4}$.
  - (Infinity norm: $||z||_\infty = \max_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

**Why?** Inequality (2) implies

$$|x^T Ay - x^T Ay^\epsilon| \leq ||x||_1 ||Ay - Ay^\epsilon||_\infty \leq \epsilon/4 \quad (3)$$

- (1-norm: $||z||_1 = \sum_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Now, for pure strategy $e^i$ for $i = 1, \ldots, m$ of Alice, we have

$x^T Ay^\epsilon$
For Alice, we want $x^T Ay^\epsilon \leq (e_i)^T Ay^\epsilon + \frac{\epsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.
- Suffices to have
  \[
  |(Ay)_i - (Ay^\epsilon)_i| \leq \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m. \tag{2}
  \]
- This is the same as saying $||Ay - Ay^\epsilon||_\infty \leq \frac{\epsilon}{4}$.
  - (Infinity norm: $||z||_\infty = \max_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

**Why?** Inequality (2) implies
\[
|x^T Ay - x^T Ay^\epsilon| \leq ||x||_1 ||Ay - Ay^\epsilon||_\infty \leq \frac{\epsilon}{4} \tag{3}
\]
- (1-norm: $||z||_1 = \sum_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Now, for pure strategy $e^i$ for $i = 1, \ldots, m$ of Alice, we have
\[
x^T Ay^\epsilon \leq x^T Ay + \frac{\epsilon}{4}
\]
For Alice, we want $x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have
  $$| (Ay)_i - (Ay^\epsilon)_i | \leq \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m. \quad (2)$$
  - This is the same as saying $||Ay - Ay^\epsilon||_\infty \leq \frac{\epsilon}{4}$.
    - (Infinity norm: $||z||_\infty = \max_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Why? Inequality (2) implies
$$|x^T Ay - x^T Ay^\epsilon| \leq ||x||_1 ||Ay - Ay^\epsilon||_\infty \leq \frac{\epsilon}{4} \quad (3)$$

- (1-norm: $||z||_1 = \sum_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Now, for pure strategy $e^i$ for $i = 1, \ldots, m$ of Alice, we have
$$x^T Ay^\epsilon \leq x^T Ay + \frac{\epsilon}{4} \leq (e^i)^T Ay + \frac{\epsilon}{4}$$
For Alice, we want $x^T Ay^\varepsilon \leq (e^i)^T Ay^\varepsilon + \frac{\varepsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have
  
  $$|(Ay)_i - (Ay^\varepsilon)_i| \leq \frac{\varepsilon}{4} \text{ for } i = 1, \ldots, m. \quad (2)$$

- This is the same as saying $||Ay - Ay^\varepsilon||_\infty \leq \frac{\varepsilon}{4}$.
  
  (Infinity norm: $||z||_\infty = \max_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

**Why?** Inequality (2) implies

$$|x^T Ay - x^T Ay^\varepsilon| \leq ||x||_1 ||Ay - Ay^\varepsilon||_\infty \leq \varepsilon/4 \quad (3)$$

(1-norm: $||z||_1 = \sum_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Now, for pure strategy $e^i$ for $i = 1, \ldots, m$ of Alice, we have

$$x^T Ay^\varepsilon \leq x^T Ay + \frac{\varepsilon}{4} \leq (e^i)^T Ay + \frac{\varepsilon}{4} \leq (e^i)^T Ay^\varepsilon + \frac{\varepsilon}{4} + \frac{\varepsilon}{4}.$$
For Alice, we want $x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have

  $$|(Ay)_i - (Ay^\epsilon)_i| \leq \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m. \quad (2)$$

- This is the same as saying $||Ay - Ay^\epsilon||_\infty \leq \frac{\epsilon}{4}$.

  (Infinity norm: $||z||_\infty = \max_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Why? Inequality (2) implies

$$|x^T Ay - x^T Ay^\epsilon| \leq ||x||_1 ||Ay - Ay^\epsilon||_\infty \leq \epsilon/4 \quad (3)$$

(1-norm: $||z||_1 = \sum_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Now, for pure strategy $e^i$ for $i = 1, \ldots, m$ of Alice, we have

$$x^T Ay^\epsilon \leq x^T Ay + \frac{\epsilon}{4} \leq (e^i)^T Ay + \frac{\epsilon}{4} \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{4} + \frac{\epsilon}{4}.$$

- Remember that $(Ay)_i = (e^i)^T Ay$
For Alice, we want $x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have

  $$|(Ay)_i - (Ay^\epsilon)_i| \leq \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m.$$ (2)

- This is the same as saying $||Ay - Ay^\epsilon||_\infty \leq \frac{\epsilon}{4}$.

  (Infinity norm: $||z||_\infty = \max_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Why? Inequality (2) implies

$$|x^T Ay - x^T Ay^\epsilon| \leq ||x||_1 ||Ay - Ay^\epsilon||_\infty \leq \epsilon/4$$ (3)

(1-norm: $||z||_1 = \sum_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Now, for pure strategy $e^i$ for $i = 1, \ldots, m$ of Alice, we have

$$x^T Ay^\epsilon \leq x^T Ay + \frac{\epsilon}{4} \leq (e^i)^T Ay + \frac{\epsilon}{4} \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{4} + \frac{\epsilon}{4}.$$ 

- Remember that $(Ay)_i = (e^i)^T Ay$

- Inequalities use (3), fact that $(x, y)$ is MNE, and (2) (respectively).
For Alice, we want $x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have
  
  \[ |(Ay)_i - (Ay^\epsilon)_i| \leq \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m. \]  

  \[ (2) \]

- This is the same as saying $||Ay - Ay^\epsilon||_\infty \leq \frac{\epsilon}{4}$.
  - (Infinity norm: $||z||_\infty = \max |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

**Why?** Inequality (2) implies

\[ |x^T Ay - x^T Ay^\epsilon| \leq ||x||_1 ||Ay - Ay^\epsilon||_\infty \leq \epsilon/4 \]  

\[ (3) \]

- (1-norm: $||z||_1 = \sum_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Now, for pure strategy $e^i$ for $i = 1, \ldots, m$ of Alice, we have

\[ x^T Ay^\epsilon \leq x^T Ay + \frac{\epsilon}{4} \leq (e^i)^T Ay + \frac{\epsilon}{4} \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{4} + \frac{\epsilon}{4}. \]

- Remember that $(Ay)_i = (e^i)^T Ay$

- Inequalities use (3), fact that $(x, y)$ is MNE, and (2) (respectively).
For Alice, we want $x^T Ay^\epsilon \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{2}$ for $i = 1, \ldots, m$.

The expected cost per row for Alice should not change much.

- Suffices to have
  
  $$|(Ay)_i - (Ay^\epsilon)_i| \leq \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m. \quad (2)$$

- This is the same as saying $||Ay - Ay^\epsilon||_{\infty} \leq \frac{\epsilon}{4}$.
  - (Infinity norm: $||z||_{\infty} = \max_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

**Why?** Inequality (2) implies

$$|x^T Ay - x^T Ay^\epsilon| \leq ||x||_1 ||Ay - Ay^\epsilon||_{\infty} \leq \epsilon/4 \quad (3)$$

- (1-norm: $||z||_1 = \sum_i |z_i|$ for $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$.)

Now, for pure strategy $e^i$ for $i = 1, \ldots, m$ of Alice, we have

$$x^T Ay^\epsilon \leq x^T Ay + \frac{\epsilon}{4} \leq (e^i)^T Ay + \frac{\epsilon}{4} \leq (e^i)^T Ay^\epsilon + \frac{\epsilon}{4} + \frac{\epsilon}{4}.$$

- Remember that $(Ay)_i = (e^i)^T Ay$
- Inequalities use (3), fact that $(x, y)$ is MNE, and (2) (respectively).

**Exercise:** Show that having $|x^T Ay - x^T Ay^\epsilon| < \epsilon/2$ is not sufficient!
To summarize, $(x, y^\epsilon)$ will be an $\frac{\epsilon}{2}$-MNE, if $y^\epsilon$ satisfies

$$\left| x^T By - x^T By^\epsilon \right| \leq \frac{\epsilon}{2}$$

$$\|Ay - Ay^\epsilon\|_\infty \leq \frac{\epsilon}{4}$$
To summarize, \((x, y^\epsilon)\) will be an \(\epsilon/2\)-MNE, if \(y^\epsilon\) satisfies

\[
\left|x^T By - x^T By^\epsilon\right| \leq \epsilon/2
\]

\[
\|Ay - Ay^\epsilon\|_\infty \leq \epsilon/4
\]

Does there always exist such a vector \(y^\epsilon\) with \(\text{Supp}(y^\epsilon) = O(\log(m)/\epsilon^2)\)?
To summarize, \((x, y^\epsilon)\) will be an \(\frac{\epsilon}{2}\)-MNE, if \(y^\epsilon\) satisfies
\[
| x^T B y - x^T B y^\epsilon | \leq \frac{\epsilon}{2} \\
|| A y - A y^\epsilon ||_{\infty} \leq \frac{\epsilon}{4}
\]

Does there always exist such a vector \(y^\epsilon\) with \(\text{Supp}(y^\epsilon) = O(\log(m)/\epsilon^2)\)?

Yes!
To summarize, \((x, y^\epsilon)\) will be an \(\frac{\epsilon}{2}\)-MNE, if \(y^\epsilon\) satisfies
\[
\left| x^T By - x^T B y^\epsilon \right| \leq \frac{\epsilon}{2} \\
\| A y - A y^\epsilon \|_\infty \leq \frac{\epsilon}{4}
\]

*Does there always exist such a vector \(y^\epsilon\) with \(\text{Supp}(y^\epsilon) = O(\log(m)/\epsilon^2)\)?*

Yes!

A concise representation of requirements
To summarize, \((x, y^\epsilon)\) will be an \(\frac{\epsilon}{2}\)-MNE, if \(y^\epsilon\) satisfies
\[
\begin{align*}
|x^T By - x^T By^\epsilon| &\leq \frac{\epsilon}{2} \\
||Ay - Ay^\epsilon||_\infty &\leq \frac{\epsilon}{4}
\end{align*}
\]

Does there always exist such a vector \(y^\epsilon\) with \(\text{Supp}(y^\epsilon) = O(\log(m)/\epsilon^2)\)?
Yes!

A concise representation of requirements
Consider \((m + 1) \times n\) matrix obtained by appending row-vector \(x^T B\) to \(A\), i.e.,
To summarize, \((x, y^\epsilon)\) will be an \(\frac{\epsilon}{2}\)-MNE, if \(y^\epsilon\) satisfies
\[
\begin{align*}
|x^T By - x^T B y^\epsilon| &\leq \frac{\epsilon}{2} \\
||Ay - Ay^\epsilon||_{\infty} &\leq \frac{\epsilon}{4}
\end{align*}
\]

*Does there always exist such a vector \(y^\epsilon\) with \(\text{Supp}(y^\epsilon) = O(\log(m)/\epsilon^2)\)?*

*Yes!*

A concise representation of requirements
Consider \((m + 1) \times n\) matrix obtained by appending row-vector \(x^T B\) to \(A\), i.e.,
\[
A' = \begin{pmatrix} A \\ x^T B \end{pmatrix}.
\]
To summarize, \((x, y^\epsilon)\) will be an \(\frac{\epsilon}{2}\)-MNE, if \(y^\epsilon\) satisfies

\[
\left| x^T B y - x^T B y^\epsilon \right| \leq \frac{\epsilon}{2}
\]
\[
\| A y - A y^\epsilon \|_{\infty} \leq \frac{\epsilon}{4}
\]

Does there always exist such a vector \(y^\epsilon\) with \(\text{Supp}(y^\epsilon) = O(\log(m)/\epsilon^2)\)?

Yes!

A concise representation of requirements
Consider \((m + 1) \times n\) matrix obtained by appending row-vector \(x^T B\) to \(A\), i.e.,

\[
A' = \begin{pmatrix} A \\ x^T B \end{pmatrix}
\]

The pair \((x, y^\epsilon)\) will be an \(\frac{\epsilon}{2}\)-MNE, if \(y^\epsilon \in \Delta_B\) satisfies

\[
\| A'y - A'y^\epsilon \|_{\infty} \leq \frac{\epsilon}{4}.
\]
Theorem (Sparse vector approximation)

Let \( D \in [-1, 1]^{(m+1) \times n} \) and let \( y \in \Delta_B = \Delta_n \).

Example (Empirical distribution)

Let \( n = 4 \). If \( S_{\epsilon} = \{ b_1, b_2, b_3, b_2, b_3, b_2 \} \), then \( y_{\epsilon} = (1/6, 3/6, 2/6, 0) \).

Remark

It holds that \( |\text{Supp}(y_{\epsilon})| \leq O(\log(m)/\epsilon^2) \), i.e., the vector \( y_{\epsilon} \) has at most \( O(\log(m)/\epsilon^2) \) non-zero entries.
Theorem (Sparse vector approximation)

Let $D \in [-1, 1]^{(m+1) \times n}$ and let $y \in \Delta_B = \Delta_n$. For any $\varepsilon > 0$ there is a multi-set $S_{\varepsilon}$ of columns in $\{b_1, \ldots, b_n\}$ of size $|S_{\varepsilon}| = O\left(\frac{\log(m)}{\varepsilon^2}\right)$ such that the empirical distribution $y_{\varepsilon} = \frac{1}{|S_{\varepsilon}|} \sum_{j \in S_{\varepsilon}} e_j$ satisfies

$$||Dy - Dy_{\varepsilon}||_\infty = \max_{i=1, \ldots, m+1} |(Dy)_i - (Dy_{\varepsilon})_i| \leq \varepsilon/4.$$ 

Here $e_j \in \{0, 1\}^n$ is defined as usual (with $e_j^k = 1$ if and only if $j = k$).

Example (Empirical distribution)

Let $n = 4$. If $S_{\varepsilon} = \{b_1, b_2, b_3, b_2, b_3, b_2\}$, then $y_{\varepsilon} = \frac{1}{6}(1, 3, 2, 0)$.

Remark

It holds that $|\text{Supp}(y_{\varepsilon})| \leq O\left(\frac{\log(m)}{\varepsilon^2}\right)$, i.e., the vector $y_{\varepsilon}$ has at most $O\left(\frac{\log(m)}{\varepsilon^2}\right)$ non-zero entries.
Theorem (Sparse vector approximation)

Let $D \in [-1, 1]^{(m+1) \times n}$ and let $y \in \Delta_B = \Delta_n$. For any $\epsilon > 0$ there is a multi-set $S_\epsilon$ of columns in $\{b_1, \ldots, b_n\}$ of size $|S_\epsilon| = O(\log(m)/\epsilon^2)$ such that the empirical distribution

$$y^\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j$$

Here $e^j \in \{0, 1\}^n$ is defined as usual (with $e^j_k = 1$ if and only if $j = k$).
Theorem (Sparse vector approximation)

Let \( D \in [-1, 1]^{(m+1) \times n} \) and let \( y \in \Delta_B = \Delta_n \). For any \( \epsilon > 0 \) there is a multi-set \( S_\epsilon \) of columns in \( \{b_1, \ldots, b_n\} \) of size \( |S_\epsilon| = O(\log(m)/\epsilon^2) \) such that the empirical distribution

\[
y_\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j
\]

satisfies

\[
\|Dy - Dy_\epsilon\|_\infty = \max_{i=1, \ldots, m+1} |(Dy)_i - (Dy_\epsilon)_i| \leq \epsilon/4.
\]
Sparse approximation of vectors

**Theorem (Sparse vector approximation)**

Let $D \in [-1, 1]^{(m+1) \times n}$ and let $y \in \Delta_B = \Delta_n$. For any $\epsilon > 0$ there is a multi-set $S_\epsilon$ of columns in $\{b_1, \ldots, b_n\}$ of size $|S_\epsilon| = O(\log(m)/\epsilon^2)$ such that the empirical distribution

$$y^\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j$$

satisfies $||Dy - Dy^\epsilon||_\infty = \max_{i=1,\ldots,m+1} |(Dy)_i - (Dy^\epsilon)_i| \leq \epsilon/4$.

Here $e^j \in \{0, 1\}^n$ is defined as usual (with $e^j_k = 1$ if and only if $j = k$).
**Theorem (Sparse vector approximation)**

Let \( D \in [-1, 1]^{(m+1) \times n} \) and let \( y \in \Delta_B = \Delta_n \). For any \( \epsilon > 0 \) there is a multi-set \( S_\epsilon \) of columns in \( \{b_1, \ldots, b_n\} \) of size \( |S_\epsilon| = O(\log(m)/\epsilon^2) \) such that the empirical distribution

\[
y_\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j
\]

satisfies

\[
\|Dy - Dy_\epsilon\|_\infty = \max_{i=1,\ldots,m+1} |(Dy)_i - (Dy_\epsilon)_i| \leq \epsilon/4.
\]

Here \( e^j \in \{0, 1\}^n \) is defined as usual (with \( e^j_k = 1 \) if and only if \( j = k \)).

**Example (Empirical distribution)**

Let \( n = 4 \). If \( S_\epsilon = \{b_1, b_2, b_3, b_2, b_3, b_2\} \), then \( y_\epsilon = \frac{1}{6}(1, 3, 2, 0) \).
Sparse approximation of vectors

Theorem (Sparse vector approximation)

Let $D \in [-1, 1]^{(m+1) \times n}$ and let $y \in \Delta_B = \Delta_n$. For any $\epsilon > 0$ there is a multi-set $S_\epsilon$ of columns in $\{b_1, \ldots, b_n\}$ of size $|S_\epsilon| = O(\log(m)/\epsilon^2)$ such that the empirical distribution

$$y^\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j$$

satisfies $\|Dy - Dy^\epsilon\|_\infty = \max_{i=1,\ldots,m+1} |(Dy)_i - (Dy^\epsilon)_i| \leq \epsilon/4$.

Here $e^j \in \{0, 1\}^n$ is defined as usual (with $e^j_k = 1$ if and only if $j = k$).

Example (Empirical distribution)

Let $n = 4$. If $S_\epsilon = \{b_1, b_2, b_3, b_2, b_3, b_2\}$, then $y^\epsilon = \frac{1}{6}(1, 3, 2, 0)$.

Remark

It holds that $|\text{Supp}(y^\epsilon)| \leq O(\log(m)/\epsilon^2)$,
Sparse approximation of vectors

Theorem (Sparse vector approximation)

Let \( D \in [-1, 1]^{(m+1) \times n} \) and let \( y \in \Delta_B = \Delta_n \). For any \( \epsilon > 0 \) there is a multi-set \( S_\epsilon \) of columns in \( \{b_1, \ldots, b_n\} \) of size \( |S_\epsilon| = O(\log(m)/\epsilon^2) \) such that the empirical distribution

\[
y_\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j
\]

satisfies

\[
||Dy - Dy_\epsilon||_\infty = \max_{i=1,\ldots,m+1} |(Dy)_i - (Dy_\epsilon)_i| \leq \epsilon/4.
\]

Here \( e^j \in \{0, 1\}^n \) is defined as usual (with \( e^j_k = 1 \) if and only if \( j = k \)).

Example (Empirical distribution)

Let \( n = 4 \). If \( S_\epsilon = \{b_1, b_2, b_3, b_2, b_3, b_2\} \), then \( y_\epsilon = \frac{1}{6}(1, 3, 2, 0) \).

Remark

It holds that \( |\text{Supp}(y_\epsilon)| \leq O(\log(m)/\epsilon^2) \), i.e., the vector \( y_\epsilon \) has at most \( O(\log(m)/\epsilon^2) \) non-zero entries.
Proof of theorem:

Fix $\epsilon > 0$ and let $y \in \Delta B$.

Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$. That is, we have $P(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$.

Write $e_{cr}$ for pure strategy corresponding to (random) column $c_r$.

Remember that $y \epsilon = 1^T \sum_{r=1}^T r = 1 e_{cr}$.

It suffices to show that, if $T = O(\log(m)/\epsilon^2)$, $P(\|Dy \epsilon_i - (Dy) i\| < \epsilon/4$ for $i = 1, \ldots, m + 1$) > 0 (4).

Why? Because this implies that there is some (deterministic) multi-set of columns $S \epsilon$, with $|S \epsilon| = O(\log(m)/\epsilon^2)$, for which its empirical distribution $y \epsilon$ satisfies $\|Dy \epsilon_i - (Dy) i\| < \epsilon/4$ for $i = 1, \ldots, m + 1$.

This is called the probabilistic method.

Very roughly: Define random process, and show desired object is outputted with strictly positive probability. It is non-constructive, as we do not know $y$!

Also note that $E[(Dy \epsilon_i - (Dy) i)] = E[(D(1^T \sum_{r=1}^T e_{cr})) i] = (Dy) i$. 

24 / 28
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. 
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$. 

\[ \text{Remember that } y \epsilon = 1^T \sum_{r=1}^T e_{cr} \text{ for every } r. \] 

It suffices to show that, if $T = O(\log(\frac{m}{\epsilon^2}))$, then 

\[ \Pr \left( \left| \left(Dy^\epsilon\right)_i - \left(Dy\right)_i \right| < \frac{\epsilon}{4} \right) > 0 \quad (4) \]

Why? Because this implies that there is some (deterministic) multi-set of columns $S \epsilon$, with $|S \epsilon| = O(\log(\frac{m}{\epsilon^2}))$, for which its empirical distribution $y \epsilon$ satisfies 

\[ \left| \left(Dy^\epsilon\right)_i - \left(Dy\right)_i \right| < \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m+1. \] 

This is called the probabilistic method. Very roughly: Define random process, and show desired object is outputted with strictly positive probability. It is non-constructive, as we do not know $y$!
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$.

- That is, we have $\mathbb{P}(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$. 

Remember that $y \in \Delta_B$, with $\sum_{r=1}^{T} y_r = 1$. It suffices to show that, if $T = O(\log(m)/\epsilon^2)$,

$$\mathbb{P}\left(|D_{y \epsilon} - D_{y \epsilon}| < \frac{\epsilon}{4}\right)$$

for $i = 1, \ldots, m+1$.

Why? Because this implies that there is some (deterministic) multi-set of columns $S \in \Delta_B$, with $|S| = O(\log(m)/\epsilon^2)$, for which its empirical distribution $y_S$ satisfies $|D_{y_S} - D_{y_S}| < \frac{\epsilon}{4}$ for $i = 1, \ldots, m+1$.

This is called the probabilistic method. Very roughly: Define random process, and show desired object is outputted with strictly positive probability. It is non-constructive, as we do not know $y$!
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$.

- That is, we have $\mathbb{P}(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$.
- Write $e^{c_r}$ for pure strategy corresponding to (random) column $c_r$. 

Remember that $y_\epsilon = 1^T \sum_{r=1}^T r = e^{c_r}$.

It suffices to show that, if $T = O(\log(m)/\epsilon^2)$,

$$\mathbb{P}(\left|\left|Dy_\epsilon - (Dy_\epsilon)\right|\left| < \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m + 1 \right) > 0 \quad (4)$$

Why? Because this implies that there is some (deterministic) multi-set of columns $S_\epsilon$, with $|S_\epsilon| = O(\log(m)/\epsilon^2)$, for which its empirical distribution $y_\epsilon$ satisfies $\left|\left|Dy_\epsilon - (Dy_\epsilon)\right|\left| < \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m + 1 \right.$.

This is called the probabilistic method. Very roughly: Define random process, and show desired object is outputted with strictly positive probability. It is non-constructive, as we do not know $y$!

Also note that $E[\left|Dy_\epsilon\right|_i] = E[\left|D(1^T \sum_{r=1}^T r)\right|_i] = 1^T \sum_{r=1}^T E[\left|De^{c_r}\right|_i] = (Dy_\epsilon)_i$. 


Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$.

- That is, we have $P(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$.
- Write $e^{c_r}$ for pure strategy corresponding to (random) column $c_r$.

Remember that $y^\epsilon = \frac{1}{T} \sum_{r=1}^T e^{c_r}$. 
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$.

- That is, we have $P(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$.
- Write $e^{c_r}$ for pure strategy corresponding to (random) column $c_r$.

Remember that $y^\epsilon = \frac{1}{T} \sum_{r=1}^T e^{c_r}$.

It suffices to show that, if $T = O(\log(m)/\epsilon^2)$,

$$P \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| < \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m + 1 \right) > 0 \quad (4)$$
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$.

- That is, we have $\mathbb{P}(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$.
- Write $e^{c_r}$ for pure strategy corresponding to (random) column $c_r$.

Remember that $y^\epsilon = \frac{1}{T} \sum_{r=1}^{T} e^{c_r}$.

It suffices to show that, if $T = O(\log(m)/\epsilon^2)$,

$$\mathbb{P} \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| < \epsilon/4 \text{ for } i = 1, \ldots, m+1 \right) > 0 \quad (4)$$

**Why?** Because this implies that there is some (deterministic) multi-set of columns $S_\epsilon$, with $|S_\epsilon| = O(\log(m)/\epsilon^2)$,
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$.

- That is, we have $P(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$.
- Write $e^{c_r}$ for pure strategy corresponding to (random) column $c_r$.

Remember that $y^\epsilon = \frac{1}{T} \sum_{r=1}^T e^{c_r}$.

It suffices to show that, if $T = O(\log(m)/\epsilon^2)$,

$$P\left(\left|\left(Dy^\epsilon\right)_i - (Dy)_i\right| < \epsilon/4 \text{ for } i = 1, \ldots, m + 1\right) > 0 \quad (4)$$

Why? Because this implies that there is some (deterministic) multi-set of columns $S_{\epsilon}$, with $|S_{\epsilon}| = O(\log(m)/\epsilon^2)$, for which its empirical distribution $y^\epsilon$ satisfies $\left|\left(Dy^\epsilon\right)_i - (Dy)_i\right| < \epsilon/4$ for $i = 1, \ldots, m + 1$. 


Proof of theorem: Fix \( \epsilon > 0 \) and let \( y \in \Delta_B \). Let \( c_1, \ldots, c_T \) be random columns in \( \{b_1, \ldots, b_n\} \) distributed according to \( y \).

- That is, we have \( \mathbb{P}(c_r = b_j) = y_j \) for \( j = 1, \ldots, n \) and every \( r \).
- Write \( e^{c_r} \) for pure strategy corresponding to (random) column \( c_r \).

Remember that \( y^\epsilon = \frac{1}{T} \sum_{r=1}^T e^{c_r} \).

It suffices to show that, if \( T = O(\log(m)/\epsilon^2) \),

\[
\mathbb{P} \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| < \epsilon/4 \text{ for } i = 1, \ldots, m + 1 \right) > 0 \quad (4)
\]

Why? Because this implies that there is some (deterministic) multi-set of columns \( S_\epsilon \), with \( |S_\epsilon| = O(\log(m)/\epsilon^2) \), for which its empirical distribution \( y^\epsilon \) satisfies \( \left| (Dy^\epsilon)_i - (Dy)_i \right| < \epsilon/4 \) for \( i = 1, \ldots, m + 1 \).

- This is called the probabilistic method.
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$.

- That is, we have $\mathbb{P}(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$.
- Write $e^{c_r}$ for pure strategy corresponding to (random) column $c_r$.

Remember that $y^\epsilon = \frac{1}{T} \sum_{r=1}^T e^{c_r}$.

It suffices to show that, if $T = O(\log(m)/\epsilon^2)$,

$$\mathbb{P} \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| < \epsilon/4 \text{ for } i = 1, \ldots, m+1 \right) > 0 \quad (4)$$

Why? Because this implies that there is some (deterministic) multi-set of columns $S_\epsilon$, with $|S_\epsilon| = O(\log(m)/\epsilon^2)$, for which its empirical distribution $y^\epsilon$ satisfies $\left| (Dy^\epsilon)_i - (Dy)_i \right| < \epsilon/4$ for $i = 1, \ldots, m+1$.

- This is called the probabilistic method.
  - Very roughly: Define random process,
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$.

- That is, we have $\mathbb{P}(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$.
- Write $e^{c_r}$ for pure strategy corresponding to (random) column $c_r$.

Remember that $y^\epsilon = \frac{1}{T} \sum_{r=1}^{T} e^{c_r}$.

It suffices to show that, if $T = O(\log(m)/\epsilon^2)$,

$$
\mathbb{P} \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| < \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m+1 \right) > 0 \quad (4)
$$

Why? Because this implies that there is some (deterministic) multi-set of columns $S_\epsilon$, with $|S_\epsilon| = O(\log(m)/\epsilon^2)$, for which its empirical distribution $y^\epsilon$ satisfies $\left| (Dy^\epsilon)_i - (Dy)_i \right| < \frac{\epsilon}{4}$ for $i = 1, \ldots, m+1$.

- This is called the probabilistic method.
  - Very roughly: Define random process, and show desired object is outputted with strictly positive probability.
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$.
- That is, we have $\mathbb{P}(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$.
- Write $e^{c_r}$ for pure strategy corresponding to (random) column $c_r$.

Remember that $y^\epsilon = \frac{1}{T} \sum_{r=1}^{T} e^{c_r}$.

It suffices to show that, if $T = O(\log(m)/\epsilon^2)$,

$$\mathbb{P} \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| < \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m + 1 \right) > 0 \quad (4)$$

Why? Because this implies that there is some (deterministic) multi-set of columns $S_\epsilon$, with $|S_\epsilon| = O(\log(m)/\epsilon^2)$, for which its empirical distribution $y^\epsilon$ satisfies $\left| (Dy^\epsilon)_i - (Dy)_i \right| < \epsilon/4$ for $i = 1, \ldots, m + 1$.

- This is called the probabilistic method.
- Very roughly: Define random process, and show desired object is outputted with strictly positive probability.

It is non-constructive, as we do not know $y$!
Proof of theorem: Fix $\epsilon > 0$ and let $y \in \Delta_B$. Let $c_1, \ldots, c_T$ be random columns in $\{b_1, \ldots, b_n\}$ distributed according to $y$.

- That is, we have $\mathbb{P}(c_r = b_j) = y_j$ for $j = 1, \ldots, n$ and every $r$.
- Write $e^{c_r}$ for pure strategy corresponding to (random) column $c_r$.

Remember that $y^\epsilon = \frac{1}{T} \sum_{r=1}^{T} e^{c_r}$.

It suffices to show that, if $T = O(\log(m)/\epsilon^2)$,

$$\mathbb{P} \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| < \frac{\epsilon}{4} \text{ for } i = 1, \ldots, m+1 \right) > 0 \quad (4)$$

Why? Because this implies that there is some (deterministic) multi-set of columns $S_\epsilon$, with $|S_\epsilon| = O(\log(m)/\epsilon^2)$, for which its empirical distribution $y^\epsilon$ satisfies $\left| (Dy^\epsilon)_i - (Dy)_i \right| < \frac{\epsilon}{4}$ for $i = 1, \ldots, m+1$.

- This is called the probabilistic method.
  - Very roughly: Define random process, and show desired object is outputted with strictly positive probability.
  - It is non-constructive, as we do not know $y$!

Also note that

$$\mathbb{E} \left[ (Dy^\epsilon)_i \right] = \mathbb{E} \left[ \left( D \left( \frac{1}{T} \sum_{r=1}^{T} e^{c_r} \right) \right)_i \right] = \frac{1}{T} \sum_{r=1}^{T} \mathbb{E} \left[ (De^{c_r})_i \right] = (Dy)_i.$$
In order to show (4), it suffices to show that for every individual $i$,

$$
\mathbb{P} \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| > \epsilon/4 \right) < \frac{1}{m + 1},
$$

(5)
In order to show (4), it suffices to show that for every individual $i$,

$$\mathbb{P}\left(\left| (Dy^\varepsilon)_i - (Dy)_i \right| > \varepsilon/4 \right) < \frac{1}{m + 1}, \quad (5)$$

- This follows from a union bound argument (check yourself!).
In order to show (4), it suffices to show that for every individual $i$,

$$\mathbb{P} \left( \left| \left( Dy^\epsilon \right)_i - (Dy)_i \right| > \epsilon/4 \right) < \frac{1}{m+1},$$

(5)

- This follows from a union bound argument (check yourself!).
- Remember that $\mathbb{E} [(Dy^\epsilon)_i] = (Dy)_i$ and

$$y^\epsilon = \frac{1}{T} \sum_{r=1}^{T} e^{cr}.$$
In order to show (4), it suffices to show that for every individual $i$,

$$P \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| > \epsilon/4 \right) < \frac{1}{m + 1}, \quad (5)$$

- This follows from a union bound argument (check yourself!).
- Remember that $\mathbb{E} [(Dy^\epsilon)_i] = (Dy)_i$ and

$$y^\epsilon = \frac{1}{T} \sum_{r=1}^{T} e^{cr}.$$ 

In order to bound probability that a random variable attains a value far away from its expectation, one needs a concentration inequality.
In order to show (4), it suffices to show that for every individual $i$,

$$\mathbb{P}\left(\left|\left(Dy^\epsilon\right)_i - (Dy)_i\right| > \epsilon/4\right) < \frac{1}{m + 1}, \tag{5}$$

- This follows from a union bound argument (check yourself!).
- Remember that $\mathbb{E}\left[(Dy^\epsilon)_i\right] = (Dy)_i$ and

$$y^\epsilon = \frac{1}{T} \sum_{r=1}^{T} e^{cr}.$$  

In order to bound probability that a random variable attains a value far away from its expectation, one needs a concentration inequality.

Hoeffding’s inequality implies that

$$\mathbb{P}\left(\left|\left(Dy^\epsilon\right)_i - (Dy)_i\right| > \epsilon/4\right) \leq 2 \exp\left(-\frac{T \epsilon^2}{16}\right).$$
In order to show (4), it suffices to show that for every individual $i$,
\[ P \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| > \epsilon/4 \right) < \frac{1}{m + 1}, \tag{5} \]

- This follows from a union bound argument (check yourself!).
- Remember that $E[(Dy^\epsilon)_i] = (Dy)_i$ and
\[ y^\epsilon = \frac{1}{T} \sum_{r=1}^{T} e^{cr}. \]

In order to bound probability that a random variable attains a value far away from its expectation, one needs a concentration inequality.

Hoeffding’s inequality implies that
\[ P \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| > \epsilon/4 \right) \leq 2 \exp \left( -\frac{T \epsilon^2}{16} \right). \]

How large should $T$ be so that (5) is satisfied?
In order to show (4), it suffices to show that for every individual $i$,

$$\mathbb{P} \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| > \epsilon/4 \right) < \frac{1}{m + 1}, \quad (5)$$

- This follows from a union bound argument (check yourself!).
- Remember that $\mathbb{E} [(Dy^\epsilon)_i] = (Dy)_i$ and

$$y^\epsilon = \frac{1}{T} \sum_{r=1}^{T} e^{cr}.$$

In order to bound probability that a random variable attains a value far away from its expectation, one needs a concentration inequality.

Hoeffding’s inequality implies that

$$\mathbb{P} \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| > \epsilon/4 \right) \leq 2 \exp \left( - \frac{T \epsilon^2}{16} \right).$$

How large should $T$ be so that (5) is satisfied? Take $T = O(\log(m)/\epsilon^2)$. 
In order to show (4), it suffices to show that for every individual $i$,

$$
P \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| > \epsilon/4 \right) < \frac{1}{m + 1}, \quad (5)
$$

- This follows from a union bound argument (check yourself!).
- Remember that $E[(Dy^\epsilon)_i] = (Dy)_i$ and

$$
y^\epsilon = \frac{1}{T} \sum_{r=1}^{T} e^{cr}.
$$

**In order to bound probability that a random variable attains a value far away from its expectation, one needs a concentration inequality.**

**Hoeffding’s inequality** implies that

$$
P \left( \left| (Dy^\epsilon)_i - (Dy)_i \right| > \epsilon/4 \right) \leq 2 \exp \left( -\frac{Te^2}{16} \right).
$$

**How large should $T$ be so that (5) is satisfied?** Take $T = O(\log(m)/\epsilon^2)$. \hfill \Box
Computation of approximate MNE

Final remarks
We use the following notation for finite game $\Gamma = (N, (S_i), (C_i))$ here.
Small support equilibria in multi-player games

We use the following notation for finite game $\Gamma = (N, (S_i), (C_i))$ here.

- $k = |N| \geq 2$ is the number of players.

Theorem (Lipton, Markakis and Mehta, 2003)

For every $\epsilon > 0$, there exists an $\epsilon$-MNE $(z_1, \ldots, z_k)$ where $|\text{Supp}(z_i)| = O\left(\frac{k^2 \log(k^2 m)}{\epsilon^2}\right)$

Can be improved to $O\left(\frac{\log(mk)}{\epsilon^2}\right)$ [Babichenko-Barman-Peretz, 2014].
Small support equilibria in multi-player games

We use the following notation for finite game $\Gamma = (N, (S_i), (C_i))$ here.

- $k = |N| \geq 2$ is the number of players.
- $m$ is number of strategies of every player, i.e., $|S_i| = m \ \forall i \in N$. 

Theorem (Lipton, Markakis and Mehta, 2003)

For every $\epsilon > 0$, there exists an $\epsilon$-MNE $(z_1, \ldots, z_k)$ where $|\text{Supp}(z_i)| = O\left(\frac{k^2 \log(mk)}{\epsilon^2}\right)$. 

Can be improved to $O\left(\log(mk)\epsilon^2\right)$ [Babichenko-Barman-Peretz, 2014].
Small support equilibria in multi-player games

We use the following notation for finite game $\Gamma = (N, (S_i), (C_i))$ here.

- $k = |N| \geq 2$ is the number of players.
- $m$ is number of strategies of every player, i.e., $|S_i| = m \forall i \in N$.

**Theorem (Lipton, Markakis and Mehta, 2003)**

For every $\epsilon > 0$, there exists an $\epsilon$-MNE $(z^1, \ldots, z^k)$ where

$$|\text{Supp}(z^i)| = O(k^2 \log(k^2 m)/\epsilon^2)$$

Can be improved to $O(\log(mk)/\epsilon^2)$ [Babichenko-Barman-Peretz, 2014].
Small support equilibria in multi-player games

We use the following notation for finite game $\Gamma = (N, (S_i), (C_i))$ here.

- $k = |N| \geq 2$ is the number of players.
- $m$ is number of strategies of every player, i.e., $|S_i| = m \ \forall \ i \in N$.

**Theorem (Lipton, Markakis and Mehta, 2003)**

For every $\epsilon > 0$, there exists an $\epsilon$-MNE $(z^1, \ldots, z^k)$ where

$$|\text{Supp}(z^i)| = O(k^2 \log(k^2 m)/\epsilon^2)$$

- Can be improved to $O\left(\frac{\log(mk)}{\epsilon^2}\right)$ [Babichenko-Barman-Peretz, 2014].
Remarks on sparse vector approximation

**Theorem (Sparse vector approximation)**

Let $D \in \left[-1, 1\right]^{(m+1) \times n}$ and let $y \in \Delta_B = \Delta_n$. For any $\epsilon > 0$ there is a multi-set $S_\epsilon$ of columns in $\{b_1, \ldots, b_n\}$ of size $|S_\epsilon| = O(\log(m)/\epsilon^2)$ such that the empirical distribution

$$y^\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j$$

satisfies $\|Dy - Dy^\epsilon\|_\infty = \max_i |(Dy)_i - (Dy^\epsilon)_i| \leq \epsilon/4.$
Remarks on sparse vector approximation

Theorem (Sparse vector approximation)

Let $D \in [-1, 1]^{(m+1) \times n}$ and let $y \in \Delta_B = \Delta_n$. For any $\epsilon > 0$ there is a multi-set $S_\epsilon$ of columns in $\{b_1, \ldots, b_n\}$ of size $|S_\epsilon| = O(\log(m)/\epsilon^2)$ such that the empirical distribution

$$y^\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j$$

satisfies $\|Dy - Dy^\epsilon\|_\infty = \max_i |(Dy)_i - (Dy^\epsilon)_i| \leq \epsilon/4$.

There exist many similar theorems like the above:
Remarks on sparse vector approximation

**Theorem (Sparse vector approximation)**

Let \( D \in [-1, 1]^{(m+1) \times n} \) and let \( y \in \Delta_B = \Delta_n \). For any \( \epsilon > 0 \) there is a multi-set \( S_\epsilon \) of columns in \( \{b_1, \ldots, b_n\} \) of size \( |S_\epsilon| = O(\log(m) / \epsilon^2) \) such that the empirical distribution

\[
y_\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j
\]

satisfies

\[
\|Dy - Dy_\epsilon\|_\infty = \max_i |(Dy)_i - (Dy_\epsilon)_i| \leq \epsilon/4.
\]

There exist many similar theorems like the above:

- Related to Maurey’s lemma, approximate Carathéodory’s theorem, . . .
Remarks on sparse vector approximation

**Theorem (Sparse vector approximation)**

Let $D \in [-1, 1]^{(m+1) \times n}$ and let $y \in \Delta_B = \Delta_n$. For any $\epsilon > 0$ there is a multi-set $S_{\epsilon}$ of columns in $\{b_1, \ldots, b_n\}$ of size $|S_{\epsilon}| = O(\log(m)/\epsilon^2)$ such that the empirical distribution

$$y^\epsilon = \frac{1}{|S_{\epsilon}|} \sum_{j \in S_{\epsilon}} e^j$$

satisfies $\|Dy - Dy^\epsilon\|_\infty = \max_i |(Dy)_i - (Dy^\epsilon)_i| \leq \epsilon/4$.

There exist many similar theorems like the above:

- Related to Maurey’s lemma, approximate Carathéodory’s theorem, …
- There is an $\ell_p$-norm version [Barman, 2018].
Remarks on sparse vector approximation

**Theorem (Sparse vector approximation)**

Let $D \in [-1, 1]^{(m+1) \times n}$ and let $y \in \Delta_B = \Delta_n$. For any $\epsilon > 0$ there is a multi-set $S_\epsilon$ of columns in \{ $b_1, \ldots, b_n$ \} of size $|S_\epsilon| = O(\log(m)/\epsilon^2)$ such that the empirical distribution

$$y^\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j$$

satisfies $\|Dy - Dy^\epsilon\|_\infty = \max_i |(Dy)_i - (Dy^\epsilon)_i| \leq \epsilon/4$.

There exist many similar theorems like the above:

- Related to Maurey’s lemma, approximate Carathéodory’s theorem, …
- There is an $\ell_p$-norm version [Barman, 2018].

There is also a refinement in terms of VC (or pseudo)-dimension of matrix $D$. 
Remarks on sparse vector approximation

**Theorem (Sparse vector approximation)**

Let \( D \in [-1, 1]^{(m+1) \times n} \) and let \( y \in \Delta_B = \Delta_n \). For any \( \epsilon > 0 \) there is a multi-set \( S_\epsilon \) of columns in \( \{b_1, \ldots, b_n\} \) of size \( |S_\epsilon| = O(\log(m)/\epsilon^2) \) such that the empirical distribution

\[
y^\epsilon = \frac{1}{|S_\epsilon|} \sum_{j \in S_\epsilon} e^j
\]

satisfies

\[
\|Dy - Dy^\epsilon\|_\infty = \max_i |(Dy)_i - (Dy^\epsilon)_i| \leq \epsilon/4.
\]

There exist many similar theorems like the above:

- Related to Maurey’s lemma, approximate Carathéodory’s theorem, …
- There is an \( \ell_p \)-norm version [Barman, 2018].

There is also a refinement in terms of VC (or pseudo)-dimension of matrix \( D \).

- Used to prove the “Fundamental Theorem of Statistical Learning”.

[Barman, 2018]