Generative Adversarial Networks (GAN) and related mathematics

A/Prof Richard Yi Da Xu
richardxu.com

University of Technology Sydney (UTS)

February 14, 2021

Content

1. Traditional GAN
2. Mathematics on W-GAN
3. Duality and KKT conditions
4. info-GAN
5. Bayesian GAN

This lecture is referenced heavily from:

- https://vincentherrmann.github.io/blog/wasserstein/
- https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html
- https://towardsdatascience.com/infogan-generative-adversarial-networks-part-iii-380c0c6712cd
- http://www.math.ubc.ca/~israel/m340/kkt2.pdf
- https://spaces.ac.cn/archives/6280
- https://spaces.ac.cn/archives/6051
- https://arxiv.org/pdf/1705.09558.pdf

Traditional GAN

GAN Objective

- look at GAN objective:

$$\min_G \max_D \left(\mathcal{L}(D, G) \equiv \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \right)$$
$$= \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[\log D(\mathbf{x})] \underbrace{+\mathbb{E}_{x \sim p_g(x)}[\log(1 - D(\mathbf{x})]}_{\text{alternative expression}}$$

- note that only $p_g(x)$ is parameterized, you can **not** learn $p_r(\mathbf{x})$
- **tradtional view of** D: D maximize the difference between $p_r(\mathbf{x})$ and $p_g(\mathbf{x})$, and G minimize the difference between $p_r(\mathbf{x})$ and $p_g(\mathbf{x})$
- **critic view of** D: D gives a critic between $p_r(\mathbf{x})$ and $p_g(\mathbf{x})$ in terms the largest of their distance (i.e, the most strict critic/judge), by maximize the difference measure between p_r and p_g
  G tries to make it better $\big(p_g(\mathbf{x})$ to look like $p_r(\mathbf{x})\big)$ using the current measure
  moral of story: D presents a way to measure between p_r and p_g, i.e., some kind of divergence

$$\left(\max_D \left(\mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{x \sim p_g(x)}[\log(1 - D(\mathbf{x})]\right) \right) \quad \text{gives the strictest critic!}$$

- be careful of the signs:
- using $-\log(D)$ trick: $\mathcal{L}(D,G) \approx \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log D(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim p_g}[\log(D(G(.)))]$:
- let $U(\mathbf{x}) \equiv -\log D(\mathbf{x})$ and to fix G: (comes later for Energy GAN representation)

$$
\begin{aligned}
D^* &= \arg\max_D \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[-U] - \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}[-U] \\
&= \arg\max_D -\mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[U(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}[U(\mathbf{x})] \\
&= \arg\min_D \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[U(\mathbf{x})] - \mathbb{E}_{x\sim p_g(\mathbf{x})}[U(\mathbf{x})]
\end{aligned}
$$

change the variable $D \to U$:

$$
U^* = \arg\min_U \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[U(\mathbf{x})] - \mathbb{E}_{z\sim q(z)}[U(G(z))]
$$

CONs of KL divergence

$$\text{KL}(p\|q) = \int_x p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$$

- in cases where $p(x) \rightarrow 0$, but $q(x) >> 0$, effect of q(x) is disregarded
  1. p = 0.000001; q = 0.999999; print p* np.log(p/q): -1.3815509557963774e-05
  2. p = 0.000001; q = 0.100000; print p* np.log(p/q): -1.1512925464970228e-05

PROs of KL divergence

- we try to find q as a proposal distribution for π
- it may turn into a **PRO** when finding approximations for $\pi(x)$ by proposal $q(x)$ by minimizing their KL:

$$\text{KL}(q\|\pi) = \int_x q(x) \log \frac{q(x)}{\pi(x)} \text{d}x \qquad \text{note the order of } \pi \text{ and } q$$

- make sure any x very *improbable* to be drawn from $\pi(x)$ would also be very *improbable* to be drawn from $q(x)$:
  1. when $q(x) >> 0$ AND $\pi(x) \rightarrow 0 \implies$ KL $\rightarrow$ high:
     prevents draw samples where $\pi(x)$ is low prohibitive
     pi = 0.000001; q = 0.999999; print q* np.log(q/pi): 13.81549574245421
  2. when $q(x) \rightarrow 0$ AND $\pi(x) >> 0 \implies$ KL $\rightarrow 0$:
     prevents draw samples where $\pi(x)$ is high more forgiven
     pi = 0.999999; q = 0.000001; print q* np.log(q/pi): -1.3815509557963774e-0

KL divergence for GAN setting

- **same** as previous page, we change $q \to p_g$, and $\pi \to p_r$:

$$\mathrm{KL}(p_g \| p_r) = \int_{\mathbf{x}} p_g \log \frac{p_g(\mathbf{x})}{p_r(\mathbf{x})} d\mathbf{x}$$

1. when $p_g(\mathbf{x}) >> 0$ AND $p_r(\mathbf{x}) \to 0 \implies \mathrm{KL} \to$ high:
   prohibitive for Generator to generate "unreal" image (p_r is low)
   pr = 0.000001; pg = 0.999999; print pg* np.log(pg/pr): 13.81549574245421
   consequence Generator generate **less diverse** samples
   may lead towards mode collapse

2. when $p_g(\mathbf{x}) \to 0$ AND $p_r(\mathbf{x}) >> 0 \implies \mathrm{KL} \to 0$:
   more forgiven if Generator unable to generate "real" samples (p_r is high)
   pr = 0.999999; pg = 0.000001; print pg* np.log(pg/pr): -1.3815509557963774e-0

- another reason why KL divergnece isn't great for GAN's critic!

- **JS divergence**:

$$\text{JS}(p\|q) = \frac{1}{2}\text{KL}\left(p\middle\|\frac{p+q}{2}\right) + \frac{1}{2}\text{KL}\left(q\middle\|\frac{p+q}{2}\right)$$

Find optimal D^* after fixed G (part 1)

- fix G first:

$$\min_G \max_D \mathcal{L}(D,G) = \underbrace{\mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}[\log(1-D(\mathbf{x}))]}_{\mathcal{L}(G,D)}$$

$$\implies \mathcal{L}(G,D) = \int_{\mathbf{x}} \Big(\underbrace{p_r(\mathbf{x})\log(D(\mathbf{x})) + p_g(\mathbf{x})\log(1-D(\mathbf{x}))}_{F(x,D(x))} \Big) \mathrm{d}\mathbf{x}$$

- look at functional $J = \int_{\mathbf{x}} \Big(\underbrace{p_r(\mathbf{x})\log(D(\mathbf{x})) + p_g(\mathbf{x})\log(1-D(\mathbf{x}))}_{F(x,D(x))} \Big) \mathrm{d}\mathbf{x}$:
- Euler Lagrange says: to find stationary function **f** of functional F:

$$\int_a^b F(x, \mathbf{f}(x), \mathbf{f}'(x))\, \mathrm{d}x$$

- then **f** of a real argument x, a stationary point of the functional F when:

$$\frac{\partial F}{\partial \mathbf{f}} - \frac{\mathrm{d}}{\mathrm{d}x}\frac{\partial F}{\partial \mathbf{f}'} = 0$$

- in our case, we have x and $\mathbf{f} \equiv D(x)$ and **not** have $D'(x)$:

$$\frac{\partial F}{\partial D(x)} = 0$$

Find optimal D^* after fixed G (part 2)

- let: $J = \int_x \Big(\underbrace{p_r(x)\log(D(x)) + p_g(x)\log(1 - D(x))}_{F(x,D(x))} \Big) \mathrm{d}x$

$$
\begin{aligned}
F(x, D(x)) &= p_r(x)\log D(x) + p_g(x)\log(1 - D(x)) \\
\frac{\partial F(x, D(x))}{\partial D(x)} &= p_r(x)\frac{1}{D(x)} - p_g(x)\frac{1}{1 - D(x)} = \left(\frac{p_r(x)}{D(x)} - \frac{p_g(x)}{1 - D(x)}\right) \\
&= \frac{p_r(x) - (p_r(x) + p_g(x))D(x)}{D(x)(1 - D(x))}
\end{aligned}
$$

- Let $\frac{\mathrm{d}F(x,D(x))}{\mathrm{d}D(x)} = 0$:

$$
\begin{aligned}
\frac{p_r(x) - \big(p_r(x) + p_g(x)\big)D(x)}{D(x)(1 - D(x))} &= 0 \\
\Longrightarrow \quad p_r(x) - \big(p_r(x) + p_g(x)\big)D(x) &= 0 \\
D^*(x) &= \frac{p_r(x)}{p_r(x) + p_g(x)}
\end{aligned}
$$

- can be thought of as $p(z|x)$ in mixture density. visualize 1-d diagram

substitute Optimal $D^* = \frac{p_r(x)}{p_r(x)+p_g(x)}$ into $\mathcal{L}$:

- substitute $D^*(\mathbf{x}) = \frac{p_r(\mathbf{x})}{p_r(\mathbf{x})+p_g(\mathbf{x})}$:

$$\begin{aligned}\mathcal{L}(G, D^*) &= \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log D^*(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}[\log(1 - D^*(\mathbf{x})]\\ &= \underbrace{\mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\log \frac{p_r(\mathbf{x})}{p_r(\mathbf{x}) + p_g(\mathbf{x})}\right] + \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}\left[\log\left(1 - \frac{p_r(\mathbf{x})}{p_r(\mathbf{x}) + p_g(\mathbf{x})}\right)\right]}\end{aligned}$$

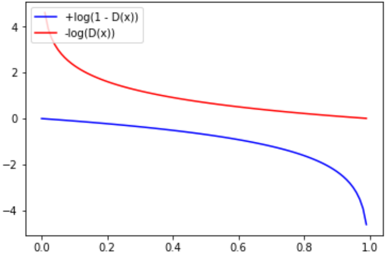
- A better way to find its relationship with JS divergence:

$$\begin{aligned}\mathrm{JS}(p_r \| p_g) &= \frac{1}{2}\mathrm{KL}\left(p_r \Big\| \frac{p_r + p_g}{2}\right) + \frac{1}{2}\mathrm{KL}\left(p_g \Big\| \frac{p_r + p_g}{2}\right)\\ &= \frac{1}{2}\left(\int_{\mathbf{x}} p_r(\mathbf{x}) \log \frac{p_r(\mathbf{x})}{\frac{p_r(\mathbf{x})+p_g(\mathbf{x})}{2}} dx\right) + \frac{1}{2}\left(\int_{\mathbf{x}} p_g(\mathbf{x}) \log \frac{p_g(\mathbf{x})}{\frac{p_r(\mathbf{x})+p_g(\mathbf{x})}{2}} \mathrm{d}\mathbf{x}\right)\\ &= \frac{1}{2}\left(\log 2 + \int_{\mathbf{x}} p_r(\mathbf{x}) \log \frac{p_r(\mathbf{x})}{p_r + p_g(\mathbf{x})} \mathrm{d}\mathbf{x}\right) + \frac{1}{2}\left(\log 2 + \int_{\mathbf{x}} p_g(\mathbf{x}) \log \frac{p_g(x)}{p_r + p_g(x)} \mathrm{d}\mathbf{x}\right)\\ &= \frac{1}{2}\left(\log 4 + \underbrace{\int_{\mathbf{x}} p_r(\mathbf{x}) \log \frac{p_r(\mathbf{x})}{p_r + p_g(\mathbf{x})} \mathrm{d}\mathbf{x} + \int_{\mathbf{x}} p_g(\mathbf{x}) \log \frac{p_g(x)}{p_r + p_g(x)} \mathrm{d}\mathbf{x}}\right)\\ &= \frac{1}{2}\left(\log 4 + \mathcal{L}(G, D^*)\right)\end{aligned}$$

$$\Longrightarrow \mathcal{L}(G, D^*) = 2\mathrm{JS}(p_r \| p_g) - 2\log 2$$

- $\mathcal{L}(D, G)$ can be approximated by:

$$\begin{aligned}\mathcal{L}(D,G) &= \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}[\log(1 - D(\mathbf{x})]\\ &\approx \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}[-\log(D(\mathbf{x}))]\\ &\approx \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log D(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}[\log(D(\mathbf{x}))]\end{aligned}$$



substitute D^* in $-\log(D)$ trick

$$
\mathcal{L}(G, D^*) \equiv \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log D^*(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}[\log(1 - D^*(\mathbf{x})] = 2\mathrm{JS}(p_r \| p_g) - 2\log 2
$$

$$
\Longrightarrow \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}[\log(1 - D^*(\mathbf{x})] = -\mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log D^*(\mathbf{x})] + 2\mathrm{JS}(p_r \| p_g) - 2\log 2
$$

- see how we can put KL into the picture:

$$
\begin{aligned}
\mathrm{KL}(p_g \| p_r) &= \mathbb{E}_{\mathbf{x}\sim p_g}\left[\log \frac{p_g(\mathbf{x})}{p_r(\mathbf{x})}\right] \\
&= \mathbb{E}_{\mathbf{x}\sim p_g}\left[\log \frac{p_g(\mathbf{x})}{p_r(\mathbf{x})}\right] = \mathbb{E}_{\mathbf{x}\sim p_g}\left[\log \frac{\frac{p_g(\mathbf{x})}{p_r(\mathbf{x})+p_g(\mathbf{x})}}{\frac{p_r(\mathbf{x})}{p_r(\mathbf{x})+p_g(\mathbf{x})}}\right] \\
&= \mathbb{E}_{\mathbf{x}\sim p_g}\left[\log \frac{1 - D^*(\mathbf{x})}{D^*(\mathbf{x})}\right] \\
&= \mathbb{E}_{\mathbf{x}\sim p_g}\left[\log(1 - D^*(\mathbf{x}))\right] - \mathbb{E}_{\mathbf{x}\sim p_g}\left[D^*(\mathbf{x})\right] \\
\Longrightarrow \mathbb{E}_{\mathbf{x}\sim p_g}[-D^*(\mathbf{x})] &= \mathrm{KL}(p_g \| p_r) - \mathbb{E}_{\mathbf{x}\sim p_g}\left[\log(1 - D^*(\mathbf{x}))\right] \\
&= \mathrm{KL}(p_g \| p_r) - \mathbb{E}_{\mathbf{x}\sim p_g}\left[\log(1 - D^*(\mathbf{x}))\right] \\
&= \mathrm{KL}(p_g \| p_r) + \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log D^*(\mathbf{x})] - 2\mathrm{JS}(p_r \| p_g) + 2\log 2
\end{aligned}
$$

substitute D^* in $-\log(D)$ trick

- see how it works with $-\log(D)$ trick:

$$
\begin{aligned}
\mathbb{E}_{\mathbf{x}\sim p_g}[-D^*(\mathbf{x})] &= \underbrace{\mathrm{KL}(p_g \| p_r) - 2\mathrm{JS}(p_r \| p_g)}_{\text{depends on } p_g} + \underbrace{2\log 2 + \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log D^*(\mathbf{x})]}_{\text{not depend on } p_g} \\
&\propto \mathrm{KL}(p_g \| p_r) - 2\mathrm{JS}(p_r \| p_g)
\end{aligned}
$$

- using $-\log(D)$ trick as objective, optimize G after fixing D^* is hard!

What is the Optimal $\mathcal{L}$ when have both G^* and D^*

- knowing $D^*(x) = \frac{p_r(x)}{p_r(x)+p_g(x)}$, then optimal $p_g^{\theta^*}(x)$ is when it becomes identifical to $p_r(x)$:
- from previous page:

$$\begin{aligned}
\mathcal{L}(G, D^*) =& 2\text{JS}(p_r \| p_g) - 2\log 2 \\
\implies \mathcal{L}(G^*, D^*) =& \min\left(2\text{JS}(p_r \| p_g) - 2\log 2\right) \\
=& -2\log 2
\end{aligned}$$

- Given distributions P and Q of two vertical bars:

$$
\begin{aligned}
P: &\quad x = 0 && \qquad y \sim U(0,1) \\
Q: &\quad x = \theta, 0 \leq \theta \leq 1 && \qquad y \sim U(0,1)
\end{aligned}
$$

- it turns out the distances are:

$$\mathrm{KL}(P\|Q) = \underbrace{\sum_{x=0,\, y\in(0,1)}}_{\forall (x,y) P(x,y)>0} \underbrace{1}_{P(x,y)} \cdot \log \frac{\overbrace{1}^{P(x,y)}}{\underbrace{0}_{Q(x,y)}} = +\infty$$

$$\mathrm{KL}(Q\|P) = \underbrace{\sum_{x=\theta,\, y\in(0,1)}}_{\forall (x,y) Q(x,y)>0} \underbrace{1}_{Q(x,y)} \cdot \log \frac{\overbrace{1}^{Q(x,y)}}{\underbrace{0}_{P(x,y)}} = +\infty$$

$$D_{JS}(P,Q) = \frac{1}{2}\left(\sum_{x=0,y\in U(0,1)} \underbrace{1}_{P(x,y)} \cdot \log \frac{\overbrace{P(x,y)}^{1}}{\underbrace{1/2}_{\frac{P(x,y)+Q(x,y)}{2}}} + \sum_{x=\theta,y\in U(0,1)} \underbrace{1}_{Q(x,y)} \cdot \log \frac{\overbrace{Q(x,y)}^{1}}{\underbrace{1/2}_{\frac{P(x,y)+Q(x,y)}{2}}} \right)$$

$$= \log 2$$

Wasserstein-GAN

$$\min_{G} \left[\underbrace{\max_{f,\, \|f\|_L \leq 1} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})}[f(G_\theta(\mathbf{z}))]}_{\text{critic}} \right]$$

- it's pretty intuitive to see what **critic** objective does
- so why the heck we call it "Wasserstein-GAN"?
- Because discriminator/critic can be proven to be dual of Wasserstein Distance! - we prove it the other way around from primal $\rightarrow$ dual
- and it turns out that:

$$\mathcal{W}(P, Q) = |\theta|$$

it doesn't have the "zero-jump" effect like KL or JS distance

Wasserstein GAN and Earth Mover Distance

- Wasserstein distances between p_{r} and p_{g} are:

$$\mathrm{EMD}(p_{\mathrm{r}}, p_{\mathrm{g}}) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y) \sim \gamma} \|x - y\|$$

- try find a transport schedule $\gamma(x, y)$: to "move" amount of earth from one place $x \sim p_{\mathrm{g}}(x)$ (generated) distributed from over the domain of $y \sim p_{\mathrm{r}}(y)$ (real) or vice versa
- needs to ensure marginal distributions are still there:
- joint density acts the amount of *normalized* earth movement between individual factory and port.

$$\sum_{x} \gamma(x, y) = p_{\mathrm{r}}(y) \qquad \sum_{y} \gamma(x, y) = p_{\mathrm{g}}(x)$$

- this is our new **critic**

- **GAN** and **W-GAN**:
  1. GAN:
  $$\text{Discriminator: } \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \Big[\log D_{\theta_d}(\mathbf{x}_i) + \log\big(1 - D_{\theta_d}(G_{\theta_g}(\mathbf{z}_i))\big) \Big]$$
  $$\text{Generator: } \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\big(D_{\theta_d}(G_{\theta_g}(\mathbf{z}_i))\big)$$
  2. if we can change GAN into W-GAN:
  $$\text{find a critic: } \gamma^* = \inf_{\gamma \in \Pi} \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim \gamma, \mathbf{x} \sim p_g, \mathbf{y} \sim p_r} \|\mathbf{x} - \mathbf{y}\|$$
  $$\text{Generator: } \nabla_{\theta} \frac{1}{m} \sum_{i=1}^{m} \log\big(D_{\gamma^*}(G_{\theta}(\mathbf{z}_i))\big)$$
  that is all we need to do. However, it is impractical to compute:
  $$\gamma^* = \inf_{\gamma \in \Pi} \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim \gamma, \mathbf{x} \sim p_g, \mathbf{y} \sim p_r} \|\mathbf{x} - \mathbf{y}\|$$
  we need a lot of tricks!

Primal (constraint) function for EMD

$$
\underbrace{\left(\begin{array}{ccc|ccc|c|ccc|c}
\color{red}{1} & \color{red}{1} & \color{red}{\cdots} & \color{red}{0} & \color{red}{0} & \color{red}{\cdots} & \color{red}{\cdots} & \color{red}{0} & \color{red}{0} & \color{red}{\cdots} & \color{red}{\cdots} \\
0 & 0 & \cdots & 1 & 1 & \cdots & \cdots & 0 & 0 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\
0 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 1 & 1 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\
\hline
\color{blue}{1} & \color{blue}{0} & \color{blue}{\cdots} & \color{blue}{1} & \color{blue}{0} & \color{blue}{\cdots} & \color{blue}{\cdots} & \color{blue}{1} & \color{blue}{0} & \color{blue}{\cdots} & \color{blue}{\cdots} \\
0 & 1 & \cdots & 0 & 1 & \cdots & \cdots & 0 & 1 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\
0 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots
\end{array}\right)}_{\mathbf{A} \quad \text{size } n^2}
\underbrace{\left(\begin{array}{c}
\gamma(\mathbf{x}=1,\mathbf{y}=1) \\
\gamma(\mathbf{x}=1,\mathbf{y}=2) \\
\vdots \\
\hline
\gamma(\mathbf{x}=2,\mathbf{y}=1) \\
\gamma(\mathbf{x}=2,\mathbf{y}=2) \\
\vdots \\
\hline
\vdots \\
\hline
\gamma(\mathbf{x}=n,\mathbf{y}=1) \\
\gamma(\mathbf{x}=n,\mathbf{y}=2) \\
\vdots \\
\hline
\vdots
\end{array}\right)}_{\mathbf{\Gamma} \quad \text{size } n^2}
=
\underbrace{\left(\begin{array}{c}
\color{red}{p_g(1)} \\
p_g(2) \\
\vdots \\
p_g(n) \\
\vdots \\
\hline
\color{blue}{p_r(1)} \\
p_r(2) \\
\vdots \\
p_r(n) \\
\vdots
\end{array}\right)}_{\mathbf{b} \quad \text{size } 2n}
$$

- look at the RED line:

$$\sum_{\mathbf{y}} \gamma(\mathbf{x}=1,\mathbf{y}) = \color{red}{p_g(\mathbf{x}=1)}$$

- look at the BLUE line:

$$\sum_{\mathbf{x}} \gamma(\mathbf{x},\mathbf{y}=1) = \color{blue}{p_r(\mathbf{y}=1)}$$

W-GAN Linear Programming Primal and Dual form

- $\boldsymbol{\Gamma} \equiv \gamma(x, y)$ acts like a vectorized joint distribution, each element ≥ 0
- $\boldsymbol{C} \equiv \text{vec}(\mathbf{D}(x, y))$ acts like a vectorized cost
- $\mathbf{b} = \begin{bmatrix} p_r(y) \\ p_g(x) \end{bmatrix}$

primal form :

$$\min \left(z = \boldsymbol{C}^\top \boldsymbol{\Gamma} \right)$$
$$\text{s.t. } \mathbf{A}\boldsymbol{\Gamma} = \mathbf{b}$$
$$\text{and } \boldsymbol{\Gamma} \geq \mathbf{0}$$

dual form :

$$\max \left(\tilde{\mathbf{z}} = \mathbf{b}^\top \mathrm{F} \right)$$
$$\text{s.t. } \mathbf{A}^\top \mathrm{F} \leq \boldsymbol{C}$$

F is variable in dual function

Question why dual in linear programming is in such form?

Primal to Dual for Linear Programming (1)

- from `http://www.onmyphd.com/?p=duality.theory`
- let $\mathbf{x} \equiv \mathbf{\Gamma}$, and $\mathrm{F} = \mu$:

$$\min_{\mathbf{x}} \left[\boldsymbol{C}^\top \mathbf{x} \mid \underbrace{\mathbf{A}\mathbf{x} = \mathbf{b}}_{\mathbf{h}(\mathbf{x})}, \mathbf{x} \geq 0 \right]$$

$$\begin{aligned}
\mathcal{L}(\mathbf{x}, \mathrm{F}, \lambda) &= f(\mathbf{x}) + \mathrm{F}^\top \mathbf{h}(\mathbf{x}) + \lambda^\top \mathrm{g}(\mathbf{x}) \leq f(\mathbf{x}), \qquad \forall \mathbf{x} \in \mathcal{X}, \lambda \geq \mathbf{0}, \mathrm{F} \\
q(\mathrm{F}, \lambda) &= \inf_{\mathbf{x} \geq \mathbf{0}} \left[\mathcal{L}(\mathbf{x}, \mathrm{F}, \lambda) \right] \\
&= \inf_{\mathbf{x} \geq 0} \left[\boldsymbol{C}^\top \mathbf{x} + \mathrm{F}^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) \right] \\
&= \inf_{\mathbf{x} \geq 0} \left[(\boldsymbol{C}^\top + \mathrm{F}^\top \mathbf{A}) \mathbf{x} - \mathrm{F}^\top \mathbf{b} \right]
\end{aligned}$$

- **task** only include (F, λ) space which **avoid** making $q(\mathrm{F}, \lambda) = -\infty$ (maximization) constrains should be put to avoid these regions.

$$(\boldsymbol{C}^\top + \mathrm{F}^\top \mathbf{A}) < 0 \implies \mathbf{x} \text{ can be made arbitrarily large to make } q(\mathrm{F}, \lambda) \to -\infty$$

$$\textbf{if } \boldsymbol{C}^\top + \mathrm{F}^\top \mathbf{A} \geq \mathbf{0} \implies \mathbf{x}^* = 0 \implies q(\mathrm{F}, \lambda) = -\mathrm{F}^\top \mathbf{b}$$

- which means:

$$\max_{\mathrm{F}} \left[-\mathrm{F}^\top \mathbf{b} \mid \boldsymbol{C}^\top + \mathrm{F}^\top \mathbf{A} \geq 0 \right]$$

or let $\mathrm{F}' = -\mathrm{F}$:

$$\max_{\mathrm{F}'} \left[\mathrm{F}'^\top \mathbf{b} \mid \boldsymbol{C}^\top \geq \mathrm{F}'^\top \mathbf{A} \right]$$

Primal to Dual for Linear Programming (2)

let $\mathbf{x} \equiv \mathbf{\Gamma}$:

$$\text{assume the condition } \mathrm{F}^\top \mathbf{A} \leq \boldsymbol{C}^\top \ \forall \ \mathrm{F} : \qquad \text{this version works backwards}$$

$$\mathrm{F}^\top \underline{\mathbf{Ax}^*} \leq \boldsymbol{C}^\top \mathbf{x}^* \ \forall \ \mathrm{F} \ \text{ since } \mathbf{x}^* \geq 0\text{, after multiplication, no change sign}$$

$$\implies \mathrm{F}^\top \underbrace{\mathbf{b}} \leq \boldsymbol{C}^\top \mathbf{x}^* \ \forall \mathrm{F} \ \text{ assume } \mathbf{Ax}^* = \mathbf{b}$$

$$= \min_{\mathbf{x}} \left[\boldsymbol{C}^\top \mathbf{x} \,\middle|\, \mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq 0 \right]$$

$$\implies \underbrace{\max_{\mathrm{F}} [\mathrm{F}^\top \mathbf{b}]}_{\mathrm{F}^*} \leq \min_{\mathbf{x}} \left[\boldsymbol{C}^\top \mathbf{x} \,\middle|\, \mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq 0 \right]$$

$$\implies \max_{\mathrm{F}} \left[\mathrm{F}^\top \mathbf{b} | \mathrm{F}^\top \mathbf{A} \leq \boldsymbol{C}^\top \ \forall \ \mathrm{F} \right] \leq \min_{\mathbf{x}} \left[\boldsymbol{C}^\top \mathbf{x} \,\middle|\, \mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq 0 \right] \qquad \text{write the condition in}$$

Lagrangian Duality and KKT condition

Please refer to my notes on **Lagrangian Duality**
https://github.com/roboticcam/machine-learning-notes/blob/master/files/dual.pdf

Strong duality this time!

- we have proved that:

$$\max_{F} \left[F^\top \mathbf{b} | F^\top \mathbf{A} \leq \boldsymbol{C}^\top \ \forall F \right] \leq \min_{\mathbf{x}} \left[\boldsymbol{C}^\top \mathbf{x} \,\middle|\, \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq 0 \right]$$

- but we are greedy, we want to prove in w-GAN setting, it has strong duality:

$$\max_{F} \left[F^\top \mathbf{b} \,\middle|\, F^\top \mathbf{A} \leq \boldsymbol{C}^\top \ \forall F \right] = \min_{\mathbf{x}} \left[\boldsymbol{C}^\top \mathbf{x} \,\middle|\, \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq 0 \right]$$

- we can use **Farkas Lemma** to prove this

Farkas Lemma Proof Sketch

prove $\quad \max_{\mathsf{F}} \left[\mathsf{F}^\top \mathbf{b} \mid \mathsf{F}^\top \mathbf{A} \leq \boldsymbol{C}^\top \ \forall \, \mathsf{F} \right] \underbrace{=} \min_{\mathbf{x}} \left[\boldsymbol{C}^\top \mathbf{x} \,\middle|\, \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq 0 \right]$

where $z^* = \min_{\mathbf{x}} \left[\boldsymbol{C}^\top \mathbf{x} \,\middle|\, \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq 0 \right]$ is min in primal

1. extend cleverly everything by a single dimension 1:

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ -\boldsymbol{C}^\top \end{bmatrix}, \quad \hat{\mathbf{b}}_\epsilon = \begin{bmatrix} \mathbf{b} \\ -z^* + \epsilon \end{bmatrix}, \quad \hat{\mathsf{F}} = \begin{bmatrix} \mathsf{F} \\ \alpha \end{bmatrix} \text{ where } \epsilon, \alpha \in \mathbb{R}$$

2. when $\epsilon > 0$: after proved $\alpha > 0$ 2.1 using Farkas Lemma, we then prove:

$$\tilde{z} = \max_{\mathsf{F}} \left[\mathbf{b}^\top \mathsf{F} \,\middle|\, \mathbf{A}^\top \mathsf{F} \leq \boldsymbol{C} \right] > z^* - \epsilon \quad \text{(using Farkas Lemma again!)} \qquad 2.2$$

3. then it is obvious $\tilde{z} \in \Big((z^* - \epsilon), z^* \Big)$
   making ϵ infinitely small, we get

$$\tilde{z} = z^*$$

Convex and Conic combination

- matrix $\mathbf{A} \in \mathbb{R}^{d \times n} \triangleq (\mathbf{a}_1, \mathbf{a}_2, \dots \mathbf{a}_n)$
- **def** Convex combination:

$$C = \{\mathbf{a} \big| \mathbf{a} = \alpha_1 \mathbf{a}_1 + \ldots + \alpha_k \mathbf{a}_k, \alpha_1 + \ldots + \alpha_k = 1, \alpha_i \geq 0\}$$

  for example $\mathbf{A} \in \mathbb{R}^{2 \times 3}$, then it looks like a painted triangle
- **def** Conic combination is:

$$C = \{\mathbf{a} \big| \mathbf{a} = \alpha_1 \mathbf{a}_1 + \ldots + \alpha_k \mathbf{a}_k, \alpha_i \geq 0\}$$

  for example $\mathbf{A} \in \mathbb{R}^{2 \times 3}$, it looks painted cone from the origin

Farkas Lemma

- **Farkas Lemma** say, for a vector $\mathbf{b}$, there are exactly two **mutually exclusive** possibilities:
  1. $\mathbf{b}$ inside the cone:
  $$\exists\ \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \geq 0 \text{ (in every dimension)} \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}$$
  2. $\mathbf{b}$ outside the cone:
  $$\begin{aligned} \nexists\ \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \geq 0 \text{ (in every dimension)} \quad & \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b} \\ \forall\ \mathbf{x} \geq 0, \text{ (in every dimension)} \quad & \text{s.t. } \mathbf{A}\mathbf{x} \neq \mathbf{b} \end{aligned}$$

  these are not the most useful definitions, we use instead:
  $$\exists\ \mathrm{F} \in \mathbb{R}^m, \text{s.t. } \mathbf{A}^\top \mathrm{F} \leq 0 \text{ and } \mathbf{b}^\top \mathrm{F} > 0$$
  note that $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{x} \in \mathbb{R}^n$, they are not the same dimension

think about the geometry

$$\exists\, \mathrm{F} \in \mathbb{R}^m, \text{s.t. } \mathbf{A}^\top \mathrm{F} \leq 0 \text{ and } \mathbf{b}^\top \mathrm{F} > 0$$

where $\mathrm{F} \in \mathbb{R}^m$, and $\mathbf{x} \in \mathbb{R}^n$
the geometry can be thought as:

- $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ forms a cone, each $\mathbf{x}_i$ to be either an internal or external wall.
- F is the outer door, swing about the origin, that is more than $\frac{\pi}{2}$ away from each and every wall ($\mathbf{A}$), as $\mathbf{A}^\top \mathrm{F} \leq 0$
- $\mathbf{b}$ is an inner door, swing about the origin that is less than $\frac{\pi}{2}$ from outer door (F), as $\mathbf{b}^\top \mathrm{F} \geq 0$
- can made much clearer by include h (orthogonal to b):
  there is always a F and together with its orthogonal pair h to contain $\mathbf{b}$

- here comes a tricky bit: extend $\mathbf{a}_i$ by one dimension, i.e., $m \to m+1$, so the rest variables $(\mathbf{A}, \mathbf{b}, \mathrm{F})$ has an additional dimension:

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ -\boldsymbol{C}^\top \end{bmatrix}, \quad \hat{\mathbf{b}}_\epsilon = \begin{bmatrix} \mathbf{b} \\ -z^* + \epsilon \end{bmatrix} \quad \hat{\mathrm{F}} = \begin{bmatrix} \mathrm{F} \\ \alpha \end{bmatrix} \text{ where } \epsilon, \alpha \in \mathbb{R}$$

  note that **x** does **not** extend, so it can be applied in both systems
- also note that:

$$\hat{\mathbf{b}}_0 = \begin{bmatrix} \mathbf{b} \\ -z^* + 0 \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ -z^* \end{bmatrix}$$

2.1 Proving $\alpha > 0$ using Farkas Lemma (1)

- for $\epsilon = 0$, can prove $\exists\ \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \geq 0$ s.t. $\hat{\mathbf{A}}\mathbf{x} = \mathbf{b}_0 \implies \mathbf{b}_0$ inside cone, i.e,, Farkas case (1): obviously $\mathbf{x} = \mathbf{x}^*$ works!

$$\hat{\mathbf{A}}\mathbf{x}^* = \begin{bmatrix} \boldsymbol{A} \\ -\boldsymbol{c}^\top \end{bmatrix} \mathbf{x}^* = \begin{bmatrix} \boldsymbol{A}\mathbf{x}^* \\ -\boldsymbol{c}^\top \mathbf{x}^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ -z^* + 0 \end{bmatrix} = \hat{\mathbf{b}}_0$$

1. $\mathbf{b}$ inside the cone: $\exists\ \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \geq 0$ (in every dimension) s.t. $\mathbf{A}\mathbf{x} = \mathbf{b}$
2. $\mathbf{b}$ outside the cone: $\exists\ \mathrm{F} \in \mathbb{R}^m$, s.t. $\mathbf{A}^\top \mathrm{F} \leq 0$ and $\mathbf{b}^\top \mathrm{F} > 0$

since it's Farkas (1), then Farkas (2) can not exist, i.e.,:

$$\forall\ \hat{\mathbf{A}}^\top \hat{\mathrm{F}} \leq 0 \implies \underline{\hat{\mathbf{b}}_0^\top \hat{\mathrm{F}} \leq 0}$$

- **α-condition 1:** $\epsilon = 0 : \forall\ \hat{\mathbf{A}}^\top \hat{\mathrm{F}} \leq 0 \implies \hat{\mathbf{b}}_0^\top \hat{\mathrm{F}} \leq 0$

2.1 Proving $\alpha > 0$ using Farkas Lemma (2)

- for $\epsilon > 0$, there exists **no** nonnegative solution, meaning $\forall \mathbf{x}\ \hat{\mathbf{A}}\mathbf{x} \neq \hat{\mathbf{b}}_\epsilon$
- we look at:

$$\hat{\mathbf{A}}\mathbf{x} = \begin{bmatrix} \mathbf{A} \\ -\boldsymbol{C}^\top \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{A}\mathbf{x} \\ -\boldsymbol{C}^\top \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ -z^* + \epsilon \end{bmatrix} \text{ we want it to } = \hat{\mathbf{b}}_\epsilon$$

the blue part is feasible
the red part $-\boldsymbol{C}^\top \mathbf{x} = -z^* + \epsilon$ cannot be feasible, because:

$$z^* = \min_z \left(z \triangleq \boldsymbol{C}^\top \mathbf{x} \right)$$

$$\implies -z^* = \max_z \left(-\boldsymbol{C}^\top \mathbf{x} \right) = -\boldsymbol{C}^\top \mathbf{x}^* \underbrace{<}_{\text{no equal sign}} -z^* + \underbrace{\epsilon}_{>0}$$

even $\mathbf{x}^*$ can't be feasible, let alone any other $\mathbf{x}$!

- if Farkas(1) does not exist, then Farkas (2) must exist, i.e.:

$\exists\ \hat{\mathrm{F}} : \hat{\mathbf{A}}^\top \hat{\mathrm{F}} \leq 0$ and $\mathbf{b}_\epsilon^\top \hat{\mathrm{F}} > 0$ in another word, $\forall \hat{\mathrm{F}} : \hat{\mathbf{A}}^\top \hat{\mathrm{F}} \leq 0 \quad \exists\ \mathbf{b}_\epsilon^\top \hat{\mathrm{F}} > 0$

$$0 < \hat{\mathbf{b}}_\epsilon^\top \hat{\mathrm{F}} = \mathbf{b}^\top \mathrm{F} + \alpha(-z^* + \epsilon) = \underbrace{\mathbf{b}^\top \mathrm{F} + \alpha(-z^*)}_{\hat{\mathbf{b}}_0^\top \hat{\mathrm{F}}} + \alpha\epsilon = \hat{\mathbf{b}}_0^\top \hat{\mathrm{F}} + \alpha\epsilon$$

- **α-condition 2:** $\epsilon > 0 : \forall\ \hat{\mathbf{A}}^\top \hat{\mathrm{F}} \leq 0, \quad \exists\ \hat{\mathbf{b}}_0^\top \hat{\mathrm{F}} + \alpha\epsilon > 0$

2.1 Proving $\alpha > 0$ using Farkas Lemma (3)

- α**-condition 1:** $\epsilon = 0 : \forall\ \hat{\mathbf{A}}^\top \hat{\mathsf{F}} \leq 0 \implies \hat{\mathbf{b}}_0^\top \hat{\mathsf{F}} \leq 0$
- α**-condition 2:** $\epsilon > 0 : \forall\ \hat{\mathbf{A}}^\top \hat{\mathsf{F}} \leq 0 \quad \exists\ \hat{\mathbf{b}}_0^\top \hat{\mathsf{F}} + \alpha\epsilon > 0$
- since $\exists \hat{\mathsf{F}}$ satisfy both α-**conclusions**, it only works when $\alpha > 0$
- note that not every $\alpha > 0$ works, but it's a necessary conditions!

2.2 Prove $\tilde{z} > z^* - \epsilon$ using Farkas Lemma (1)

- we just proved that $\alpha > 0$, which implies by it won't change sign
- we saw when $\epsilon > 0$, there exists no non-negative solution, the **implication** is Farkas case (2): meaning when $\epsilon > 0$, there exist $\hat{\text{F}} \equiv \begin{bmatrix} \mathbf{F} \\ \alpha \end{bmatrix}$ solution such that:

$$\underbrace{\begin{bmatrix} \mathbf{A} \\ -\boldsymbol{C}^\top \end{bmatrix}^\top \begin{bmatrix} \text{F} \\ \alpha \end{bmatrix} \leq \mathbf{0}}_{\implies \mathbf{A}^\top \text{F} \leq \alpha \boldsymbol{C}} \quad \underbrace{\begin{bmatrix} \mathbf{b} \\ -z^* + \epsilon \end{bmatrix} \begin{bmatrix} \text{F} \\ \alpha \end{bmatrix} > 0}_{\implies \mathbf{b}^\top \text{F} > \alpha(z^* - \epsilon)}$$

$$\mathbf{A}^\top \text{F} \leq \alpha \boldsymbol{C} \implies \mathbf{A}^\top \frac{\text{F}}{\alpha} \leq \boldsymbol{C}$$

$$\mathbf{b}^\top \text{F} > \alpha(z^* - \epsilon) \implies \mathbf{b}^\top \frac{\text{F}}{\alpha} > (z^* - \epsilon)$$

2.2 Prove $\tilde{z} > z^* - \epsilon$ using Farkas Lemma (2)

- now we have: $\mathbf{A}^\top \frac{\mathsf{F}}{\alpha} \leq \boldsymbol{C}$ and $\mathbf{b}^\top \frac{\mathsf{F}}{\alpha} > (z^* - \epsilon)$

$$\underbrace{\mathbf{A}^\top \mathsf{F} \leq \boldsymbol{C}}_{\text{constraint}} \quad \text{and} \quad \underbrace{\mathbf{b}^\top \mathsf{F} > (z^* - \epsilon)}_{\text{obj}}$$

- combine the two above, we have:

$$\tilde{z} = \max_{\mathsf{F}} \left[\mathbf{b}^\top \mathsf{F} \middle| \mathbf{A}^\top \mathsf{F} \leq \boldsymbol{C}\right] > z^* - \epsilon$$

- we can make ϵ arbitrarily small, to make $\tilde{z} = z^*$, so we have **strong** duality!

- can be proved that if $\mathbf{Ax} \geq \mathbf{b}$ instead of $\mathbf{Ax} = \mathbf{b}$:

primal form :

$$
\begin{aligned}
&\min\left(z = \boldsymbol{C}^\top \mathbf{x}\right) \\
\text{s.t. } &\mathbf{Ax} \geq \mathbf{b} \\
\text{and } &\mathbf{x} \geq \mathbf{0}
\end{aligned}
$$

dual form :

$$
\begin{aligned}
&\max\left(\tilde{\mathbf{z}} = \mathbf{b}^\top \mathrm{F}\right) \\
\text{s.t. } &\mathbf{A}^\top \mathrm{F} \leq \boldsymbol{C} \\
&\mathrm{F} \geq \mathbf{0} \quad \text{this is added}
\end{aligned}
$$

Objective function $\mathbf{b}^\top \mathrm{F}$

Put back into Wasserstein Distance problem:

- switching generic symbols back: $\mathbf{\Gamma} \equiv \mathbf{x}$
- we know primal and dual are equal then:

$$\min_{\mathbf{\Gamma}} \left[\mathbf{\Gamma}^\top \boldsymbol{C} \;\middle|\; \mathbf{A}\mathbf{\Gamma} = \mathbf{b},\; \mathbf{\Gamma} \geq 0\right\} = \max_{\mathrm{F}} \left[\mathbf{b}^\top \mathrm{F} \middle| \; \mathbf{A}^\top \mathrm{F} \leq \boldsymbol{C}\right]$$

- by breaking up F into $\begin{bmatrix} f_r^w \\ f_g^w \end{bmatrix}$ to match with $\mathbf{b}$:

$$\mathrm{F} = \begin{bmatrix} f_g^w(\mathbf{x}=1) \\ f_g^w(\mathbf{x}=2) \\ \vdots \\ f_g^w(\mathbf{x}=n) \\ \vdots \\ \hline f_r^w(\mathbf{y}=1) \\ f_r^w(\mathbf{y}=2) \\ \vdots \\ f_r^w(\mathbf{y}=n) \\ \vdots \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} p_g(\mathbf{x}=1) \\ p_g(\mathbf{x}=2) \\ \vdots \\ p_g(\mathbf{x}=n) \\ \vdots \\ \hline p_r(\mathbf{y}=1) \\ p_r(\mathbf{y}=2) \\ \vdots \\ p_r(\mathbf{y}=n) \\ \vdots \end{bmatrix}$$

Objective function $\mathbf{b}^\top \mathrm{F}$ (2)

- from previous slide, we have:

$$
\begin{aligned}
\mathbf{b}^\top \mathrm{F} &= \sum_n p_{\mathrm{g}}(\mathbf{x} = n) f_{\mathrm{g}}^w(\mathbf{x} = n) + \sum_n p_{\mathrm{r}}(\mathbf{y} = n) f_{\mathrm{r}}^w(\mathbf{y} = n) \\
&= \sum_n p_{\mathrm{g}}(n) f_{\mathrm{g}}^w(n) + \sum_n p_{\mathrm{r}}(n) f_{\mathrm{r}}^w(n)
\end{aligned}
$$

- however, we change the variable from $n \rightarrow \mathbf{x}$:

$$
\begin{aligned}
\mathbf{b}^\top \mathrm{F} &= \sum_{\mathbf{x}} p_{\mathrm{g}}(\mathbf{x}) f_{\mathrm{g}}^w(\mathbf{x}) + \sum_{\mathbf{x}} p_{\mathrm{r}}(\mathbf{x}) f_{\mathrm{r}}^w(\mathbf{x}) \\
&= \sum_{\mathbf{x}} \left[p_{\mathrm{g}}(\mathbf{x}) f_{\mathrm{g}}^w(\mathbf{x}) + p_{\mathrm{r}}(\mathbf{x}) f_{\mathrm{r}}^w(\mathbf{x}) \right]
\end{aligned}
$$

Constraint $\mathbf{A}^\top \mathrm{F} \leq \boldsymbol{C}$

$$\min_{\boldsymbol{\Gamma}} \left[\boldsymbol{\Gamma}^\top \boldsymbol{C} \mid \mathbf{A}\boldsymbol{\Gamma} = \mathbf{b},\ \boldsymbol{\Gamma} \geq 0\right\} = \max_{\mathrm{F}} \left[\mathbf{b}^\top \mathrm{F} \mid \mathbf{A}^\top \mathrm{F} \leq \boldsymbol{C}\right]$$

$$\underbrace{\left[\begin{array}{ccccc|ccccc}
1 & 0 & \cdots & 0 & \cdots & 1 & 0 & \cdots & 0 & \cdots \\
1 & 0 & \cdots & 0 & \cdots & 0 & 1 & \cdots & 0 & \cdots \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots \\
\hline
0 & 1 & \cdots & 0 & \cdots & 1 & 0 & \cdots & 0 & \cdots \\
0 & 1 & \cdots & 0 & \cdots & 0 & 1 & \cdots & 0 & \cdots \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots \\
\hline
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots \\
\hline
0 & 0 & \cdots & 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots \\
0 & 0 & \ddots & 1 & \ddots & 0 & 1 & \ddots & 0 & \ddots \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots \\
\hline
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots
\end{array}\right]}_{\mathbf{A}^\top}
\underbrace{\left[\begin{array}{c}
f_g^W(\mathbf{x}=1) \\
f_g^W(\mathbf{x}=2) \\
\vdots \\
f_g^W(\mathbf{x}=n) \\
\vdots \\
\hline
f_r^W(\mathbf{y}=1) \\
f_r^W(\mathbf{y}=2) \\
\vdots \\
f_r^W(\mathbf{y}=n) \\
\vdots
\end{array}\right]}_{\mathrm{F}}
\leq
\underbrace{\left[\begin{array}{c}
d(1,1) \\
d(1,2) \\
\vdots \\
\hline
d(2,1) \\
d(2,2) \\
\vdots \\
\hline
\vdots \\
\hline
d(n,1) \\
d(n,2) \\
\vdots \\
\hline
\vdots
\end{array}\right]}_{\boldsymbol{C}}$$

pick any row of $\mathbf{A}^\top$, gives you:

$$f_g^W(\mathbf{x} = i) + f_r^W(\mathbf{y} = j) \leq d(i, j)$$

$$i \to \mathbf{x} \text{ and } j \to \mathbf{y}: \qquad f_g^W(\mathbf{x}) + f_r^W(\mathbf{y}) \leq d(\mathbf{x}, \mathbf{y}) \quad \forall \mathbf{x}, \mathbf{y}$$

Put Objective and Constraint together:

- dual function:

$$\mathcal{W}(p_g, p_r) = \max_{F} \left[\mathbf{b}^\top F \middle| \, \mathbf{A}^\top F \leq \boldsymbol{C} \right]$$

$$= \max_{f_r^w, f_g^w} \left\{ \underbrace{\sum_{\mathbf{x}} \left[p_g(\mathbf{x}) f_g^w(\mathbf{x}) + p_r(\mathbf{x}) f_r^w(\mathbf{x}) \right]}_{\mathbf{b}^\top F} \;\middle|\; \underbrace{f_g^w(\mathbf{x}) + f_r^w(\mathbf{y}) \leq d(\mathbf{x}, \mathbf{y}) \quad \forall \mathbf{x}, \mathbf{y}}_{\mathbf{A}^\top F \leq \boldsymbol{C}} \right\}$$

reduce argument to only f instead of f_g^w and f_r^w

- for $\mathbf{x} = \mathbf{y}$, each $\mathbf{x}$ can be constrained interdependently:

$$\max_{f_r^w, f_g^w} \left[p_g(\mathbf{x}) f_g^w(\mathbf{x}) + p_r(\mathbf{x}) f_r^w(\mathbf{x}) \,\middle|\, f_g^w(\mathbf{x}) + f_r^w(\mathbf{x}) \leq 0 \quad \forall \mathbf{x} \right]$$
$$= \max_{t_1, t_2} \left[p_1 t_1 + p_2 t_2 \,|\, t_1 + t_2 \leq 0, \quad p_1, p_2 \geq 0 \right]$$

- say we have fixed $\max(|t_1|, |t_2|)$, e.g., $= 5$
  wlog: $t_1 \leq 0, t_2 \geq 0$ suppose $|x_1| \geq |x_2|$, e.g., $t_1 = -5$, $t_2 = 3$:

$$\begin{aligned} \max(p_1, p_2) t_1 + \min(p_1, p_2) t_2 &\leq \max(p_1, p_2) t_2 + \min(p_1, p_2) t_1 \\ &\leq \max(p_1, p_2) t_2 + \min(p_1, p_2)(-t_2) \end{aligned}$$

- for $\mathbf{x} \neq \mathbf{y}$, constraint $d(\mathbf{x}, \mathbf{y})$ does not impact the objective function, but give constraints to $|t_1|$
- therefore:

$$\begin{aligned} \max_{f_r^w, f_g^w} [\mathbf{b}^\top \mathrm{F}] &= \max_{f_r^w} \int_{\mathbf{x}} \left[p_r(\mathbf{x}) f_r^w(\mathbf{x}) + p_g(\mathbf{x}) \left(-f_r^w(\mathbf{x}) \right) \right] \\ &= \max_{f} \int_{\mathbf{x}} \left[p_r(\mathbf{x}) f(\mathbf{x}) - p_g(\mathbf{x}) f(\mathbf{x}) \right] \quad \forall f(\mathbf{x}) \qquad \text{substitute } f \equiv f_r^w(x) = -f_g^w(\mathbf{x}) \end{aligned}$$

WGAN algorithm in detail

$$\Longrightarrow \mathcal{W}(p_g, p_r) = \max_{f} \left\{ \int \left[p_r(\mathbf{x}) f(\mathbf{x}) - p_g(\mathbf{x}) f(\mathbf{x}) \right] \mathrm{d}\mathbf{x} \;\middle|\; f(\mathbf{x}) - f(\mathbf{y}) \leq d(\mathbf{x}, \mathbf{y}) \right\}$$

$$= \max_{f,\ \|f\|_L \leq 1} \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_g(\mathbf{x})}[f(\mathbf{x})]$$

- put all together:

$$\mathcal{W}(p_g, p_r) = \max_{f,\ \|f\|_L \leq 1} \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_g(\mathbf{x})}[f(\mathbf{x})]$$

$$\Longrightarrow \mathcal{L}(G, f) = \min_{G} \left[\max_{f,\ \|f\|_L \leq 1} \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})}[f(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})}[f(G_\theta(\boldsymbol{z}))] \right]$$

- in words, the discriminator/critic is try to find a 1-Lipschitz function f that best aligns with real data from p_r and aligns poorly with generated data p_g

What is Mini-Max?

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\Pr(i,j)a_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n}\Pr(x=i)\Pr(y=j)a_{ij} = \mathbf{x}^\top\mathbf{A}\mathbf{y}$$

$$\begin{aligned}
\max_{\mathbf{x}}\left(\min_{\mathbf{y}}\mathbf{x}^\top\mathbf{A}\,\mathbf{y}\right) &= \max_{\mathbf{x}}\left(\min_{\mathbf{y}}\left[y_1\mathbf{x}^\top\mathbf{a}_1, \quad \ldots, \quad y_k\mathbf{x}^\top\mathbf{a}_k\right]\right)\\
&= \max_{\mathbf{x}}\left(\min\left\{y_1\mathbf{x}^\top\mathbf{a}_1, \ldots, y_k\mathbf{x}^\top\mathbf{a}_k\right\}\right)\\
&= \max_{\mathbf{x}}\left(\min_{j\in\{1,\ldots,k\}}\mathbf{x}^\top\mathbf{A}\,\mathbf{e}_j\right)
\end{aligned}$$

since nested max/min doesn't work, we have:

$$\begin{aligned}
&\max_{\mathbf{x}} v\\
\text{s.t: } v - \mathbf{a}_j^\top\mathbf{x} \leq 0 \quad \forall j \qquad &\implies \quad v \leq \mathbf{a}_j^\top\mathbf{x}\ \forall j\\
&\sum_{i=1}^{m} x_i = 1, \quad x_1, \ldots, x_m \geq 0
\end{aligned}$$

Apply Minimax theorem to WGAN formulation

$$
\begin{aligned}
W(p_r, p_\theta) &= \inf_{\gamma \in \pi} \mathbb{E}_{x,y\sim\gamma}[\|x-y\|] \\
&= \inf_{\gamma} \mathbb{E}_{x,y\sim\gamma}\Big[\|x-y\| + \underbrace{\sup_f \mathbb{E}_{s\sim p_r}[f(s)] - \mathbb{E}_{t\sim p_\theta}[f(t)] - (f(x)-f(y))}_{\begin{cases} 0, & \text{if } \gamma \in \pi \\ +\infty & \text{else} \end{cases}}\Big]
\end{aligned}
$$

as $x, y \not\sim \gamma \implies \mathbb{E}_{s\sim p_r}[f(s)] \neq f(x)$ and $\mathbb{E}_{t\sim p_\theta}[f(t)] \neq f(y)$, can apply some extreme f to make it ∞

$$
\begin{aligned}
&= \inf_{\gamma} \sup_f \mathbb{E}_{x,y\sim\gamma}[\|x-y\| + \mathbb{E}_{s\sim p_r}[f(s)] - \mathbb{E}_{t\sim p_\theta}[f(t)] - (f(x)-f(y))] \\
&= \sup_f \inf_{\gamma} \mathbb{E}_{x,y\sim\gamma}[\|x-y\| + \mathbb{E}_{s\sim p_r}[f(s)] - \mathbb{E}_{t\sim p_\theta}[f(t)] - (f(x)-f(y))]
\end{aligned}
$$

can swap inf, sup due to convex-concave

$$
= \sup_f \mathbb{E}_{s\sim p_r}[f(s)] - \mathbb{E}_{t\sim p_\theta}[f(t)] + \underbrace{\inf_{\gamma} \mathbb{E}_{x,y\sim\gamma}[\|x-y\| - (f(x)-f(y))]}_{\begin{cases} 0, & \text{if } \|f\|_L \leq 1 \\ -\infty & \text{else} \end{cases}}
$$

in the case of $\|f\|_L \leq 1$:

$$
\begin{aligned}
\|f(x_1) - f(x_2)\| &\leq \underbrace{K}_{=1} \|x_1 - x_2\| \\
\implies \|x_1 - x_2\| &\geq (f(x_1) - f(x_2)) \\
\implies \|x_1 - x_2\| - (f(x_1) - f(x_2)) &\geq 0
\end{aligned}
$$

think $4 - 3 > 0$ and $4 - (-3) > 0$

L-Lipschitz gradient

- remaining question is about L-Lipschitz function:

$$\max_{f,\ \|f\|_L \leq 1} \mathbb{E}_{\mathbf{x} \sim p_g(\mathbf{x})}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[f(\mathbf{x})]$$

- the key is to know why:

$$L = \max_x |f'(x)|$$

- i.e., a differentiable function f is L-Lipschitz if and only if it has gradients with norm at most L everywhere.
- we can then do both Gradient-Clipping and Gradient-Penalty!

why L-Lipschitz f has gradients with norm at most L everywhere

- for L-Lipschitz function in general, i.e., include non-convex f:
- Given $x < y$ in interval (a, b), (prove the case of $y < x$ is equally easy):

$$
\begin{aligned}
|f(x) - f(y)| &= \underbrace{\left| \int_x^y f'(t)\mathrm{d}t \right| \leq \int_x^y |f'(t)|\mathrm{d}t}_{|a+b| \leq |a|+|b|} \\
&\leq \max_{t \in [x,y]} |f'(t)| \int_x^y 1\mathrm{d}t = \underbrace{\max_{t \in [x,y]} |f'(t)|}_{L} |x - y|
\end{aligned}
$$

- we conclude that:

$$
|f(x) - f(y)| \leq L|x - y| \implies L = \max_{t \in [x,y]} |f'(t)|
$$

Ensure function f is 1-Lipschitz: Weight Clipping

- since the the weights w are written as $w^\top \mathbf{x}$ in neural network, derivative w.r.t input $\mathbf{x}$ $\frac{\partial \mathcal{W}}{\partial \mathbf{x}}$ will be in terms of w, so:
- need to limit all weights $w_i \in [-c, c]$

Ensure function f is 1-Lipschitz: Weight ClippingGradient Penalty

- since largest of gradient of a 1-Lipschitz function ∇,

$$\mathcal{W}_{\text{GP}} = \underbrace{\mathbb{E}_{\tilde{x} \sim p_{\text{g}}}[f(\tilde{x})] - \mathbb{E}_{x \sim p_{\text{r}}}[f(x)]}_{\text{critic loss}} + \underbrace{\lambda \mathbb{E}_{\hat{x} \sim P_{\hat{x}}} \left[(\|\nabla_{\hat{x}} f(\hat{x})\|_2 - 1)^2 \right]}_{\text{Gradient Penalty}}$$

- the above **critic loss** is a minimization instead of maximization, so we switched the term around, i.e., instead of:

$$\mathbb{E}_{x \sim p_{\text{r}}}[f(x)] - \mathbb{E}_{\tilde{x} \sim p_{\text{g}}}[f(\tilde{x})]$$

where

$$\hat{\mathbf{x}} = t\tilde{\mathbf{x}} + (1 - t)\mathbf{x} \qquad 0 \leq t \leq 1$$

Lipschiz property and norms of matrix parameters neural networks

- what if we add some norm based regularizer to the matrix parameter $||W||$?
- when kind of L-Lipschiz does it correspond to?

Lipschiz property for Neural Networks

- given $f = \sigma(W^\top \mathbf{x} + b)$, we may want to have a look at what L-Lipschiz is this?

$$
\begin{aligned}
\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\| &\leq L\|(\mathbf{x}_1 - \mathbf{x}_2)\| \\
\implies \|\sigma(W^\top \mathbf{x}_1 + b) - \sigma(W^\top \mathbf{x}_2 + b)\| &\leq L\|(\mathbf{x}_1 - \mathbf{x}_2)\|
\end{aligned}
$$

Let

$$
\begin{aligned}
f(\mathbf{x}_1) - f(\mathbf{x}_2) &\approx (\nabla_{\mathbf{x}} f(\mathbf{x}))(\mathbf{x}_1 - \mathbf{x}_2) \qquad \text{where } \mathbf{x}_1 \leq \mathbf{x} \leq \mathbf{x}_2 \\
&= \nabla_{\mathbf{x}} \sigma(W^\top \mathbf{x} + b)(\mathbf{x}_1 - \mathbf{x}_2) \qquad \text{and } \nabla_{\mathbf{x}} \sigma(W^\top \mathbf{x} + b) = \sigma'(\underbrace{W^\top \mathbf{x} + b}_{z}) \times \underbrace{W}_{\frac{dz}{d\mathbf{x}}} \\
&= \sigma'(W^\top \mathbf{x} + b) W (\mathbf{x}_1 - \mathbf{x}_2)
\end{aligned}
$$

$\sigma'(W^\top \mathbf{x} + b)$ can be chosen to be bounded!

- so we need to look at:

$$
\begin{aligned}
\|W^\top(\mathbf{x}_1 - \mathbf{x}_2)\| &\leq L\|(\mathbf{x}_1 - \mathbf{x}_2)\| \\
\text{wlof : } \|W(\mathbf{x}_1 - \mathbf{x}_2)\| &\leq L\|(\mathbf{x}_1 - \mathbf{x}_2)\|
\end{aligned}
$$

Frobenius norm $\|\cdot\|_F$

- **definition**:

$$
\begin{aligned}
\|W\|_F &= \sqrt{\left(\sum_{i,j=1}^{n} |W_{ij}|^2\right)} \\
&= \sqrt{\mathrm{tr}(WW^\top)} = \sqrt{\mathrm{tr}(W^\top W)} \\
&= \text{is the L2 regularizer!}
\end{aligned}
$$

- it's a matrix norm, therefore:

$$
\|WB\|_F \leq \|W\|_F \|B\|_F
$$

- unitary invariant, for all unitary vector, U and V, where $U^\top = U^{-1}$

$$
\|W\|_F = \|UW\|_F = \|WV\|_F = \|UWV\|_F
$$

- can prove the following:

$$
\|W\|_2 = \sqrt{\sigma_{\max}(W^\top W)} \leq \|W\|_F = \sqrt{n}\sqrt{\sigma_{\max}(W^\top W)}
$$

- Frobenius norm is an upper-bound of spectral norm!

Frobenius Norm for L-Lipschiz

- using cauchy schwarz:

$$
\begin{aligned}
\|W\mathbf{x}\|^2 &= \sum_{i=1}^{m} \Big| \sum_{j=1}^{n} W_{ij} x_j \Big|^2 \leq \sum_{i=1}^{m} \left\{ \Big(\sum_{j=1}^{n} |W_{ij}|^2 \Big) \Big(\sum_{j=1}^{n} |x_j|^2 \Big) \right\} \\
&= \Big(\sum_{i=1}^{m} \sum_{j=1}^{n} |W_{ij}|^2 \Big) \|\mathbf{x}\|^2 \\
&= \|W\|_F^2 \|\mathbf{x}\|^2 \\
\implies \|W\mathbf{x}\| &\leq \|W\|_F \|\mathbf{x}\| \quad \forall \mathbf{x} \\
\implies \|W(\mathbf{x}_1 - \mathbf{x}_2)\| &\leq \underbrace{\|W\|_F}_{L} \|\mathbf{x}_1 - \mathbf{x}_2\|
\end{aligned}
$$

- adding $\mathbf{W}\|_F^2$, a.k.a, L2 regularizer helps with neural network with a $(L = \mathbf{W}\|_F)$-Lipschiz, but it may not be tight enough
- since $\|W\|_2 = \sqrt{\sigma_{\max}(W^\top W)} \leq \|W\|_F$, let's see if we can use $L = \|W\|_2$, aka, spectral norm

Dual Norm

- Given a linear function $f_z(\cdot)$, how "big" is its output, i.e., how big is the number $f_z(x) = z^\top x$ relative to the size (norm) of x? This is exactly the number:

$$\frac{z^T x}{\|x\|}$$

  we need to normalize by $\|x\|$ to remove the effects of input x
- We say that norm of z is the largest this quantity can possibly be:

$$\|z\|_* = \sup_{x \neq 0} \frac{z^T x}{\|x\|}$$

- or more generically:

$$\underbrace{\|z\|_*}_{\text{dual norm}} = \sup \left\{ x^\top z \mid \underbrace{\|x\|}_{\text{"ordinary" norm}} \leq 1 \right\}$$

- Dual norm of L_2 norm is the L_2 norm. Dual norm of L_1 norm is L_∞ norm

Dual Norm

- Dual norm of L_2 norm is the L_2 norm:

$$\sup\{z^\top x \mid \|x\|_{L_2} \leq 1\} = \|z\|_{L_2}$$

max occurs when x is a unit vector pointing in the same direction as z

- Dual norm of L_1 norm is L_∞ norm and vice versa:

$$\sup\{z^\top x \mid \underbrace{\|x_{L_\infty}\|}_{\max(|x_1|,\ldots,|x_n|)} \} \leq 1 = \|z\|_{L_2}$$

max occurs when x is in corner of a square where signs of each dimesion matches betwee z and x

for example, $z = (-5, 5)^\top \implies x = (-1, 1)$

Matrix norm: p-norm vector

1. in general:

$$\|A\|_p = \sup_{\|x\| \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$
$$= \sup\{\|Ax\|_p \mid \|x\|_p = 1\}$$

2. $p = 1$:

$$\|A\|_1 = \sup_{\|x\|_1 = 1} \|Ax\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$$

   the "chosen" x will be a one hot vector: like a column selector to find a column with max sum of absolute value
3. $p = \infty$:

$$\|A\|_\infty = \sup_{\|x\|_\infty = 1} \|Ax\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$$

   the "chosen" x will be a vector of $\{+1, -1\}$ to suit the row with max sum of absolute values
4. $p = 2$: **spectral norm**

$$\|A\|_2 = \sup_{\|x\|_2 = 1} \|Ax\|_2 = \sqrt{\lambda_{\max}(A^\top A)} = \sqrt{\lambda_{\max}(AA^\top)}$$

Spectral Norm

$$
\begin{aligned}
\|\mathbf{A}\|_2^2 &= \sup_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2^2 \\
& \sup_{\|\mathbf{x}\|_2=1} (\mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x}) \\
&= \max_{\|\mathbf{x}\|_2=1} \mathbf{x}^\top U \operatorname{diag}(\lambda_1, \dots, \lambda_n) U^\top \mathbf{x} \\
&= \max_{\|\mathbf{y}\|_2=1} \mathbf{y}^\top \operatorname{diag}(\lambda_1, \dots, \lambda_n) \mathbf{y} \qquad \text{since } U \text{ is orthogonal matrix } \|\mathbf{x}\|_2 = \| \underbrace{U\mathbf{x}}_{\mathbf{y}} \|_2 \\
&= \max_{\|\mathbf{y}\|_2=1} \lambda_1 y_1^2 + \dots + \lambda_n y_n^2 \\
&= \max\{\lambda_1, \dots, \lambda_n\} \text{ the chosen } \mathbf{y} \text{ is when } (y_1^2, \dots y_n^2) \text{ is a one hot corresponding to largest } \lambda \\
&= \lambda_{\max}(\mathbf{A}^\top \mathbf{A})
\end{aligned}
$$

Question: what is wrong with instead finding a vector $\begin{bmatrix} y_1^2 & \dots & y_n^2 \end{bmatrix}$ that is in the same direction as $[\lambda_1 \quad \dots \quad \lambda_n]$?

Answer: $\|\mathbf{y}\|_2 = 1 \implies [y_1 \quad \dots \quad y_n]$ is a unit vector and $\begin{bmatrix} y_1^2 & \dots & y_n^2 \end{bmatrix}$ is not!

$$
\|\mathbf{A}\|_2 = \sup_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}^\top \mathbf{A})} = \sqrt{\lambda_{\max}(\mathbf{A}\mathbf{A}^\top)}
$$

Compute Spectral Norm

- compute $\sigma_{\max}(A^\top A)$ is hard!
- however, we can approximate it by:

$$
\begin{aligned}
&\textbf{repeat}: \\
&\qquad u \leftarrow \frac{(\mathbf{A}^\top \mathbf{A})u}{\|(\mathbf{A}^\top \mathbf{A})u\|} \\
&\|\mathbf{A}\|_2^2 \approx u^\top \mathbf{A}^\top \mathbf{A} u
\end{aligned}
$$

- or

$$
\begin{aligned}
&\textbf{repeat}: \\
&\qquad v \leftarrow \frac{\mathbf{A}^\top u}{\|\mathbf{A}^\top u\|}, \; u \leftarrow \frac{\mathbf{A}v}{\|\mathbf{A}v\|} \\
&\|\mathbf{A}\|_2^2 \approx u^\top \mathbf{A}^\top \mathbf{A} v
\end{aligned}
$$

- why it works?

Why it works?

- this is very similar to Power Method: https://github.com/roboticcam/machine-learning-notes/blob/master/stochastic_matrices.pdf
- however, this time, $\lambda_{\max}(K \equiv \mathbf{A}^\top \mathbf{A}) \neq 1$!
- but the same can still apply:

$$
\begin{aligned}
u^{(0)} &= c_1 v_1 + \cdots + c_n v_n \\
\Longrightarrow \quad K^t u^{(0)} &= c_1 K^r v_1 + \cdots + c_n K^r v_n \\
&= c_1 \lambda_1^r v_1 + \cdots + c_n \lambda_n^r v_n \\
&\approx c_1 \lambda_1^r v_1
\end{aligned}
$$

  means $K^t u^{(0)}$ gives a good approximation to un-normalized v_1
- which we can see the first term dominates! However, it may grow significantly big! We therefore, need a normalization term:

$$
\tilde{v}_1 \leftarrow \frac{Ku}{\|Ku\|}
$$

- finally

$$
\begin{aligned}
A\tilde{v}_1 &= \lambda_1 \tilde{v}_1 \\
\Longrightarrow \quad \tilde{v}_1^\top A \tilde{v}_1 &= \lambda_1 \tilde{v}_1^\top \tilde{v}_1 = \lambda_1 = \|\mathbf{A}\|_2^2
\end{aligned}
$$

Spectral Norm for L-Lipschiz

$$
\begin{aligned}
\|W\|_2 &= \max_{x \neq 0} \frac{\|W\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \\
\implies \; \|W\mathbf{x}\|_2 &\leq \|W\|_2 \; \|\mathbf{x}\|_2 \\
\|W\mathbf{x}\| &\leq \|W\|_2 \; \|\mathbf{x}\| \;\; \forall \mathbf{x} \;\; \text{drop the L2 norm index for vector} \\
\implies \; \|W(\mathbf{x}_1 - \mathbf{x}_2)\| &\leq \underbrace{\|W\|_2}_{L} \; \|\mathbf{x}_1 - \mathbf{x}_2\|
\end{aligned}
$$

enforcing W to keep its norm value closer to $\|W\|_2$, makes the function more robust than Frobenius norm!

- Enough of W-GAN, talk something new!
- **Discriminator**

$$
\begin{aligned}
U &= \arg\min_U \mathbb{E}_{x\sim p(x)}[U(x)] - \mathbb{E}_{x\sim \hat{q}(x)}[u(X)] + \lambda \mathbb{E}_{x\sim p(x)}[\|\nabla_x U(x)\|^2] \\
&= \arg\min_U \mathbb{E}_{x\sim p(x)}[U(x)] - \mathbb{E}_{z\sim q(z)}[U(G(z))] + \lambda \mathbb{E}_{x\sim p(x)}[\|\nabla_x U(x)\|^2]
\end{aligned}
$$

- **Generator**

$$
G = \arg\min_G \mathbb{E}_{z\sim q(z)}[U(G(z))]
$$

InfoGAN

- Original GAN:

$$\min_G \max_D \left(L(D, G) \equiv \mathbb{E}_{x \sim p_r(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \right)$$
$$\implies L(D, G) = \underbrace{\mathbb{E}_{x \sim p_r(x)}[\log D(x)] + \mathbb{E}_{x \sim p_g(x)}[\log(1 - D(x)]}_{V(D,G)}$$

problem is that the z sampled is not controllable. We need to append it with a code c

- infoGAN:

$$\min_G \max_D L(D, G) = V(D, G) - \lambda \mathbf{I}(c; G(z, c))$$

- $\mathbf{I}(c, \mathbf{x})$ is mutural information, how much we know about c when we know $\mathbf{x}$ and vice versa
- if $\mathbf{x}$ and c are completely uncorrelated: $\implies \mathbf{I}(c, \mathbf{x})$ is low
- if $\mathbf{x}$ and c are correlated: $\implies \mathbf{I}(c, \mathbf{x})$ is high

Conditional Entropy

- Conditional entropy:

$$\mathrm{H}(Y|X) = -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)} = -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log p(y|x)$$

- note that conditional entropy $\mathrm{H}(Y|X)$ and cross entropy $H(P\|Q)$ are not the same thing!

Variational stuff

- $\mathbf{I}(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$: $\quad c \equiv Y$ and $G(z, c) \equiv X$

$$
\begin{aligned}
\mathbf{I}(c; G(z, c)) &= H(c) - H(c|G(z, c)) \\
&= \mathbb{E}_{x \sim G(z,c)} \Big[\underbrace{\mathbb{E}_{c' \sim p(c|x)} \big[\log P(c'|x) \big]}_{-H(P)} \Big] + H(c)
\end{aligned}
$$

- use variational stuff, and remember, from `https://github.com/roboticcam/machine-learning-notes/blob/master/regression.pdf`:

$$
H(P\|Q) = H(P) + \mathrm{KL}(P\|Q) \implies H(P\|Q) \geq H(P) \implies -H(P) \geq -H(P\|Q)
$$

$$
\implies \mathbf{I}(c; G(z, c)) \geq \mathbb{E}_{x \sim G(z,c)} \Big[\underbrace{\mathbb{E}_{c' \sim p(c|x)} \big[\log Q(c'|x) \big]}_{-H(P\|Q)} \Big] + H(c)
$$

$$
\mathcal{L}_I(G, Q) = \mathbb{E}_{c \sim P(c), x \sim G(z,c)} [\log Q(c|x)] + H(c)
$$

- from previous page:

$$\mathbf{I}(c; G(z,c)) \geq \mathbb{E}_{x\sim G(z,c)}\left[\mathbb{E}_{c'\sim \underbrace{p(c|x)}_{\text{too hard!}}}\left[\log Q(c'|x)\right]\right] + H(c) \quad (1)$$
$$= \mathcal{L}_I(G, Q)$$

- so instead of sample $p(x, c') = p(x)p(c'|x)$, we make it $p(x, c) = p(c)p(x|c)$:

$$\mathcal{L}_I(G, Q) = \mathbb{E}_{\underbrace{c \sim P(c)}_{\text{easy to sample!}},\, x\sim G(z,c)}[\log Q(c|x)] + H(c) \quad (2)$$

Why sample 1 and 2 are the same?

- sample 1 : $x \sim p(x)$, then sample $y|x$, then sample back $x'|y$. Finally, back and to compute $f(x', y)$:

$$
\begin{aligned}
\underbrace{E_{x\sim X, y\sim Y|x, x'\sim X|y}\left[f(x', y)\right]}_{1} &= \int_x p(x) \int_y p(y|x) \int_{x'} p(x'|y) f(x', y) \mathrm{d}x' \mathrm{d}y \mathrm{d}x \\
&= \int_y p(y) \int_x p(x|y) \int_{x'} p(x'|y) f(x', y) \mathrm{d}x' \mathrm{d}x \mathrm{d}y \\
&= \int_y p(y) \int_{x'} p(x'|y) f(x', y) \underbrace{\int_x p(x|y) \mathrm{d}x}_{=1} \mathrm{d}x' \mathrm{d}y \\
&= \int_y p(y) \int_x p(x|y) f(x, y) \mathrm{d}x \mathrm{d}y \\
&= \int_x p(x) \int_y p(y|x) f(x, y) \mathrm{d}y \mathrm{d}x \\
&= \underbrace{E_{x\sim X, y\sim Y|x}\left[f(x, y)\right]}_{2}
\end{aligned}
$$

- 2 : it has the same effect of sample (x, y) directly from $f(x, y)$, an then to compute $f(x, y)$

InfoGAN procedure

1. sample a noise $z \sim p(z)$ and $c \sim p(c)$
2. generate $\mathbf{x} = G(c, z)$
3. D differentiates real and fake as usual
4. calculate $Q(c|\mathbf{x})$

Bayesian GAN

- **Generator**

$$p(\theta_g|\mathbf{z}, \theta_d) \propto \left(\prod_{i=1}^{n_g} D_{\theta_d}\left(G_{\theta_g}(\mathbf{z}^{(i)}) \right) \right) p(\theta_g|\alpha)$$

- **Discriminator**

$$p(\theta_d|\mathbf{z}, \mathbf{X}, \theta_g) \propto \prod_{i=1}^{n_d} D_{\theta_d}(\mathbf{x}^{(i)}) \times \prod_{i=1}^{n_g} \left(1 - D_{\theta_d}(G_{\theta_g}(\mathbf{z}^{(i)}) \right) \times p(\theta_g|\alpha)$$

marginalization

- $p(\theta_g|\theta_d)$

$$p(\theta_g|\theta_d) = \int p(\theta_g, \mathbf{z}|\theta_d)\mathrm{d}\mathbf{z} = \int p(\theta_g|\mathbf{z}, \theta_d) \underbrace{p(\mathbf{z}|\theta_d)}_{\text{independent of } \theta_d} \mathrm{d}\mathbf{z}$$

$$= \int p(\theta_g|\mathbf{z}, \theta_d)p(\mathbf{z})\mathrm{d}\mathbf{z}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} p(\theta_g|\mathbf{z}^{(i)}, \theta_d) \qquad \mathbf{z}^{(i)} \sim p(\mathbf{z})$$

- $p(\theta_d|\theta_g)$

$$p(\theta_d|\theta_g) \equiv p(\theta_d|\mathbf{X}, \theta_g) = \int_{\mathbf{z}} p(\theta_d, \mathbf{z}|\mathbf{X}, \theta_g)\mathrm{d}\mathbf{z} = \int p(\theta_d|\mathbf{z}, \mathbf{X}, \theta_g) \underbrace{p(\mathbf{z}|\mathbf{X}, \theta_g)} \mathrm{d}\mathbf{z}$$

$$= \int_{\mathbf{z}} p(\theta_d|\mathbf{z}, \mathbf{X}, \theta_g)p(\mathbf{z})\mathrm{d}\mathbf{z}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} p(\theta_g|\mathbf{z}^{(i)}, \mathbf{X}, \theta_g) \qquad \mathbf{z}^{(i)} \sim p(\mathbf{z})$$