GANs 101

Aziz Temirkhanov

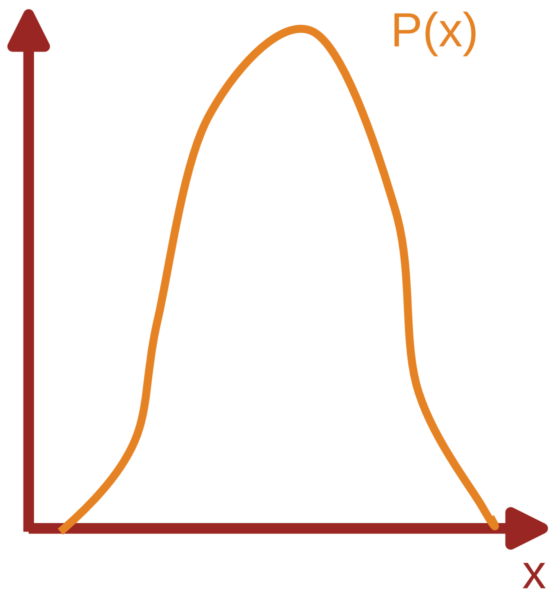
Laboratory for methods of big data analysis

LAMBDA • HSE

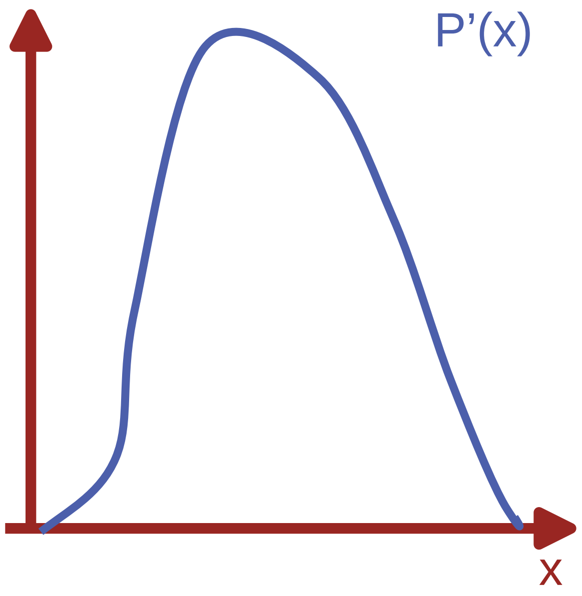
Fall 2023

Total Variation Distance

What we measure



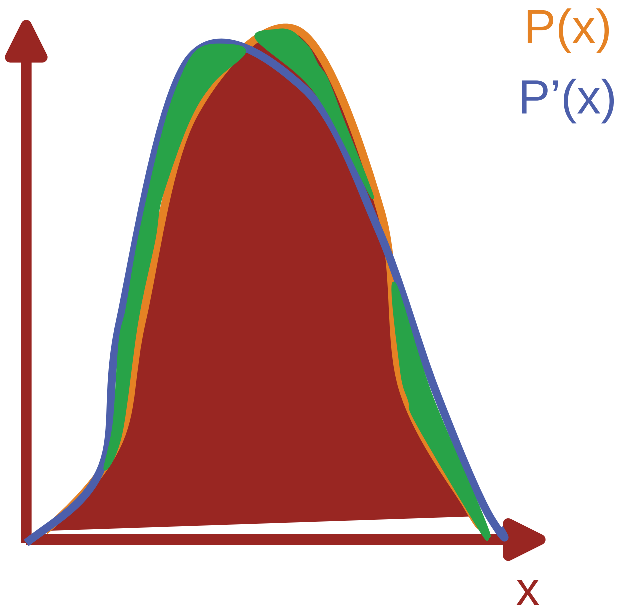
True Probability Density



Fitted Probability Density

P'(x) is similar to P(x)?

First idea: absolute difference



$$\int |P(x) - P'(x)|\, dx$$

Total Variation Distance

For $p(x)$ and $q_\theta(x)$ being PDFs:

$$D(p(x), q_\theta(x)) = \frac{1}{2}\int |p(x) - q_\theta(x)| dx$$

This can be rewritten using Scheffe's theorem

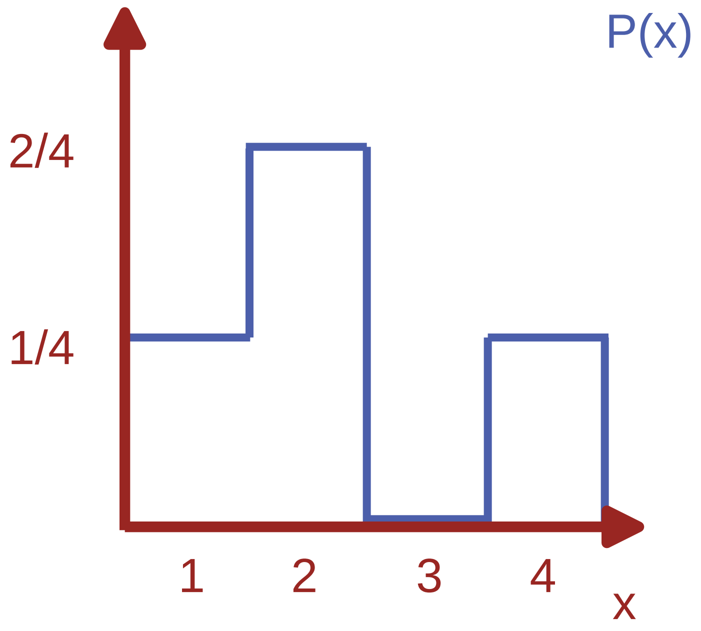
$$D(p(x), q_\theta(x)) = \sup_{\mathrm{A}} \left| \int_{\mathrm{A}} \mathrm{p(x)dx} - \int_{\mathrm{A}} \mathrm{q_\theta(x)dx} \right|$$

Where A is any measurable set.

A. B. Tsybakov, Introduction to Nonparametric Estimation, sec 2.4

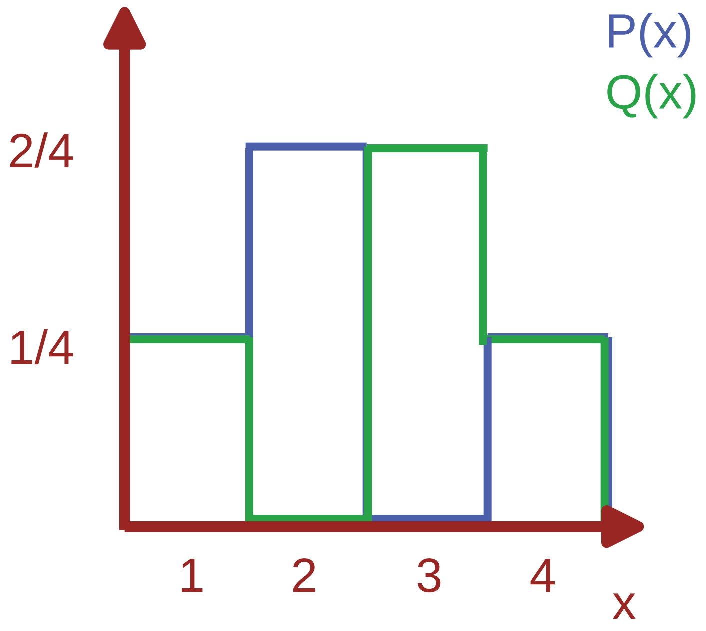
Total Variation Distance: example 1D

- discrete case for two PDFs



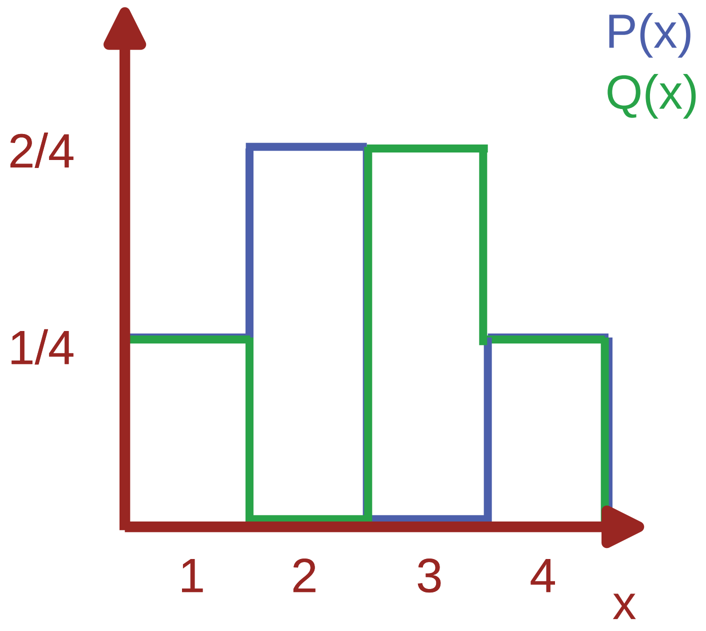
Total Variation Distance: example 1D

- discrete case for two PDFs



Total Variation Distance: example 1D

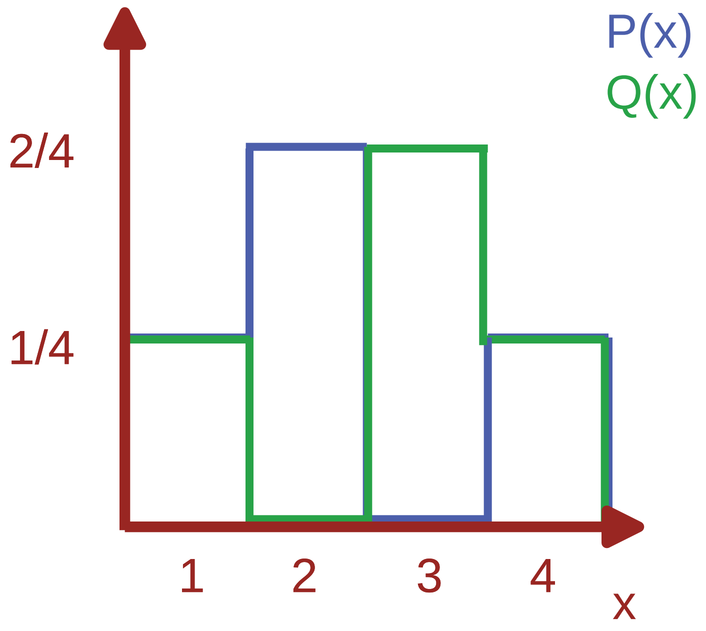
- discrete case for two PDFs
- calculate in two ways:



Total Variation Distance: example 1D

- discrete case for two PDFs
- calculate in two ways:
  - construct all possible subsets:

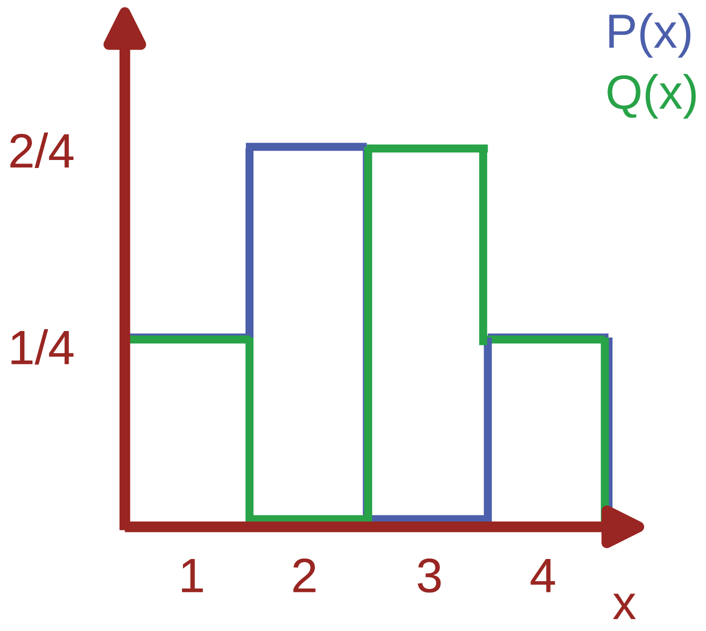
    {1}, {2}. {3}, {4}, {1;2}, {1;3}, {1;4}, {2;3}, {2;4}, {3;4}, {1;2;3}, {1;2;4}, {1;3;4}, {1,2,3,4}.



Total Variation Distance: example 1D

- discrete case for two PDFs
- calculate in two ways:
  - construct all possible subsets:

    D(p,q) = 0.5

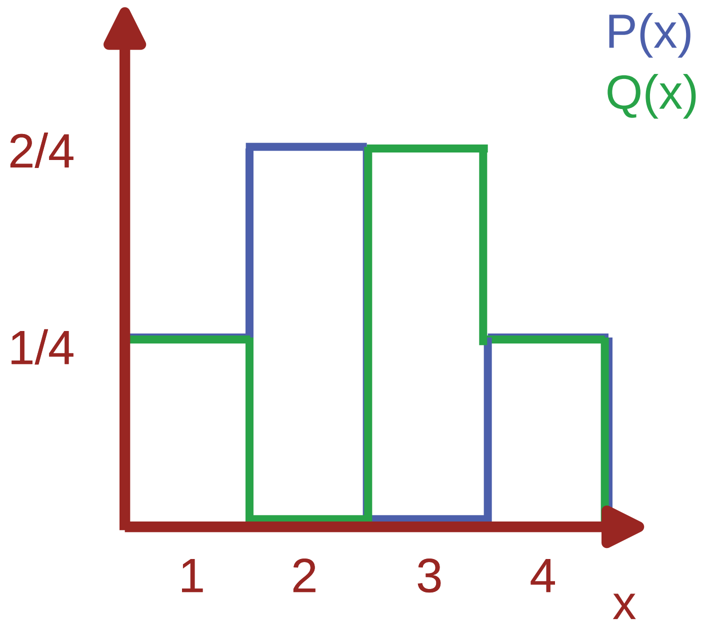


Total Variation Distance: example 1D

- discrete case for two PDFs
- calculate in two ways:
  - construct all possible subsets:

    D(p,q) = 0.5
  - integrate over full range:

    D(p,q) = 0.5



Total Variation Distance: observations

- Symmetric D(p, q) = D(q, p)
- Interpretable (using Scheffe lemma)
- Connected to hypothesis testing (D is the sum of errors)

Total Variation Distance: observations

- Symmetric D(p, q) = D(q, p)
- Interpretable (using Scheffe's theorem)
- Connected to hypothesis testing (D is the sum of errors)
- Too strong:

The distance might ignore the growing number of trials.

$X_1, \ldots, X_n \sim \pm 1, S_n = \sum_n X_i$. Than

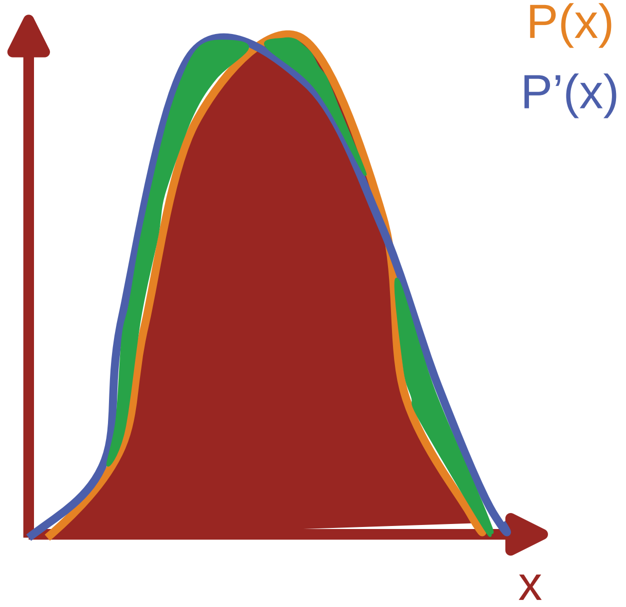
$$S_n/\sqrt{n} \to \mathcal{N}(0,1),$$

but $D(S_n, \mathcal{N}(0,1)) = 1$ for any n).

A. L. Gibbs, F. E. Su On Choosing and Bounding Probability Metrics
F Pollard, Total variation distance between measures

Kullback-Leibler Divergence

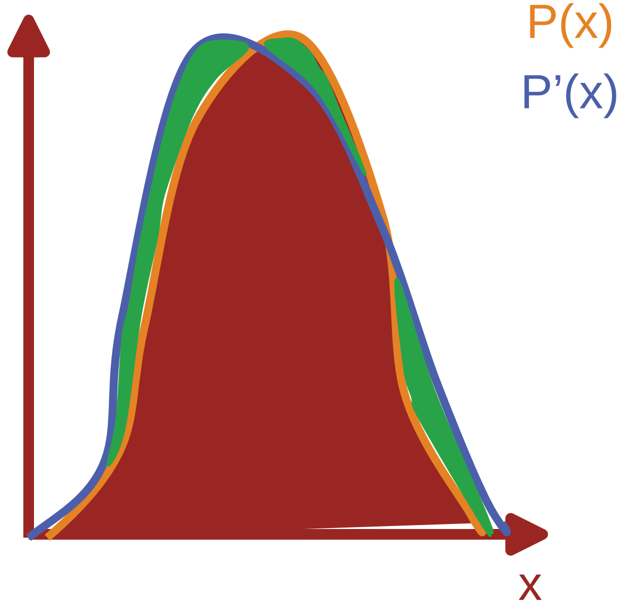
Kullback-Leibler divergence: ideas



Previously:

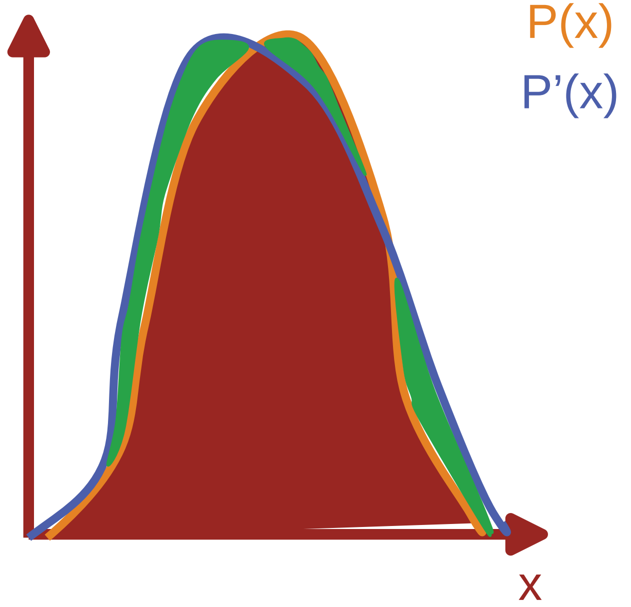
$$\int |P(x) - P'(x)| \, dx$$

Kullback-Leibler divergence: ideas



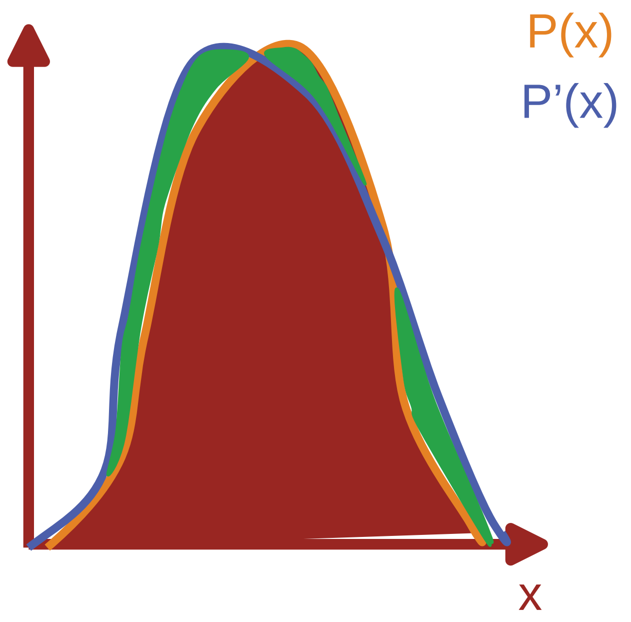
$$\frac{P(x)}{P'(x)}$$

Kullback-Leibler divergence: ideas



$$\ln \frac{P(x)}{P'(x)}$$

Kullback-Leibler divergence: ideas



$$\int P(x) \ln \frac{P(x)}{P'(x)} dx$$

Kullback-Leibler divergence: definition

For p(x) and q(x), two probability distributions,

$$KL(p||q_\theta) = \int p(x)\log\left(\frac{p(x)}{q_\theta(x)}\right)dx$$

Kullback-Leibler divergence: definition

For p(x) and q(x), two probability distributions,

$$KL(p||q_\theta) = \int p(x) \log\left(\frac{p(x)}{q_\theta(x)}\right) dx$$

- not symmetric $KL(P||Q) \neq KL(Q||P)$
- invariant under change of variables
- additive for independent variables
- nonnegative

Kullback-Leibler divergence: observations

- **KL divergence is connected to cross-entropy:**

$$KL(p||q) = H(p) + H(p,q),$$

where $H(p,q) = \mathbb{E}_p(\log q)$.

KL and Maximum Likelihood

Find the optimal parameter, θ^* :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} KL(p(x)||q_\theta(x))$$

KL and Maximum Likelihood

Find the optimal parameter, θ^* :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} KL(p(x)||q_\theta(x))$$

$$= \underset{\theta}{\operatorname{argmin}}(\mathbb{E}_{x\sim p}[\log p(x)] - \mathbb{E}_{x\sim p}[\log q_\theta(x)])$$

KL and Maximum Likelihood

Find the optimal parameter, θ^* :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} KL(p(x)||q_\theta(x))$$

$$= \underset{\theta}{\operatorname{argmin}} (\mathbb{E}_{x\sim p}[\log p(x)] - \mathbb{E}_{x\sim p}[\log q_\theta(x)])$$

$$= -\underset{\theta}{\operatorname{argmin}} \mathbb{E}_{x\sim p}[\log q_\theta(x)]$$

KL divergence: observations

- **KL divergence is connected to cross-entropy:**

$$KL(p||q) = H(p) + H(p, q),$$

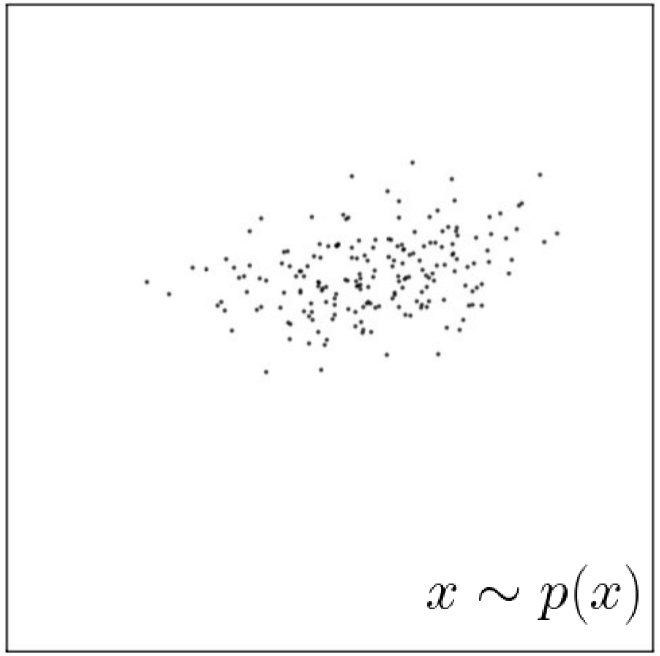
where $H(p, q) = \mathbb{E}_p(\log q)$.

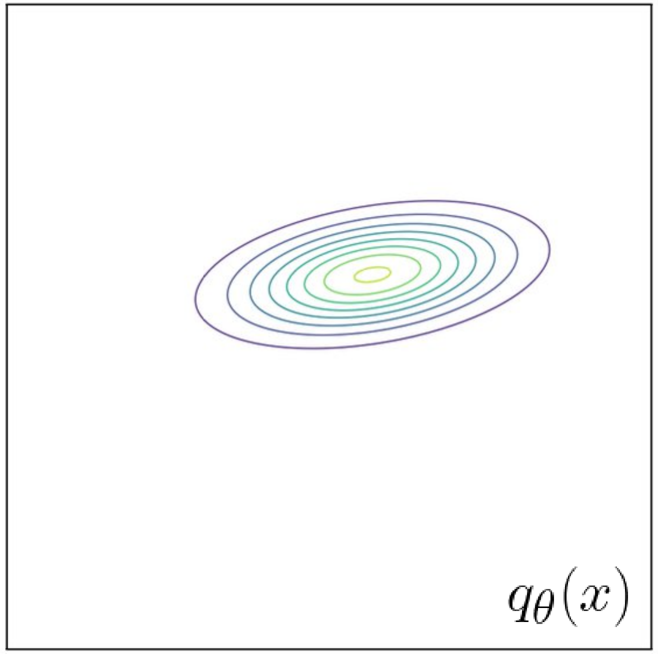
- **Minimizing KL divergence is equivalent to maximizing the likelihood.**

$$\theta^* = \underset{\theta}{\operatorname{argmin}} KL(p(x)||q_\theta(x)) = \underset{\theta}{\operatorname{argmax}} \mathcal{L}(q_\theta(x); x)$$

Using in fits

Fit data points from 2D Gaussian function

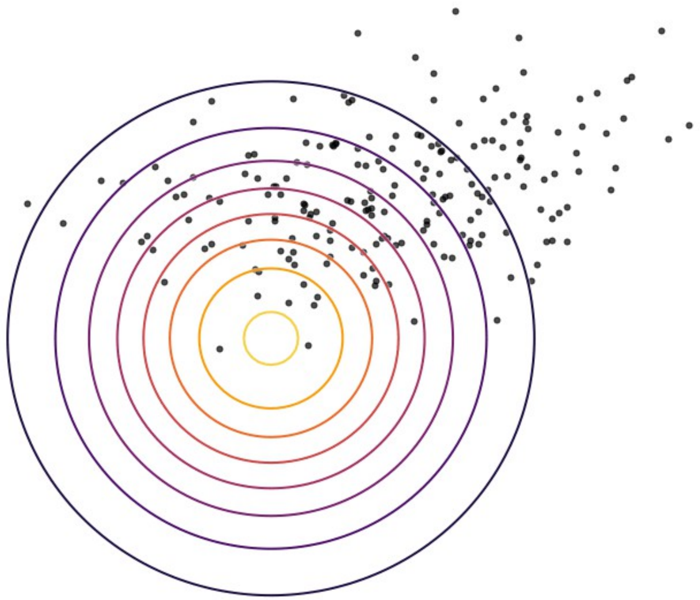




…with 2D Gaussian function

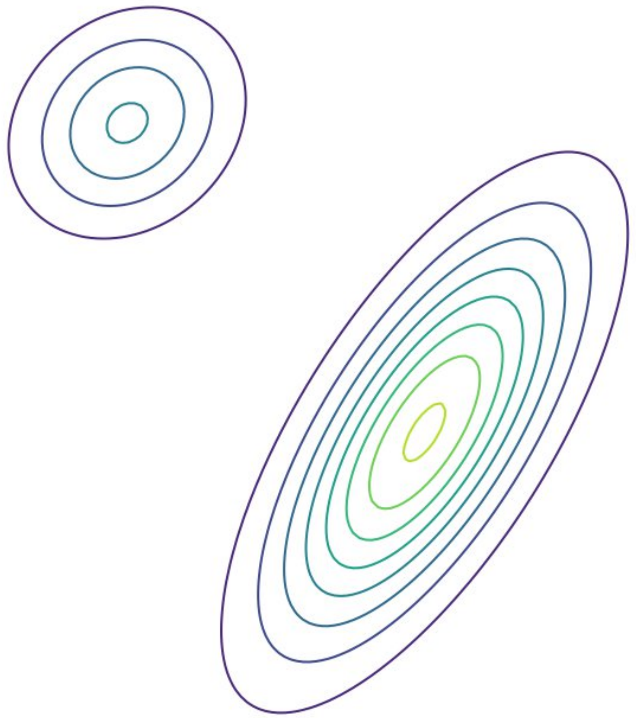
Here and Later: Colin Raffel's blog

Using in fits



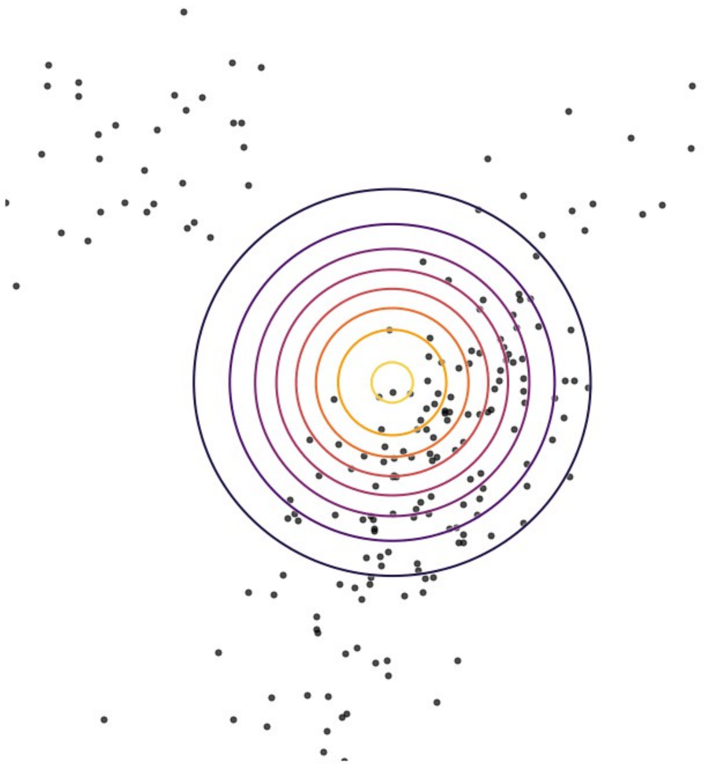
- Runs smoothly for simple data

Using in fits: Multimodal data



- Runs smoothly for simple data

Using in fits: Multimodal data

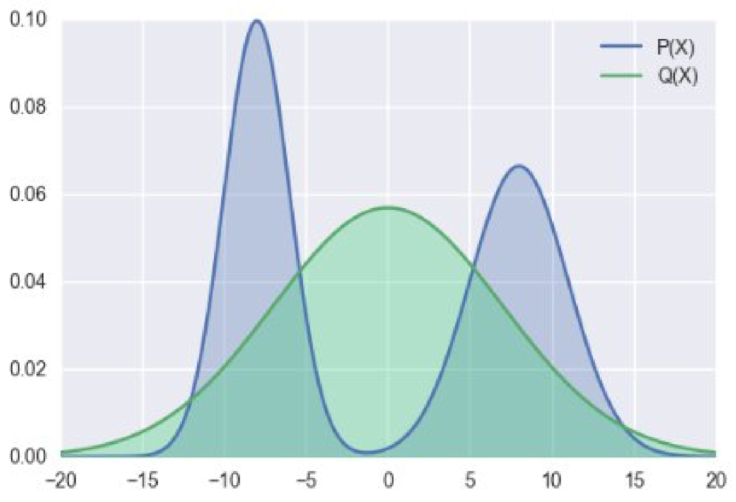


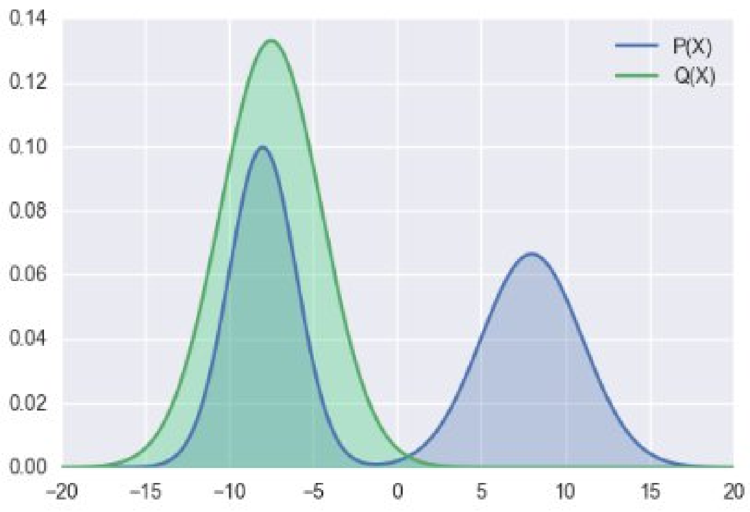
- Runs smoothly for simple data
- Problems for multimodal data
- Covers significant amount of empty spaces

KL divergence: study

$$KL(p||q_\theta) = \int p(x) \log \left(\frac{p(x)}{q_\theta(x)} \right) dx$$

$$KL(q_\theta||p) = \int q_\theta(x) \log \left(\frac{q_\theta(x)}{p(x)} \right) dx$$





KL is zero avoiding, as it is avoiding q(x) = 0 whenever p(x) > 0

Reverse KL is zero forcing, as it forces q(X) to be 0 on some areas, even if p(X) > 0

Agustinus Kristiadi's Blog

Reverse KL divergence: fits

Find the optimal parameter, θ^* :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} KL(q_\theta(x)||p(x))$$

Reverse KL divergence: fits

Find the optimal parameter, θ^* :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} KL(q_\theta(x)||p(x))$$

$$= \underset{\theta}{\operatorname{argmin}} (\mathbb{E}_{\tilde{x}\sim q_\theta}[\log q_\theta(x)] - \mathbb{E}_{\tilde{x}\sim q_\theta}[\log p(x)])$$

Reverse KL divergence: fits

Find the optimal parameter, θ^* :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} KL(q_\theta(x)||p(x))$$

$$= \underset{\theta}{\operatorname{argmin}}(\mathbb{E}_{\tilde{x}\sim q_\theta}[\log q_\theta(x)] - \mathbb{E}_{\tilde{x}\sim q_\theta}[\log p(x)])$$

$$= \underset{\theta}{\operatorname{argmax}}(-\mathbb{E}_{\tilde{x}\sim q_\theta}[\log q_\theta(x)] + \mathbb{E}_{\tilde{x}\sim q_\theta}[\log p(x)])$$

Reverse KL divergence: fits

Find the optimal parameter, θ^* :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} KL(q_\theta(x)||p(x))$$

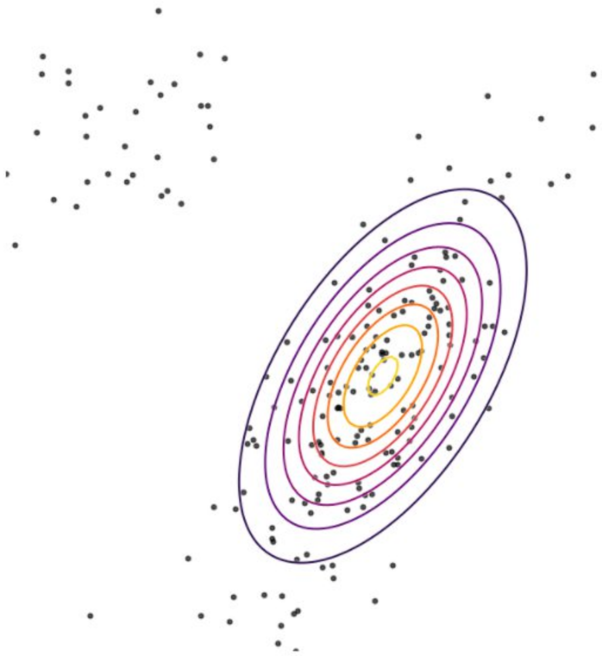
entropy for the
fitted model

$$= \underset{\theta}{\operatorname{argmax}}(-\mathbb{E}_{\tilde{x}\sim q_\theta}[\log q_\theta(x)] + \mathbb{E}_{\tilde{x}\sim q_\theta}[\log p(x)])$$

relation between fitted
and generated

Reverse KL divergence: fits

- $q_\theta(x)$ covers only regions with data
- reasonable in multi-modal data for one solution



Critical: we do not have direct access to p(x).

Jensen-Shannon Divergence

Jensen-Shannon Divergence: idea

- KL divergence is asymmetric

Jensen-Shannon Divergence: idea

- KL divergence is asymmetric

$$KL(p||q) + KL(q||p)$$

Jensen-Shannon Divergence: idea

- KL divergence is asymmetric

- KL can become infinite

$$KL(p||q) + KL(q||p)$$

Jensen-Shannon Divergence: idea

- KL divergence is asymmetric
- KL can become infinite

$$KL(p(x)||\frac{p(x)+q_\theta(x)}{2})+KL(q_\theta(x)||\frac{p(x)+q_\theta(x)}{2})$$

Jensen-Shannon Divergence: Definition

For p(x) and q(x), two probability distributions,

$$JS(p,q) = \frac{1}{2}\left(KL(p(x)||\frac{p(x)+q_\theta(x)}{2}) + KL(q_\theta(x)||\frac{p(x)+q_\theta(x)}{2})\right)$$

- symmetric
- nonnegative $0 \leq JS(P,Q) \leq \ln(2)$
- can be transformed to a true distance $\sqrt{JS(p,q)}$

J. Lin Divergence measures based on the Shannon entropy

f-divergences

Definition

- Let $f:(0;\infty) \to \mathbb{R}$ be a convex function with f(1) = 0.
- P and Q - two probability distributions on a measurable space $(\mathscr{X}, \mathscr{F})$.
- p and q - absolutely continuous with respect to a base measure dx defined on $\mathscr{X}$.
- f-divergence is defined:

$$D_f(P||Q) = \int q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

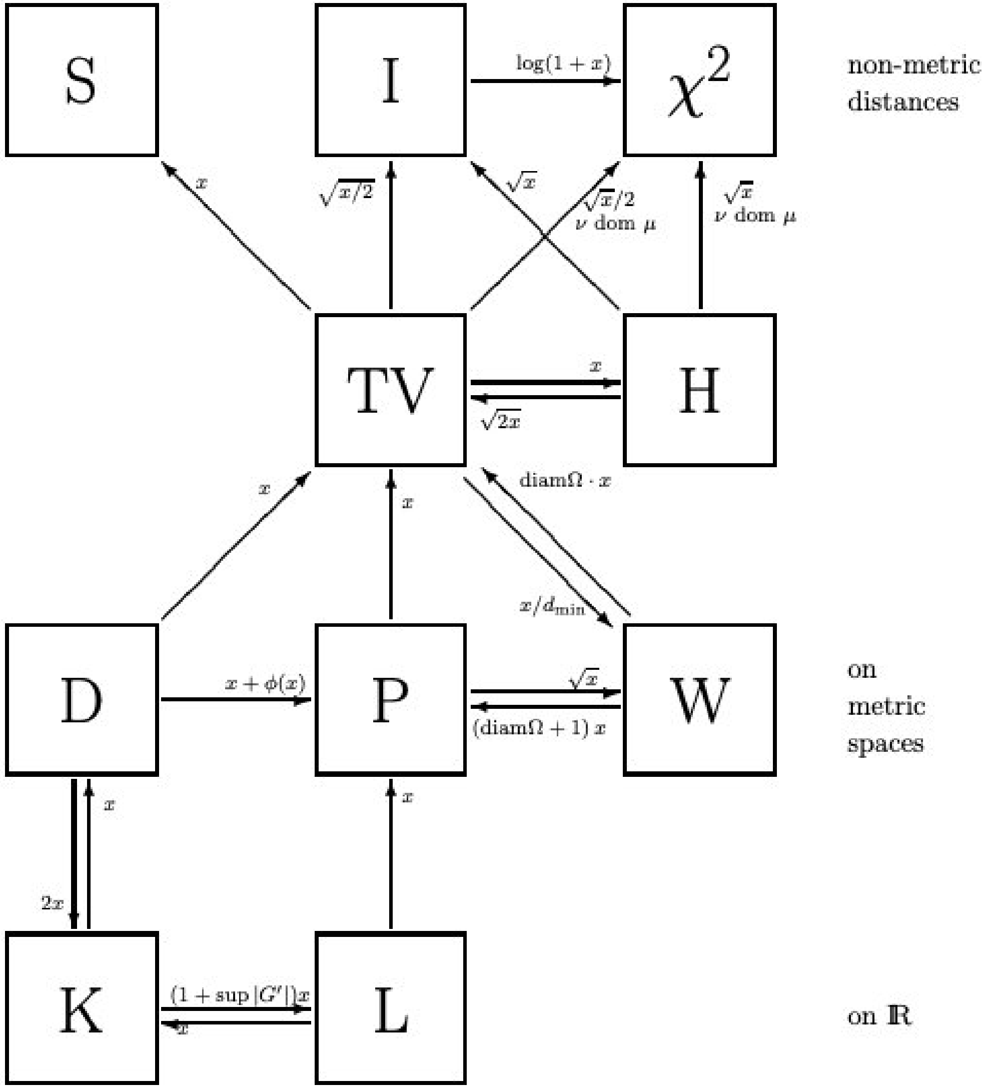
- f is called generator.

Examples

Name	$D_f(P\|Q)$	Generator $f(u)$
Total variation	$\frac{1}{2}\int \lvert p(x)-q(x)\rvert \,\mathrm{d}x$	$\frac{1}{2}\lvert u-1\rvert$
Kullback-Leibler	$\int p(x)\log\frac{p(x)}{q(x)}\,\mathrm{d}x$	$u\log u$
Reverse Kullback-Leibler	$\int q(x)\log\frac{q(x)}{p(x)}\,\mathrm{d}x$	$-\log u$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)}\,\mathrm{d}x$	$(u-1)^2$
Neyman χ^2	$\int \frac{(p(x)-q(x))^2}{q(x)}\,\mathrm{d}x$	$\frac{(1-u)^2}{u}$
Squared Hellinger	$\int \left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2\mathrm{d}x$	$(\sqrt{u}-1)^2$
Jeffrey	$\int (p(x)-q(x))\log\left(\frac{p(x)}{q(x)}\right)\mathrm{d}x$	$(u-1)\log u$
Jensen-Shannon	$\frac{1}{2}\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x$	$-(u+1)\log\frac{1+u}{2}+u\log u$
Jensen-Shannon-weighted	$\int p(x)\pi\log\frac{p(x)}{\pi p(x)+(1-\pi)q(x)}+(1-\pi)q(x)\log\frac{q(x)}{\pi p(x)+(1-\pi)q(x)}\,\mathrm{d}x$	$\pi u\log u-(1-\pi+\pi u)\log(1-\pi+\pi u)$

f-divergence inequalities

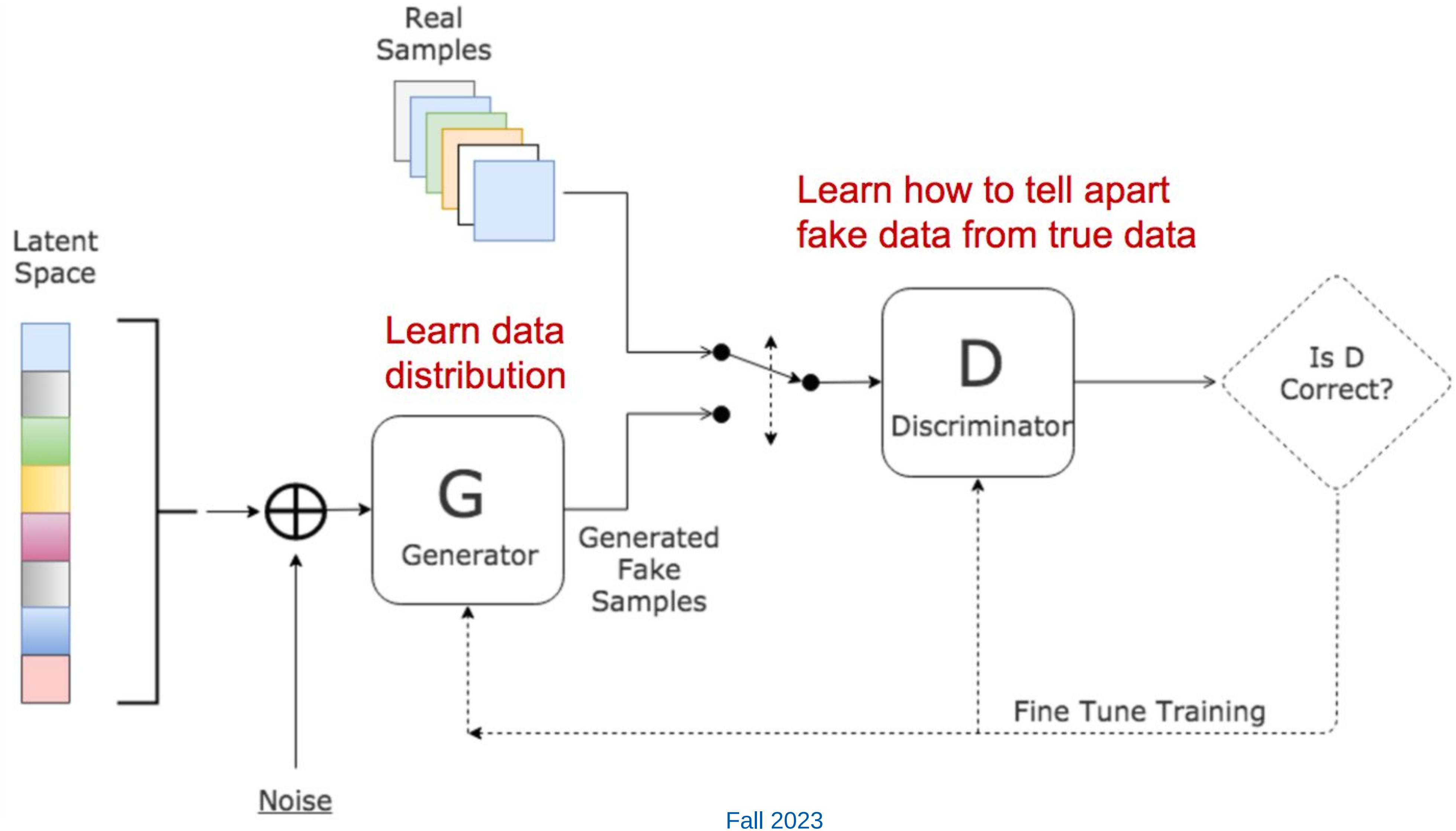
Abbreviation	Metric
D	Discrepancy
H	Hellinger distance
I	Relative entropy (or Kullback-Leibler divergence)
K	Kolmogorov (or Uniform) metric
L	Lévy metric
P	Prokhorov metric
S	Separation distance
TV	Total variation distance
W	Wasserstein (or Kantorovich) metric
χ^2	χ^2 distance



A. L. Gibbs, F. E. Su On Choosing and Bounding Probability Metrics

Idea

Idea



Real
Samples
Learn how to tell apart
fake data from true data
Latent
Space
Learn data
distribution
D
Discriminator
Is D
Correct?
G
Generator
Generated
Fake
Samples
Fine Tune Training
Noise

Generator

- G_θ is a **generator**. It should sample from a random noise:

$$p_z \sim N(0;1);$$

$$x_j = G_\theta(p_z).$$

- Our aim is G_θ as a neural network.
- We thus have a sample:

$$\{x_j\} \sim p_g(x)$$

- G_θ can be defined in many ways. For example, physics generator.

Borisyak M et al. Adaptive divergence for rapid adversarial optimization. *PeerJ Computer Science* 6:e274 (2020)

Discriminator

- Add a classifying neural network, **discriminator** D_ϕ, to distinguish between the real and generated samples.
- Optimize:

$$\max_{\phi} \Big(\mathbb{E}_{x \sim p_r(x)}[\log D(x)]\Big) + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]\Big)$$

Real samples

Generated samples

GAN minimax game

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

- Two models are trained to find a Nash equilibrium to a two-player non-cooperative game

Optimal Solution

- Let us define an optimal D:

$$L(G, D) = \int_x \left(p_r(x) \log(D(x)) + p_g(x) \log(1 - D(x)) \right) dx$$

- Then, label:

$$\tilde{x} = D(x), A = p_r(x), B = p_g(x)$$

- Now, inside the integral

$$f(\tilde{x}) = Alog\tilde{x} + Blog(1 - \tilde{x})$$

$$\frac{df(\tilde{x})}{d\tilde{x}} = A\frac{1}{ln10}\frac{1}{\tilde{x}} - B\frac{1}{ln10}\frac{1}{1-\tilde{x}} = \frac{1}{ln10}(\frac{A}{\tilde{x}} - \frac{B}{1-\tilde{x}}) = \frac{1}{ln10}\frac{A-(A+B)\tilde{x}}{\tilde{x}(1-\tilde{x})}$$

Thus, set $\frac{df(\tilde{x})}{d\tilde{x}} = 0$, we get the best value of the discriminator:
$D^*(x) = \tilde{x}^* = \frac{A}{A+B} = \frac{p_r(x)}{p_r(x)+p_g(x)} \in [0, 1].$

Optimal Solution

- When both G and D are at their optimal values, we have $p_g = p_r$ and $D^*(x) = 1/2$, and the loss function becomes:

$$
\begin{aligned}
L(G, D^*) &= \int_x \bigg(p_r(x) \log(D^*(x)) + p_g(x) \log(1 - D^*(x)) \bigg) dx \\
&= \log \frac{1}{2} \int_x p_r(x) dx + \log \frac{1}{2} \int_x p_g(x) dx \\
&= -2 \log 2
\end{aligned}
$$

Optimal Solution

- Recall the JS divergrence from the beginning

$$D_{JS}(p\|q) = \frac{1}{2}D_{KL}(p\|\frac{p+q}{2}) + \frac{1}{2}D_{KL}(q\|\frac{p+q}{2})$$

- Thus, JS divergence between p_g and p_r can be computed as:

$$\begin{aligned}
D_{JS}(p_r\|p_g) =& \frac{1}{2}D_{KL}(p_r||\frac{p_r+p_g}{2}) + \frac{1}{2}D_{KL}(p_g||\frac{p_r+p_g}{2}) \\
=& \frac{1}{2}\left(\log 2 + \int_x p_r(x)\log\frac{p_r(x)}{p_r+p_g(x)}dx\right) + \\
& \frac{1}{2}\left(\log 2 + \int_x p_g(x)\log\frac{p_g(x)}{p_r+p_g(x)}dx\right) \\
=& \frac{1}{2}\left(\log 4 + L(G, D^*)\right)
\end{aligned}$$

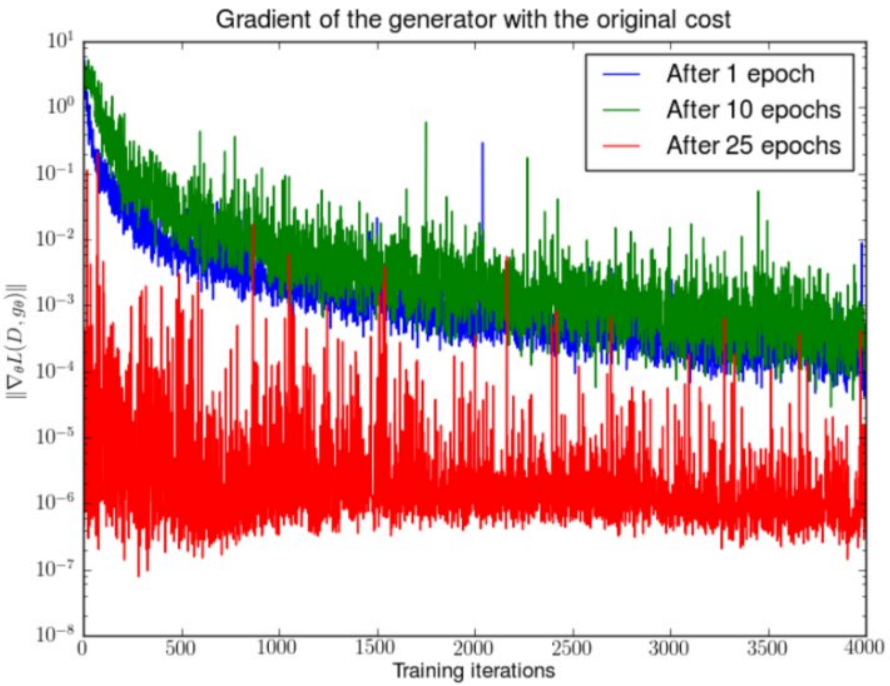
Optimal Solution

- Thus, $L(G, D^*) = 2D_{JS}(p_r \| p_g) - 2\log 2$
- The best generator *G** must yield a perfect replication of real data, which leads to the minimum of $L(G^*, D^*) = -2\log 2$

Problems

Game Approach Problems

- Disctiminator must be optimal at every step of convergence
- But if is true, loss function falls to zero, and we end up with no gradient to update loss during learning iterations

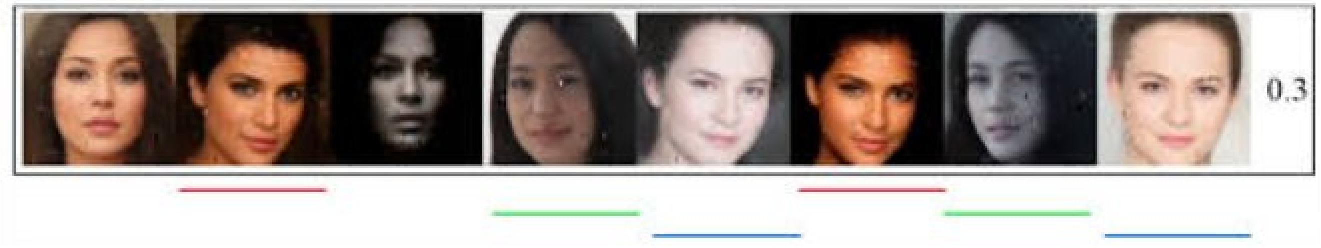


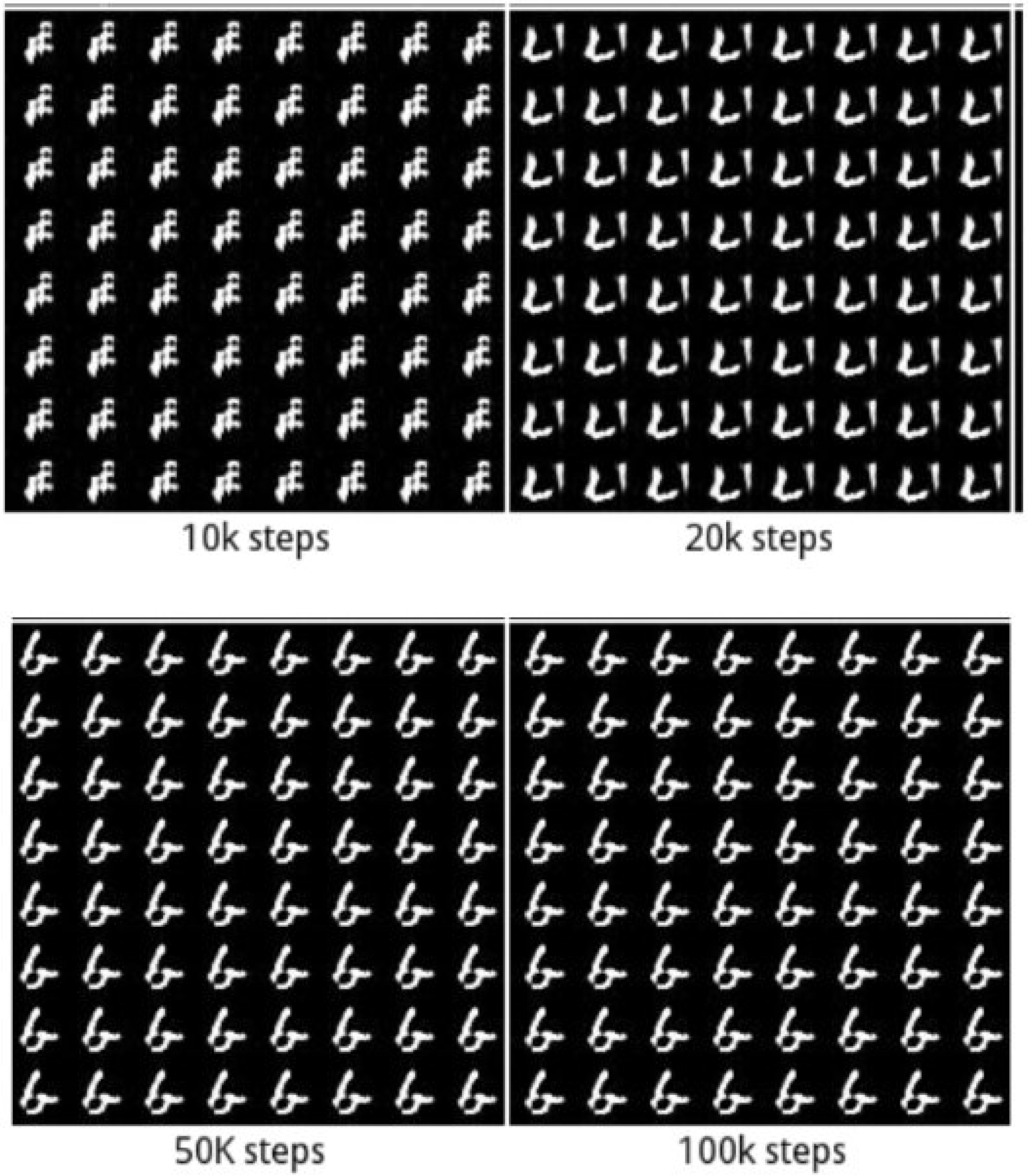
Martin Arjovsky, Towards Principled Methods for Training Generative Adversarial Networks , ICLR17

Mode Collapse

- GANs choose to generate a small number of modes due to a defect in the training procedure, rather than due to the divergence they aim to minimize.

I. Goodfellow NIPS 2016 Tutorial: Generative Adversarial Network



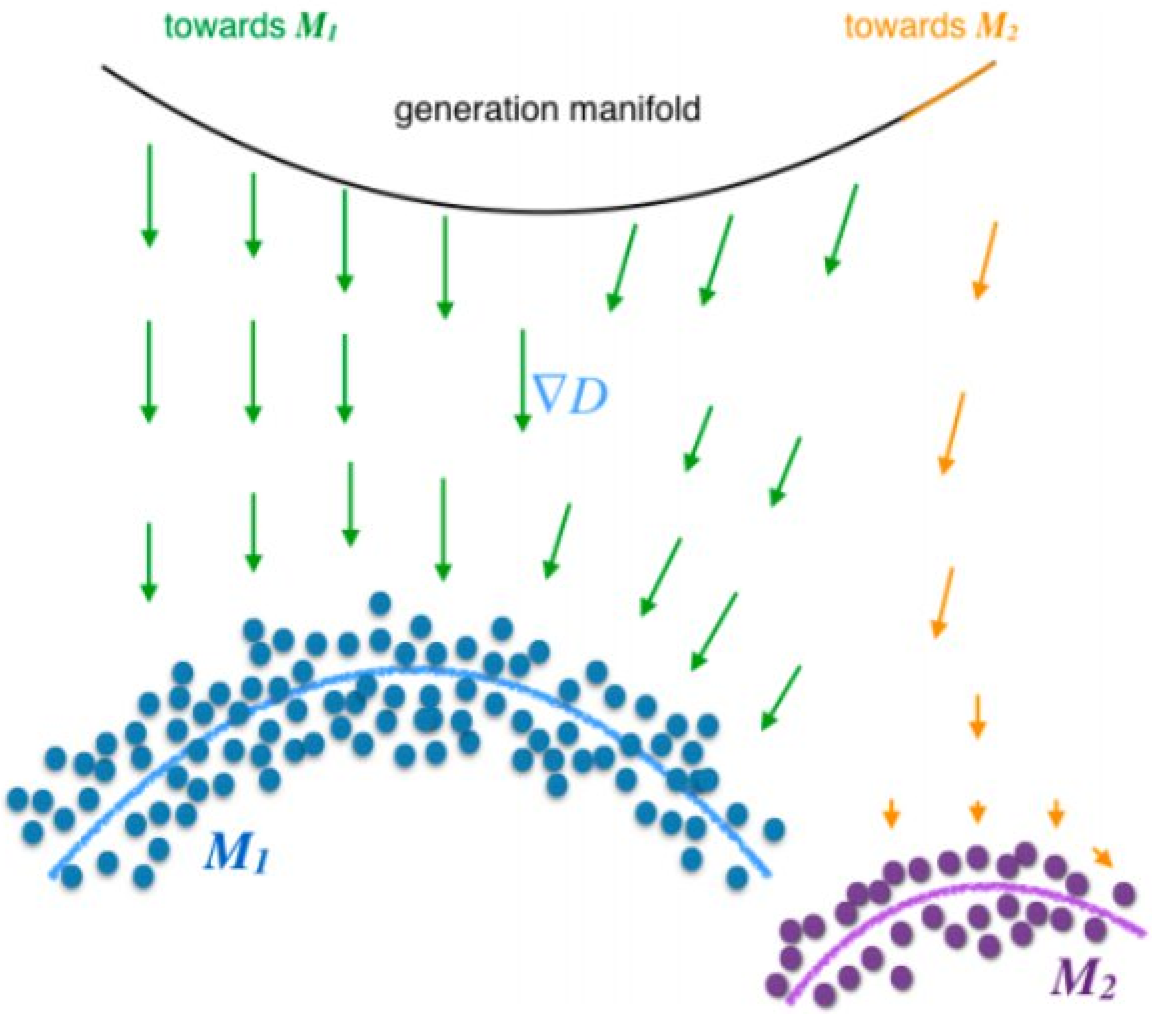


Luke Metz et al Unrolled Generative Adversarial Networks ICLR 2017

Mode Collapse

- For fixed D:
  - G tends to converge to a point x^* that fools D the most.
  - In extreme cases, G becomes independent on z.
  - Gradient on z diminishes.

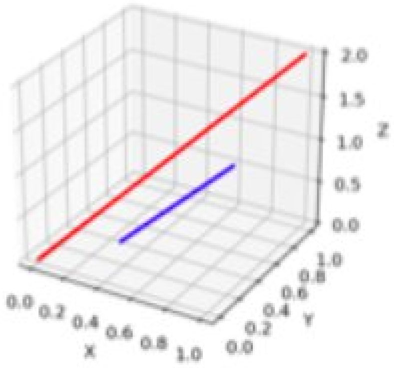
- When D restarts:
  - Easily finds this x^*.
  - Pushes G to the next point x^{**}.

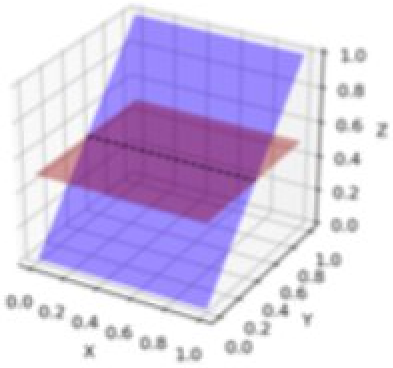


T. Che Mode Regularized Generative Adversarial Networks ICLR 2017

Diminishing Gradients

- We have seen already that signal data is located on manifold.
- GAN case is in fact more complicated, as we need a discriminator that distinguishes two supports.
- This is way too easy, if supports are disjoint.





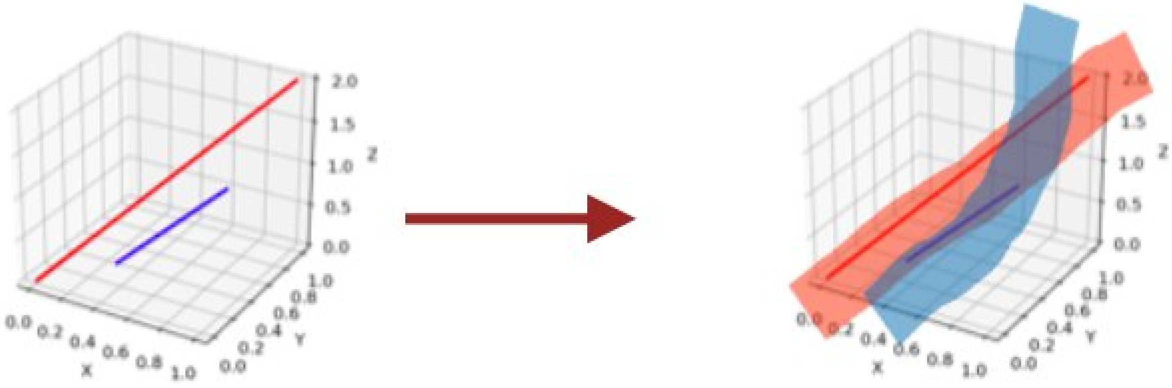
Solutions

Diminishing Gradients: Noisy Supports

- Let's make the problem harder: introduce random noise $\varepsilon \sim N(0;\sigma^2 I)$:

$$\mathbb{P}_{x+\varepsilon(x)} = \mathbb{E}_{y\sim P(x)}\mathbb{P}_{\varepsilon}(x-y).$$

- This will make noisy supports, that makes it difficult for discriminator.



Martin Arjovsky, Towards Principled Methods for Training Generative Adversarial Networks , ICLR17

Feature Matching

- Change the objective of the generator:

$$||\mathbb{E}_{x\sim p(x)} f(x) - \mathbb{E}_{z\sim p_z(z)} f(G(z))||^2$$

- Here $f(x)$ can be any property we need (including the output of another network.

Danger of overtrain to match known tests!

Historical Averaging

- average with previous parameter values:

$$||\theta - \frac{1}{t}\sum_{i=1}^{t}\theta[i]||^2$$

- this allows to create a fake agent that plays the game.
- and solves the problems only in low dimensions.

One-sided Label Smoothing

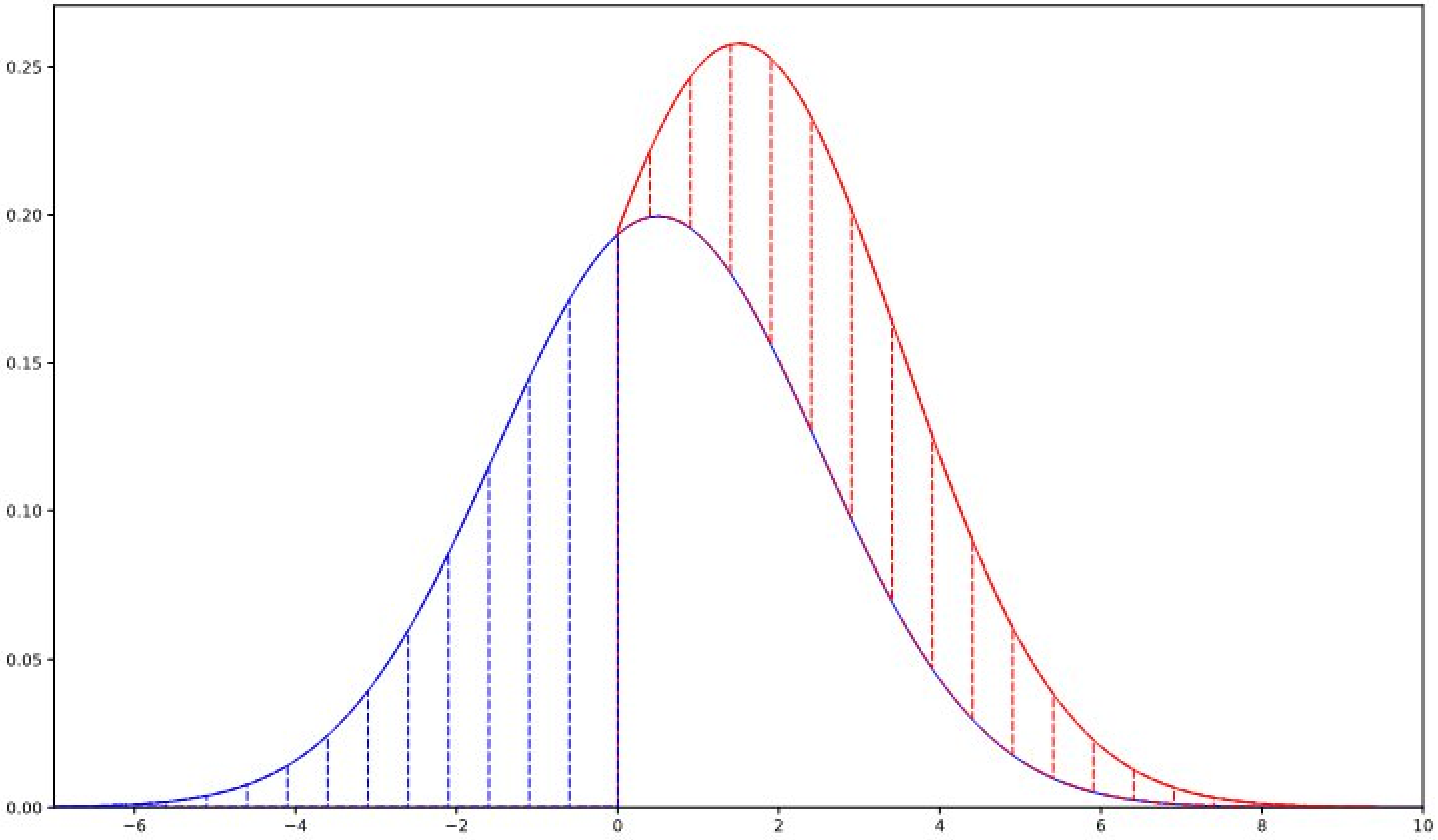
- When feeding the discriminator, instead of providing 1 and 0 labels, use soften values such as 0.9 and 0.1. It is shown to reduce the networks' vulnerability.

Wasserstein GAN

Wasserstein distance

Also called "Earth mover's distance" (EMD)

- Distributions $P(x)$ and $Q(x)$ are viewed as describing the **amounts of "dirt" at point x**
- We want to convert one distribution into the other by **moving around** some amounts of dirt



- The cost of moving an amount m from x_1 to x_2 is $m \times \|x_2 - x_1\|$
- $\mathrm{EMD}(P, Q)$ = **minimum total cost** of converting P into Q

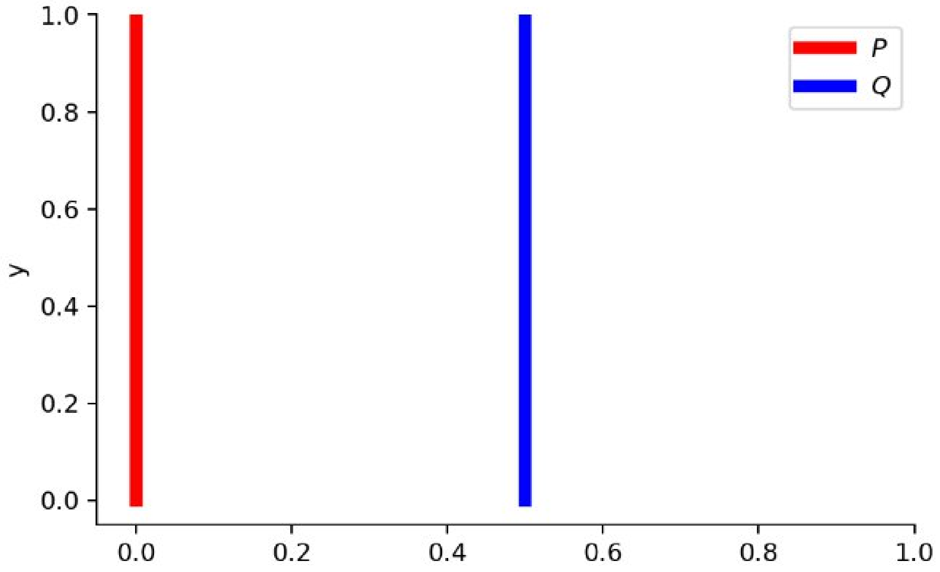
Wasserstein distance

For continuous case, there are a set of p-Wasserstein distances, with $W_p(p_x, q_y)$ defined with $x \in M, y \in M$ and a distance D on x, y:

$$W_p(p_x, q_y) = \inf_{\gamma \in \Pi(x,y)} \int_{M \times M} D(x, y)^p d\gamma(x, y),$$

where $\Pi(x, y)$ is a set of all joint distributions having p_x, q_y as their marginals.

Why Wasserstrin?

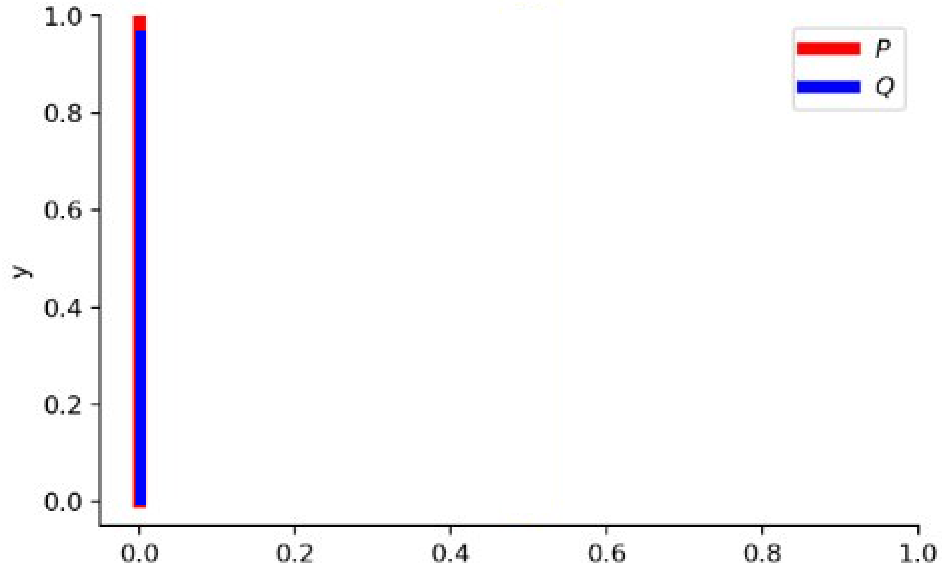


$$D_{KL}(P\|Q) = \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{KL}(Q\|P) = \sum_{x=\theta, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{JS}(P,Q) = \frac{1}{2}\left(\sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2} + \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2}\right) = \log 2$$

$$W(P,Q) = |\theta|$$



$$D_{KL}(P\|Q) = D_{KL}(Q\|P) = D_{JS}(P,Q) = 0$$

$$W(P,Q) = 0 = |\theta|$$

Wasserstein distance as loss function

- It is intractable to exhaust all the possible joint distributions in $\Pi(p_r, p_g)$ to compute $\inf_{\gamma \sim \Pi(p_r, p_g)}$.
- But by applying Kantorovich-Rubinshtein duality:

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

- In that case, fucntion *f* must be K-Lipschitz continuous.

https://vincentherrmann.github.io/blog/wasserstein/

Lipschitz continuity

- A real-valued fucntion $f : \mathbb{R} \to \mathbb{R}$ is called *K*-Lipschitz continuous if there exists a real contstant $K \geq 0$ such that, for all $x_1, x_2 \in \mathbb{R}$,

$$|f(x_1) - f(x_2)| \leq K|x_1 - x_2|$$

- Then,

$$L(p_r, p_g) = W(p_r, p_g) = \max_{w \in W} \mathbb{E}_{x \sim p_r}[f_w(x)] - \mathbb{E}_{z \sim p_r(z)}[f_w(g_\theta(z))]$$

WGAN Algorithm

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
    for $t = 0, ..., n_{\text{critic}}$ **do**
        Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
        Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
        $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
        $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
        $w \leftarrow \text{clip}(w, -c, c)$
    end for
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
    $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

FID

- FID score is the distance between the distribution of the activations for some deep layers in a classifier, when comparing a sample of test images and one of generated images. If activation distributions are similar, we can conclude the underlying image distributions are also alike.

$$\mathrm{FID} = ||\mu - \mu_w||_2^2 + \mathrm{tr}(\Sigma + \Sigma_w - 2(\Sigma^{1/2}\Sigma_w\Sigma^{1/2})^{1/2}).$$

In conclusion

- GANs utilize simple yet powerful idea from game theory
- In practice, it is flawful, but some of the flaws may be mitigated:
  - Noisy supports, smoothed input, and better mertrics can help