CPSC 768:
Scalable and Private Graph Algorithms

Lecture 16, 17, 18: Multiplicative Weight Updates

Quanquan C. Liu
quanquan.liu@yale.edu

Announcements

- **Progress reports (2-3 pages) for final project: Due April 5th.**
  - The final project as well as the 30 min presentation is due on the last day of class: **April 24th.**
- Class notes and schedule for the lectures for the rest of the semester have been posted on the course page
- Check the Course Slack for OPS announcements!

Last Time

- Weighted majority algorithm for predicting stock market
  - Take the majority opinion from sum of weights of N experts
  - Decrease weight of experts who were wrong

Last Time

- Weighted majority algorithm for predicting stock market
  - Take the majority opinion from sum of weights of N experts
  - Decrease weight of experts who were wrong

Theorem: # weighted majority mistakes $\leq$
$2(1+\varepsilon)\cdot$ best expert's # of mistakes + $O\left(\frac{\log(N)}{\varepsilon}\right)$

Another Stock Market Game

- Each day, more complicated interaction with stock, loss vector in $[-1, 1]$

Another Stock Market Game

- Each day, more complicated interaction with stock, loss vector in $[-1, 1]$
- N experts have **losses m_i^t for $i \in [N]$ on day $t \in [T]$ in $[-1, 1]$** which means you gain money for negative values and lose money for positive values

Another Stock Market Game

- Each day, more complicated interaction with stock, loss vector in $[-1, 1]$
- N experts have **losses $\boldsymbol{m_i^t}$ for $\boldsymbol{i \in [N]}$ on day $\boldsymbol{t \in [T]}$ in $\boldsymbol{[-1, 1]}$** which means you gain money for negative values and lose money for positive values
- We come up with a **probability distribution $\vec{\boldsymbol{p}}^{\boldsymbol{t}} = [\boldsymbol{p_1^t}, \dots, \boldsymbol{p_N^t}]$** where with probability p_i^t we pick expert i's opinion to output

Another Stock Market Game

- Each day, more complicated interaction with stock, loss vector in $[-1, 1]$
- N experts have **losses m_i^t for $i \in [N]$ on day $t \in [T]$ in $[-1, 1]$** which means you gain money for negative values and lose money for positive values
- We come up with a **probability distribution $\vec{p}^t = [p_1^t, \dots, p_N^t]$** where with probability p_i^t we pick expert i's opinion to output

Loss on day t: $\langle \vec{m}^t, \vec{p}^t \rangle$

Another Stock Market Game

- Each day, more complicated interaction with stock, loss vector in $[-1, 1]$
- N experts have **losses m_i^t for $i \in [N]$ on day $t \in [T]$ in $[-1, 1]$** which means you gain money for negative values and lose money for positive values
- We come up with a **probability distribution $\vec{p}^t = [p_1^t, \dots, p_N^t]$** where with probability p_i^t we pick expert i's opinion

Loss on day t: $\langle \vec{m}^t, \vec{p}^t \rangle$

Expected Loss

Hedge Algorithm

1. Initialize each $w_i^1 \leftarrow 1$ for each $i \in [N]$

Hedge Algorithm

1. Initialize each $w_i^1 \leftarrow 1$ for each $i \in [N]$
2. For each $t \in [T]$:

Hedge Algorithm

1. Initialize each $w_i^1 \leftarrow 1$ for each $i \in [N]$
2. For each $t \in [T]$:
   a) Set $p_i^t \leftarrow \frac{w_i^t}{\sum_{j \in [N]} w_j^t}$

Hedge Algorithm

1. Initialize each $w_i^1 \leftarrow 1$ for each $i \in [N]$
2. For each $t \in [T]$:
   a) Set $p_i^t \leftarrow \frac{w_i^t}{\sum_{j \in [N]} w_j^t}$
   b) Observe the loss vector $\vec{m}^t$

Hedge Algorithm

1. Initialize each $w_i^1 \leftarrow 1$ for each $i \in [N]$
2. For each $t \in [T]$:
    a) Set $p_i^t \leftarrow \frac{w_i^t}{\sum_{j \in [N]} w_j^t}$
    b) Observe the loss vector $\vec{m}^t$
    c) For each $i \in [N]$:

Hedge Algorithm

1. Initialize each $w_i^1 \leftarrow 1$ for each $i \in [N]$
2. For each $t \in [T]$:
   a) Set $p_i^t \leftarrow \frac{w_i^t}{\sum_{j \in [N]} w_j^t}$
   b) Observe the loss vector $\vec{m}^t$
   c) For each $i \in [N]$:
      i. Set $w_i^{t+1} \leftarrow w_i^t \cdot \exp\left(-\varepsilon \cdot m_i^t\right)$

Hedge Algorithm

1. Initialize each $w_i^1 \leftarrow 1$ for each $i \in [N]$
2. For each $t \in [T]$:
   a) Set $p_i^t \leftarrow \frac{w_i^t}{\sum_{j \in [N]} w_j^t}$
   b) Observe the loss vector $\vec{m}^t$
   c) For each $i \in [N]$:
      i. Set $w_i^{t+1} \leftarrow w_i^t \cdot \exp\left(-\varepsilon \cdot m_i^t\right)$

$m_i^t > 0$, **decrease** i's weight; otherwise **increase** i's weight

Show the Expected Loss is Bounded

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{p}^t, \vec{m}^t \rangle \leq \sum_{t\in[T]} m_i^t + \frac{\ln(N)}{\varepsilon} + \varepsilon T$$

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{p}^t, \vec{m}^t \rangle \leq \sum_{t\in[T]} m_i^t + \frac{\ln(N)}{\varepsilon} + \varepsilon T$$

Proof: Define potential function as before $\Phi^t = \sum_{i\in[N]} w_i^t$

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{\boldsymbol{p}}^{t}, \vec{\boldsymbol{m}}^{t} \rangle \leq \sum_{t\in[T]} \boldsymbol{m}_i^t + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Proof: Define potential function as before $\Phi^t = \sum_{i\in[N]} w_i^t$

First note: $\Phi^1 = N$ and $\Phi^{t+1} \geq w_i^{t+1} = \exp\left(-\varepsilon \cdot \sum_{t' \leq t} m_i^{t'}\right)$

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{\boldsymbol{p}}^{t}, \vec{\boldsymbol{m}}^{t} \rangle \leq \sum_{t\in[T]} \boldsymbol{m}_i^t + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Proof: Define potential function as before $\Phi^t = \sum_{i\in[N]} w_i^t$

First note: $\Phi^1 = N$ and $\Phi^{t+1} \geq w_i^{t+1} = \exp\left(-\varepsilon \cdot \sum_{t'\leq t} m_i^{t'}\right)$

Then, $\Phi^{t+1} = \sum_{j\in[N]} w_j^{t+1} = \sum_{j\in[N]} w_j^t \cdot \exp(-\varepsilon \cdot m_j^t)$

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{\boldsymbol{p}}^t, \vec{\boldsymbol{m}}^t \rangle \leq \sum_{t\in[T]} \boldsymbol{m}_i^t + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Proof: Define potential function as before $\Phi^t = \sum_{i\in[N]} w_i^t$

First note: $\Phi^1 = N$ and $\Phi^{t+1} \geq w_i^{t+1} = \exp\left(-\varepsilon \cdot \sum_{t'\leq t} m_i^{t'}\right)$

Then, $\Phi^{t+1} = \sum_{j\in[N]} w_j^{t+1} = \sum_{j\in[N]} w_j^t \cdot \exp\left(-\varepsilon \cdot m_j^t\right)$

Since by the Taylor series $e^x \leq 1 + x + x^2$ for $x \in [-1,1]$,

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{\boldsymbol{p}}^{t}, \vec{\boldsymbol{m}}^{t} \rangle \leq \sum_{t\in[T]} \boldsymbol{m}_i^t + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Proof: Define potential function as before $\Phi^t = \sum_{i\in[N]} w_i^t$

First note: $\Phi^1 = N$ and $\Phi^{t+1} \geq w_i^{t+1} = \exp\left(-\varepsilon \cdot \sum_{t' \leq t} m_i^{t'}\right)$

Then, $\Phi^{t+1} = \sum_{j\in[N]} w_j^{t+1} = \sum_{j\in[N]} w_j^t \cdot \exp(-\varepsilon \cdot m_j^t)$

Since by the Taylor series $e^x \leq 1 + x + x^2$ for $x \in [-1,1]$,

$$\Phi^{t+1} \leq \sum_{j\in[N]} w_j^t \cdot \left(1 - \varepsilon \cdot m_j^t + \varepsilon^2 \left(m_j^t\right)^2\right)$$

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t \in [T]} \langle \vec{\boldsymbol{p}}^t, \vec{\boldsymbol{m}}^t \rangle \leq \sum_{t \in [T]} \boldsymbol{m}_i^t + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Proof: $\Phi^{t+1} \leq \sum_{j \in [N]} w_j^t \cdot \left(1 - \varepsilon \cdot m_j^t + \varepsilon^2 \left(m_j^t\right)^2\right)$

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{\boldsymbol{p}}^t, \vec{\boldsymbol{m}}^t \rangle \leq \sum_{t\in[T]} \boldsymbol{m}_i^t + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Proof:

$$\Phi^{t+1} \leq \textstyle\sum_{j\in[N]} w_j^t \cdot \left(1 - \varepsilon \cdot m_j^t + \varepsilon^2 \left(m_j^t\right)^2\right)$$

$$\leq \sum_{j\in[N]} w_j^t \cdot \left(1 - \varepsilon \cdot m_j^t + \varepsilon^2\right)$$

Since $\left(\boldsymbol{m}_j^t\right)^2 \leq 1$

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{p}^t, \vec{m}^t \rangle \leq \sum_{t\in[T]} m_i^t + \frac{\ln(N)}{\varepsilon} + \varepsilon T$$

Proof:

$$\Phi^{t+1} \leq \sum_{j\in[N]} w_j^t \cdot \left(1 - \varepsilon \cdot m_j^t + \varepsilon^2 \left(m_j^t\right)^2\right)$$

$$\leq \sum_{j\in[N]} w_j^t \cdot \left(1 - \varepsilon \cdot m_j^t + \varepsilon^2\right)$$

$$\leq \sum_{j\in[N]} w_j^t \cdot (1 + \varepsilon^2) - \sum_{j\in[N]} w_j^t \cdot \varepsilon \cdot m_j^t$$

Splitting the equation

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{p}^t, \vec{m}^t \rangle \leq \sum_{t\in[T]} m_i^t + \frac{\ln(N)}{\varepsilon} + \varepsilon T$$

Proof:

$$\Phi^{t+1} \leq \sum_{j\in[N]} w_j^t \cdot \left(1 - \varepsilon \cdot m_j^t + \varepsilon^2 \left(m_j^t\right)^2\right)$$

$$\leq \sum_{j\in[N]} w_j^t \cdot \left(1 - \varepsilon \cdot m_j^t + \varepsilon^2\right)$$

$$\leq \sum_{j\in[N]} w_j^t \cdot (1+\varepsilon^2) - \sum_{j\in[N]} w_j^t \cdot \varepsilon \cdot m_j^t$$

$$\leq \Phi^t (1+\varepsilon^2) - \varepsilon \cdot \sum_{j\in[N]} \Phi^t \cdot p_j^t \cdot m_j^t$$

Since we set $p_j^t = w_j^t / \Phi^t$

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{p}^t, \vec{m}^t \rangle \leq \sum_{t\in[T]} m_i^t + \frac{\ln(N)}{\varepsilon} + \varepsilon T$$

Proof:

$$\Phi^{t+1} \leq \Phi^t (1+\varepsilon^2) - \varepsilon \cdot \sum_{j\in[N]} \Phi^t \cdot p_j^t \cdot m_j^t$$

$$= \Phi^t \cdot \left((1+\varepsilon^2) - \varepsilon \cdot \langle \vec{p}^t, \vec{m}^t \rangle\right)$$

Dot Product

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t \in [T]} \langle \vec{\boldsymbol{p}}^{t}, \vec{\boldsymbol{m}}^{t} \rangle \leq \sum_{t \in [T]} \boldsymbol{m}_i^t + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Proof:

$$\Phi^{t+1} \leq \Phi^t(1+\varepsilon^2) - \varepsilon \cdot \sum_{j \in [N]} \Phi^t \cdot p_j^t \cdot m_j^t$$

$$= \Phi^t \cdot \left((1+\varepsilon^2) - \varepsilon \cdot \langle \vec{p}^t, \vec{m}^t \rangle\right)$$

$$\leq \Phi^t \cdot \exp(\varepsilon^2 - \varepsilon \cdot \langle \vec{p}^t, \vec{m}^t \rangle)$$

Since $1 + x \leq e^x$

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{\boldsymbol{p}}^t, \vec{\boldsymbol{m}}^t \rangle \leq \sum_{t\in[T]} \boldsymbol{m}_i^t + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Proof:

$$\Phi^{t+1} \leq \Phi^t(1+\varepsilon^2) - \varepsilon \cdot \sum_{j\in[N]} \Phi^t \cdot p_j^t \cdot m_j^t$$

$$= \Phi^t \cdot \left((1+\varepsilon^2) - \varepsilon \cdot \langle \vec{p}^t, \vec{m}^t \rangle\right)$$

$$\leq \Phi^t \cdot \exp(\varepsilon^2 - \varepsilon \cdot \langle \vec{p}^t, \vec{m}^t \rangle)$$

$$\leq \Phi^1 \cdot \exp\left(\varepsilon^2 \cdot T - \varepsilon \sum_{t'\leq t} \langle \vec{p}^t, \vec{m}^t \rangle\right)$$

Substituting recursively

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t \in [T]} \langle \vec{p}^t, \vec{m}^t \rangle \leq \sum_{t \in [T]} m_i^t + \frac{\ln(N)}{\varepsilon} + \varepsilon T$$

Proof: Combining upper and lower bounds on Φ^{t+1}

$$\exp\left(-\varepsilon \cdot \sum_{t' \leq t} m_i^t\right) \leq \Phi^{t+1} \leq \Phi^1 \cdot \exp\left(\varepsilon^2 \cdot T - \varepsilon \sum_{t' \leq t} \langle \vec{p}^t, \vec{m}^t \rangle\right)$$

$$-\varepsilon \cdot \sum_{t' \leq t} m_i^t \leq \ln(N) + \varepsilon^2 \cdot T - \varepsilon \sum_{t' \leq t} \langle \vec{p}^t, \vec{m}^t \rangle$$

Take ln of both sides

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{\boldsymbol{p}}^t, \vec{\boldsymbol{m}}^t \rangle \leq \sum_{t\in[T]} \boldsymbol{m}_i^t + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Proof: Combining upper and lower bounds on Φ^{t+1}

$$\exp\left(-\varepsilon \cdot \sum_{t'\leq t} m_i^t\right) \leq \Phi^{t+1} \leq \Phi^1 \cdot \exp\left(\varepsilon^2 \cdot T - \varepsilon \sum_{t'\leq t} \langle \vec{p}^t, \vec{m}^t \rangle\right)$$

$$-\varepsilon \cdot \sum_{t'\leq t} m_i^t \leq \ln(N) + \varepsilon^2 \cdot T - \varepsilon \sum_{t'\leq t} \langle \vec{p}^t, \vec{m}^t \rangle$$

$$\sum_{t'\leq t} \langle \vec{p}^t, \vec{m}^t \rangle \leq \frac{\ln(N)}{\varepsilon} + \varepsilon \cdot T + \sum_{t'\leq t} m_i^t$$

Rearrange

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t \in [T]} \langle \vec{\boldsymbol{p}}^t, \vec{\boldsymbol{m}}^t \rangle \leq \sum_{t \in [T]} \boldsymbol{m}_i^t + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Corollary (Average cost): $\varepsilon \in (0,1], t \in [T], T \geq \frac{4\ln(N)}{\varepsilon^2}$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\frac{1}{\boldsymbol{T}} \cdot \sum_{t \in [T]} \langle \vec{\boldsymbol{p}}^t, \vec{\boldsymbol{m}}^t \rangle \leq \frac{\boldsymbol{1}}{\boldsymbol{T}} \cdot \sum_{t \in [T]} \boldsymbol{m}_i^t + \boldsymbol{2\varepsilon}$$

Theorem: Suppose $\varepsilon \in (0,1]$ and for $t \in [T]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\sum_{t\in[T]} \langle \vec{\boldsymbol{p}}^{\boldsymbol{t}}, \vec{\boldsymbol{m}}^{\boldsymbol{t}} \rangle \leq \sum_{t\in[T]} \boldsymbol{m}_{\boldsymbol{i}}^{\boldsymbol{t}} + \frac{\ln(\boldsymbol{N})}{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon T}$$

Corollary (Average cost): $\varepsilon \in (0,1], t \in [T], T \geq \frac{4\ln(N)}{\varepsilon^2}$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\frac{1}{\boldsymbol{T}} \cdot \sum_{t\in[T]} \langle \vec{\boldsymbol{p}}^{\boldsymbol{t}}, \vec{\boldsymbol{m}}^{\boldsymbol{t}} \rangle \leq \frac{\boldsymbol{1}}{\boldsymbol{T}} \cdot \sum_{t\in[T]} \boldsymbol{m}_{\boldsymbol{i}}^{\boldsymbol{t}} + \boldsymbol{2\varepsilon}$$

Multiply by $\frac{1}{T}$ on both sides and set large enough T to simplify $\frac{\ln(N)}{\varepsilon}$ term

Corollary (Average cost): $\varepsilon \in (0,1], t \in [T], T \geq \frac{4\,\rho^2 \ln(N)}{\varepsilon^2}$, $\boldsymbol{m_i^t} \in [-\boldsymbol{\rho}, \boldsymbol{\rho}]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\frac{1}{\boldsymbol{T}} \cdot \sum_{\boldsymbol{t} \in [\boldsymbol{T}]} \langle \vec{\boldsymbol{p}}^{\boldsymbol{t}}, \vec{\boldsymbol{m}}^{\boldsymbol{t}} \rangle \leq \frac{\mathbf{1}}{\boldsymbol{T}} \cdot \sum_{\boldsymbol{t} \in [\boldsymbol{T}]} \boldsymbol{m_i^t} + \mathbf{2}\boldsymbol{\varepsilon}$$

ρ^2 comes from Taylor expansion

Corollary (Average cost): $\varepsilon \in (0,1], t \in [T], T \geq \frac{4\ln(N)}{\varepsilon^2}$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\frac{1}{\boldsymbol{T}} \cdot \sum_{\boldsymbol{t} \in [\boldsymbol{T}]} \langle \vec{\boldsymbol{p}}^{\boldsymbol{t}}, \vec{\boldsymbol{m}}^{\boldsymbol{t}} \rangle \leq \frac{\mathbf{1}}{\boldsymbol{T}} \cdot \sum_{\boldsymbol{t} \in [\boldsymbol{T}]} \boldsymbol{m_i^t} + \mathbf{2}\boldsymbol{\varepsilon}$$

Solving LPs (Approximately) using MWU

- LPs with m constraints of the form

$$\min c^{\mathrm{T}} x$$
$$\text{s. t. } Ax \geq b$$
$$x \geq 0$$

Solving LPs (Approximately) using MWU

- LPs with m constraints of the form

$$\min c^{\mathrm{T}} x$$
$$\text{s. t. } Ax \geq b$$
$$x \geq 0$$

- Suppose we know $c^{\mathrm{T}} x^* = \mathrm{OPT}$ using **binary search**, find an ε-approximate solution $\tilde{x}$ s.t.

$$c^{\mathrm{T}} \tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

or output **infeasible**

Solving LPs (Approximately) using MWU

- Suppose we know $c^{\mathrm{T}} x^* = \mathrm{OPT}$ using **binary search**, find an ε-approximate solution $\tilde{x}$ s.t.

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

or output **infeasible**

- **Runtime is $O\left(\frac{\rho^2 \log m}{\varepsilon^2}\right)$** where ρ is the **width**

Solving LPs (Approximately) using MWU

- Suppose we know $c^{\mathrm{T}} x^* = \mathrm{OPT}$ using **binary search**, find an ε-approximate solution $\tilde{x}$ s.t.

$$c^{\mathrm{T}}\, \tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon \mathbf{1}$$
$$\tilde{x} \geq 0$$

or output **infeasible**

- **Runtime is $\boldsymbol{O}\left(\frac{\boldsymbol{\rho^2 \log m}}{\boldsymbol{\varepsilon^2}}\right)$** where ρ is the **width**
- Simple convex region: $K = \{x \in \mathbb{R}^n \mid x \geq 0, c^{\mathrm{T}} x = OPT\}$

Solving LPs (Approximately) using MWU

- Suppose we know $c^{\mathrm{T}} x^* = \mathrm{OPT}$ using **binary search**, find an ε-approximate solution $\tilde{x}$ s.t.

$$c^{\mathrm{T}} \tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon \mathbf{1}$$
$$\tilde{x} \geq 0$$

or output **infeasible**

- **Runtime is $O\left(\frac{\rho^2 \log m}{\varepsilon^2}\right)$** where ρ is the **width**
- Simple convex region: $K = \{x \in \mathbb{R}^n \mid x \geq 0, c^{\mathrm{T}} x = OPT\}$
- Need to check for feasibility of $\boldsymbol{A}x \geq \boldsymbol{b}$ in K

Solving LPs (Approximately) using MWU

- Simple convex region: $K = \{x \in \mathbb{R}^n \mid x \geq 0, c^{\mathrm{T}} x = OPT\}$
- Need to check for feasibility of $\boldsymbol{A}x \geq \boldsymbol{b}$ in K

Solving LPs (Approximately) using MWU

- Simple convex region: $K = \{x \in \mathbb{R}^n \mid x \geq 0, c^{\mathrm{T}}x = OPT\}$
- Need to check for feasibility of $\boldsymbol{A}x \geq \boldsymbol{b}$ in K
- Assume we have **an oracle returns $\boldsymbol{x} \in \boldsymbol{K}$** satisfying following:

Solving LPs (Approximately) using MWU

- Simple convex region: $K = \{x \in \mathbb{R}^n \mid x \geq 0, c^{\mathrm{T}}x = OPT\}$
- Need to check for feasibility of $\boldsymbol{A}x \geq \boldsymbol{b}$ in K
- Assume we have **an oracle returns $\boldsymbol{x} \in \boldsymbol{K}$** satisfying following:
  - **Either** $\boldsymbol{w}^t \cdot \boldsymbol{A}x \geq \boldsymbol{w}^t \cdot \boldsymbol{b}$ for convex combination of constraints using vector $\boldsymbol{w}$

Solving LPs (Approximately) using MWU

- Simple convex region: $K = \{x \in \mathbb{R}^n \mid x \geq 0, c^{\mathrm{T}}x = OPT\}$
- Need to check for feasibility of $\boldsymbol{A}x \geq \boldsymbol{b}$ in K
- Assume we have **an oracle returns $\boldsymbol{x} \in \boldsymbol{K}$** satisfying following:
  - **Either** $\boldsymbol{w}^t \cdot \boldsymbol{A}x \geq \boldsymbol{w}^t \cdot \boldsymbol{b}$ for convex combination of constraints using vector $\boldsymbol{w}$
  - **Or infeasible (no such x)**

Solving LPs (Approximately) using MWU

- Simple convex region: $K = \{x \in \mathbb{R}^n \mid x \geq 0, c^{\mathrm{T}}x = OPT\}$
- Need to check for feasibility of $\boldsymbol{A}x \geq \boldsymbol{b}$ in K
- Assume we have **an oracle returns $\boldsymbol{x} \in \boldsymbol{K}$** satisfying following:
  - **Either** $\boldsymbol{w}^{\boldsymbol{t}} \cdot \boldsymbol{A}x \geq \boldsymbol{w}^{\boldsymbol{t}} \cdot \boldsymbol{b}$ for convex combination of constraints using vector $\boldsymbol{w}$
  - **Or infeasible (no such $\boldsymbol{x}$)**
- **Using oracle and MWU show:**
  - For a particular ``guess” of OPT using binary search, the solution is feasible or infeasible (and take the smallest feasible ``guess”)

Solving LPs (Approximately) using MWU

- Each **constraint $\alpha^{\mathrm{T}}x \geq \beta$ is an expert**
- Total of m experts

Solving LPs (Approximately) using MWU

- Each **constraint $\alpha^{\mathrm{T}} x \geq \beta$ is an expert**
- Total of $\boldsymbol{m}$ experts
- Another way to look at it:

Solving LPs (Approximately) using MWU

- Each **constraint $\alpha^{\mathrm{T}} x \geq \beta$ is an expert**
- Total of m experts
- Another way to look at it use oracle, find x using oracle using old weights, then find new weights w:

Solving LPs (Approximately) using MWU

- Each **constraint $\alpha^{\mathrm{T}} x \geq \beta$ is an expert**
- Total of m experts
- Another way to look at it use oracle, find x using oracle using old weights, then find new weights w:

Theorem: **If there were a solution to $Ax \geq b, x \in K$, then there is a solution to $w^t \cdot Ax \geq w^t \cdot b$. Contrapositive gives infeasibility.**

Solving LPs (Approximately) using MWU

- Each **constraint $\alpha^{\mathrm{T}}x \geq \beta$ is an expert**
- Total of m experts
- Another way to look at it use oracle, find x using oracle using old weights, then find new weights w:

Theorem: **If there were a solution to $Ax \geq b, x \in K$, then there is a solution to $w^t \cdot Ax \geq w^t \cdot b$. Contrapositive gives infeasibility.**

Finds a set of non-negative weights certifying infeasibility

Finds an approximate solution certifying $a_j \cdot x - b_j \geq -\varepsilon$

Solving LPs (Approximately) using MWU

$$
\begin{aligned}
c^{\mathrm{T}}\, \tilde{x} &= \mathrm{OPT} \\
A\tilde{x} &\geq b - \varepsilon \mathbf{1} \\
\tilde{x} &\geq 0
\end{aligned}
$$

- Either:
  - **Finds a set of non-negative weights certifying infeasibility**
  - **Finds an approximate solution certifying $a_j \cdot x - b_j \geq -\varepsilon$**
- **Conditions are not necessarily disjoint**

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \ \mathrm{OPT}$$
$$A\tilde{x} \geq b\ - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- m experts, one per row

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\, \tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon \mathbf{1}$$
$$\tilde{x} \geq 0$$

- m experts, one per row
- weight w_j^t denotes weight of expert at time $t \in [T]$
  - At time $t = 1$, all weights are 1

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- m experts, one per row
- weight w_j^t denotes weight of expert at time $t \in [T]$
  - At time $t = 1$, all weights are 1
- Use ***oracle*** to solve $\boldsymbol{w^t} \cdot \boldsymbol{A}x \geq \boldsymbol{w^t} \cdot \boldsymbol{b}$ at each time t

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- m experts, one per row
- weight w_j^t denotes weight of expert at time $t \in [T]$
  - At time $t = 1$, all weights are 1
- Use ***oracle*** to solve $\boldsymbol{w^t} \cdot \boldsymbol{Ax} \geq \boldsymbol{w^t} \cdot \boldsymbol{b}$ at each time t
  - If no solution, **halt** and output infeasible

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- m experts, one per row
- weight w_j^t denotes weight of expert at time $t \in [T]$
  - At time $t = 1$, all weights are 1
- Use ***oracle*** to solve $\boldsymbol{w^t} \cdot \boldsymbol{Ax} \geq \boldsymbol{w^t} \cdot \boldsymbol{b}$ at each time t
  - If no solution, **halt** and output infeasible
  - Otherwise, take solution x^t to impose cost $m_j^t = a_i \cdot x^t - b_j$

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** or solve system at each time t
  - If no solution, **halt** and output infeasible
  - Otherwise, take solution x^t to impose cost $m_j^t = a_i \cdot x^t - b_j$
- <u>Why use this cost?</u>

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \ \mathrm{OPT}$$
$$A\tilde{x} \geq b\ - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** or solve system at each time t
  - If no solution, **halt** and output infeasible
  - Otherwise, take solution x^t to impose cost $m_j^t = a_i \cdot x^t - b_j$
- <u>Why use this cost?</u>
  - Whenever $a_i \cdot x^t - b_j > 0$, we have ``oversatisfied'' the constraint

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** or solve system at each time t
  - If no solution, **halt** and output infeasible
  - Otherwise, take solution x^t to impose cost $m_j^t = a_i \cdot x^t - b_j$
- <u>Why use this cost?</u>
  - Whenever $a_i \cdot x^t - b_j > 0$, we have ``oversatisfied'' the constraint
  - Reduce weight of constraint next round

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** or solve system at each time t
  - If no solution, **halt** and output infeasible
  - Otherwise, take solution x^t to impose cost $m_j^t = a_i \cdot x^t - b_j$
- <u>Why use this cost?</u>
  - Whenever $a_i \cdot x^t - b_j > 0$, we have ``oversatisfied'' the constraint
  - Reduce weight of constraint next round; otherwise, increase

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** or solve system at each time t
  - If no solution, **halt** and output infeasible
  - Otherwise, take solution x^t to impose cost $m_j^t = a_i \cdot x^t - b_j$
- <u>Why use this cost?</u>
  - Whenever $a_i \cdot x^t - b_j > 0$, we have ``oversatisfied'' the constraint
  - Reduce weight of constraint next round; otherwise, increase

Recall Hedge Algorithm

1. Initialize each $w_i^1 \leftarrow 1$ for each $i \in [N]$
2. For each $t \in [T]$:
   a) Set $p_i^t \leftarrow \frac{w_i^t}{\sum_{j \in [N]} w_j^t}$
   b) Observe the loss vector $\vec{m}^t$
   c) For each $i \in [N]$:
      i. Set $w_i^{t+1} \leftarrow w_i^t \cdot \exp\left(-\varepsilon \cdot m_i^t\right)$

$m_i^t > 0$, **decrease** i's weight; otherwise **increase** i's weight

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** or solve system at each time t
    - If no solution, **halt** and output infeasible
    - Otherwise, take solution x^t to impose cost $m_j^t = a_i \cdot x^t - b_j$
- **Update weights using Hedge algorithm**

Solving LPs (Approximately) using MWU

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** or solve system at each time t
  - If no solution, **halt** and output infeasible
  - Otherwise, take solution x^t to impose cost $m_j^t = a_i \cdot x^t - b_j$
- **Update weights using Hedge algorithm**
- If after T rounds (we'll define T), the solution is non-negative, then return $\overline{x} = \frac{1}{T} \cdot \sum_{t \in [T]} x^t$

Solving LPs (Approximately) using MWU

- Runtime and cost?

Solving LPs (Approximately) using MWU

- Runtime and cost?

Corollary (Average cost): $\varepsilon \in (0,1], t \in [T], T \geq \frac{4\,\rho^2 \ln(N)}{\varepsilon^2}, m_i^t \in [-\rho, \rho]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\frac{1}{T} \cdot \sum_{t \in [T]} \langle \vec{p}^t, \vec{m}^t \rangle \leq \frac{1}{T} \cdot \sum_{t \in [T]} m_i^t + 2\varepsilon$$

Solving LPs (Approximately) using MWU

- Runtime and cost?

Corollary (Average cost): $\varepsilon \in (0,1], t \in [T], T \geq \frac{4\,\rho^2 \ln(N)}{\varepsilon^2}, m_i^t \in [-\rho, \rho]$, then

Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\frac{1}{\boldsymbol{T}} \cdot \sum_{\boldsymbol{t} \in [\boldsymbol{T}]} \langle \vec{\boldsymbol{p}}^{\boldsymbol{t}}, \vec{\boldsymbol{m}}^{\boldsymbol{t}} \rangle \leq \frac{\mathbf{1}}{\boldsymbol{T}} \cdot \sum_{\boldsymbol{t} \in [\boldsymbol{T}]} \boldsymbol{m}_{\boldsymbol{i}}^{\boldsymbol{t}} + \mathbf{2}\boldsymbol{\varepsilon}$$

- Determine $\rho = \max_{j,x,t}\left\{1, \left|a_j \cdot x^t - b_j\right|\right\}$ (maximum cost at any round)
- Get , $T \geq \frac{4\,\rho^2 \ln(N)}{\varepsilon^2}$ using corollary and substitute $\vec{w}$ for $\vec{p}$

Solving LPs (Approximately) using MWU

- Runtime and cost?

Corollary (Average cost): $\varepsilon \in (0,1], t \in [T], T \geq \frac{4\,\rho^2 \ln(N)}{\varepsilon^2}, m_i^t \in [-\rho, \rho]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\frac{1}{T} \cdot \sum_{t \in [T]} \langle \vec{p}^t, \vec{m}^t \rangle \leq \frac{1}{T} \cdot \sum_{t \in [T]} m_i^t + 2\varepsilon$$

- Get $, T \geq \frac{4\,\rho^2 \ln(N)}{\varepsilon^2}$ using corollary and substitute $\vec{w}$ for $\vec{p}$
- $\frac{1}{T} \cdot \sum_{t \in T} m_i^t + 2\varepsilon = \frac{1}{T} \cdot \sum_{t \in T} \left(a_j \cdot x^t - b_j\right) + 2\varepsilon = a_j \cdot \overline{x} - b_j + 2\varepsilon$

Solving LPs (Approximately) using MWU

- Runtime and cost?

Corollary (Average cost): $\varepsilon \in (0,1], t \in [T], T \geq \frac{4\,\rho^2 \ln(N)}{\varepsilon^2}, m_i^t \in [-\rho, \rho]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\frac{1}{T} \cdot \sum_{t \in [T]} \langle \vec{p}^t, \vec{m}^t \rangle \leq \frac{\mathbf{1}}{\boldsymbol{T}} \cdot \sum_{\boldsymbol{t} \in [\boldsymbol{T}]} \boldsymbol{m}_{\boldsymbol{i}}^{\boldsymbol{t}} + 2$$

$$a_j \cdot \overline{x} - b_j + 2\varepsilon \geq 0$$

means

$$a_j \cdot \overline{x} \geq b_j - 2\varepsilon$$

- Get , $T \geq \frac{4\,\rho^2 \ln(N)}{\varepsilon^2}$ using corollary and substitute $\vec{w}$ for $\vec{p}$
- $\frac{1}{T} \cdot \sum_{t \in T} m_i^t + 2\varepsilon = \frac{1}{T} \cdot \sum_{t \in T} \left(a_j \cdot x^t - b_j\right) + 2\varepsilon = a_j \cdot \overline{x} - b_j + 2\varepsilon$

Last Time…

Solving LPs (Approximately) using MWU

$$\min c^{\mathrm{T}} x \quad \text{s.t. } Ax \geq b \quad x \geq 0$$

Binary Search for OPT

$$c^{\mathrm{T}} \tilde{x} = \mathrm{OPT} \quad A\tilde{x} \geq b - \varepsilon \mathbf{1} \quad \tilde{x} \geq 0$$

Solving LPs (Approximately) using MWU

$$\min c^{\mathrm{T}} x$$
$$\text{s. t. } Ax \geq b$$
$$x \geq 0$$

Binary Search for OPT

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** to solve convex combination $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{b}$ at each time t where **$\boldsymbol{w}$ is weight vector, initially all 1~~s~~** m

Solving LPs (Approximately) using MWU

$$\min c^{\mathrm{T}} x$$
$$\text{s.t. } Ax \geq b$$
$$x \geq 0$$

Binary Search for OPT

$$c^{\mathrm{T}} \tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon \mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** to solve convex combination $\boldsymbol{w}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{x} \geq \boldsymbol{w}^{\mathrm{T}} \boldsymbol{b}$ at each time t where $\boldsymbol{w}$ **is weight vector, initially all ~~1s~~** m

<u>Theorem</u>: If there were a solution to $\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}, \boldsymbol{x} \in \boldsymbol{K}$, then there is a solution to $\boldsymbol{w}^{t} \cdot \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{w}^{t} \cdot \boldsymbol{b}$. Contrapositive gives infeasibility.

Solving LPs (Approximately) using MWU

$$\min c^{\mathrm{T}}x$$
$$\text{s.t. } Ax \geq b$$
$$x \geq 0$$

Binary Search for OPT

$$c^{\mathrm{T}}\,\tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon\mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** to solve convex combinatio time t where **w is weight vector, initia**

Theorem: **If there were a solution to $\boldsymbol{Ax \geq b, x \in K}$, then there is a solution to $\boldsymbol{w^t \cdot Ax \geq w^t \cdot b}$. Contrapositive gives infeasibility.**

Finds a set of non-negative weights certifying infeasibility

Finds an approximate solution certifying $\boldsymbol{a_j \cdot x - b_j \geq -\varepsilon}$

Solving LPs (Approximately) using MWU

$$\min c^{\mathrm{T}} x$$
$$\text{s.t. } Ax \geq b$$
$$x \geq 0$$

Binary Search for OPT

$$c^{\mathrm{T}} \tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon \mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** to solve convex combination $\boldsymbol{w}^{\mathrm{T}} \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{w}^{\mathrm{T}} \boldsymbol{b}$ at each time t where $\boldsymbol{w}$ **is weight vector, initially all 1s**
  - If no solution, **halt** and output infeasible
  - Otherwise, take solution x^t to impose cost $m_j^t = a_i \cdot x^t - b_j$

Solving LPs (Approximately) using MWU

$$\min c^{\mathrm{T}} x$$
$$\text{s.t. } Ax \geq b$$
$$x \geq 0$$

Binary Search for OPT

$$c^{\mathrm{T}} \tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b - \varepsilon \mathbf{1}$$
$$\tilde{x} \geq 0$$

- Use ***oracle*** to solve convex combination $\boldsymbol{w}^{\mathrm{T}} \boldsymbol{A x} \geq \boldsymbol{w}^{\mathrm{T}} \boldsymbol{b}$ at each time t where **$\boldsymbol{w}$ is weight vector, initially all 1s**
  - If no solution, **halt** and output infeasible
  - Otherwise, take solution x^t to impose cost $m_j^t = a_i \cdot x^t - b_j$
- **Use Hedge algorithm to update**

Solving LPs (Approximately) using MWU

- Runtime and cost?

Corollary (Average cost): $\varepsilon \in (0,1], t \in [T], T \geq \frac{4\,\rho^2 \ln(N)}{\varepsilon^2}, m_i^t \in [-\rho, \rho]$, then Hedge returns a probability distribution where for any expert $i \in [N]$,

$$\frac{1}{\boldsymbol{T}} \cdot \sum_{t \in [T]} \langle \vec{\boldsymbol{p}}^t, \vec{\boldsymbol{m}}^t \rangle \leq \frac{\mathbf{1}}{\boldsymbol{T}} \cdot \sum_{t \in [T]} \boldsymbol{m}_i^t + \mathbf{2}\boldsymbol{\varepsilon}$$

- Get $T \geq \frac{4\,\rho^2 \ln(N)}{\varepsilon^2}$ using corollary and substitute $\vec{w}$ for $\vec{p}$

Solving LPs (Approximately) using MWU

- Analysis:
  - $\frac{1}{T} \cdot \sum_{t\in[T]} \langle \vec{p}^t, \vec{m}^t \rangle = \langle \vec{p}^t, Ax^t \; - b \rangle = w^{\mathrm{T}} Ax - w^{\mathrm{T}} \cdot b \geq 0$
  - $\frac{1}{T} \cdot \sum_{t\in[T]} m_i^t + 2\varepsilon = \frac{1}{T} \cdot \sum_{t\in T} \left(a_j \cdot x^t \; - b_j\right) + 2\varepsilon = a_j \cdot \overline{x} \; - b_j + 2\varepsilon$
- Putting it together:
  - $a_j \cdot \overline{x} \; - b_j + 2\varepsilon \geq 0$
- Satisfies:

$$c^{\mathrm{T}} \tilde{x} = \mathrm{OPT}$$
$$A\tilde{x} \geq b \; - \varepsilon' \mathbf{1}$$
$$\tilde{x} \geq 0$$

Packing and Covering LPs

- **Covering LPs:**
  - If the constraint matrix A is all positive

Packing and Covering LPs

- **Covering LPs:**
  - If the constraint matrix A is all positive, i.e. $Ax \geq 1$
  - Put enough weight on x to **cover** every constraint

Packing and Covering LPs

- **Covering LPs:**
  - If for an all positive constraint matrix A: $Ax \geq b$
  - Put enough weight on x to **cover** every constraint
- **Packing LPs:**
  - If for an all positive matrix constraint: $Ax \leq b$
  - **Packing** as much into x as possible without violating any constraint

Packing and Covering LPs

- **Covering LPs:**
  - If for an all positive constraint matrix A: $Ax \geq b$
  - Put enough weight on x to **cover** every constraint
- **Packing LPs:**
  - If for an all positive matrix constraint: $Ax \leq b$
  - **Packing** as much into x as possible without violating any constraint
- Packing LPs, just flip the feasibility constraint for the oracle:
  - $p^T A x \leq p^T b$

Example Applications: Densest Subgraph

- Problem Definition:

Densest Subgraph: Given a graph $G = (V, E)$, find a subset of vertices that maximizes $\max_{S \subseteq V} \left(\frac{E(S)}{V(S)} \right)$ the density of the induced subgraph on S.