Generative Adversarial Networks for Mathematicians

Ángel González Prieto

Universidad Complutense de Madrid

Joint work with A. Mozo, S. Gómez-Canaval y E. Talavera

New Bridges between Mathematics and Data Science

Generative Models

Let $X : \Omega \to \mathbb{R}^d$ be a d-dimensional random vector.

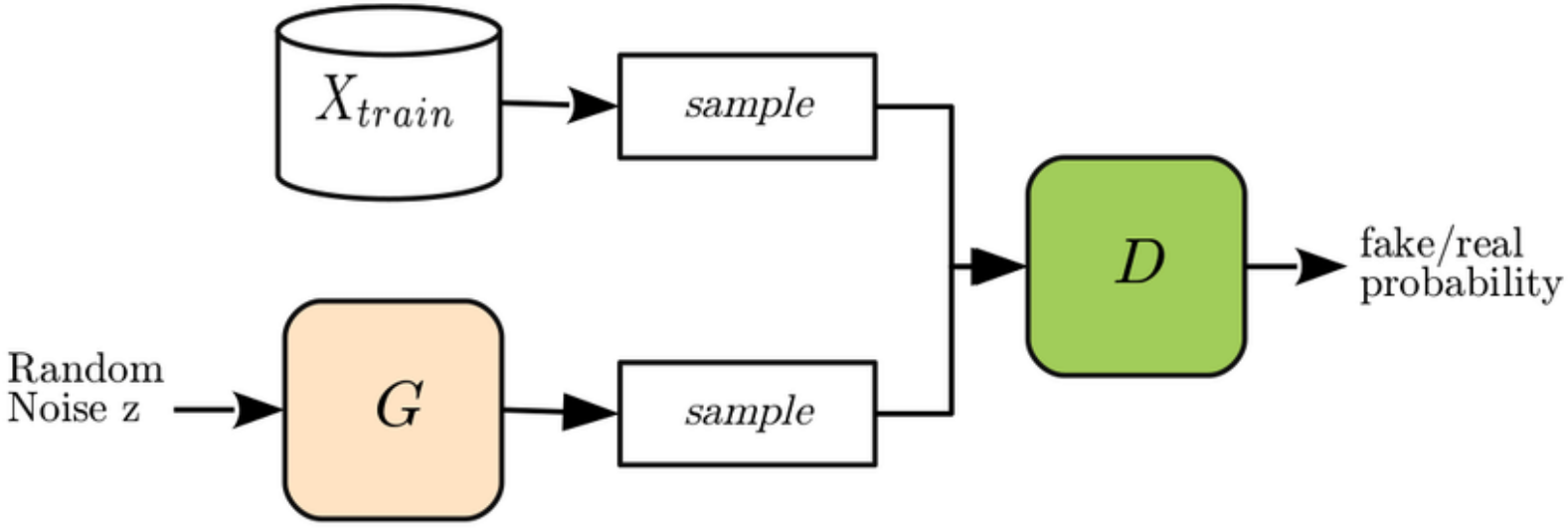
Problem: Generate 'new samples' of X of high quality but different from anything known.

The GAN game

Choose two families of functions (e.g. neural networks)

$$D : \mathbb{R}^d \times \Theta_D \to \mathbb{R}, \quad G : \Lambda \times \Theta_G \to \mathbb{R}^d,$$

called the *discriminator* and the *generator*, respectively. Here, Λ is a probability space, called the **latent space**.



Training GANs as an optimization problem

The cost functional

The success of the discriminator $D_{\theta_D} = D(-, \theta_D)$ is

$$\mathcal{F}(\theta_D, \theta_G) = \mathbb{E}_\Omega \log\left(D_{\theta_D}(X)\right) + \mathbb{E}_\Lambda \log\left(1 - D_{\theta_D}(G_{\theta_G})\right).$$

Discriminator: $D_{\theta_D}(x) = 1$ means real and $D_{\theta_D}(x) = 0$ means fake.

In practice you know neither X nor the distribution of $G_{\theta_G} : \Lambda \to \mathbb{R}^d$, so you estimate the cost functional by sampling

$$\mathcal{F}(\theta_D, \theta_G) = \frac{1}{N}\sum_{i=1}^{N} \log\left(D_{\theta_D}(x_i)\right) + \frac{1}{M}\sum_{j=1}^{M} \log\left(1 - D_{\theta_D}(G_{\theta_G}(\lambda_j))\right), \quad x_i \sim X, \lambda_i \sim \Lambda.$$

The GAN competition

Goal: To find a Nash equilibrium for the game

$$\min_{\theta_G} \max_{\theta_D} \mathcal{F}(\theta_D, \theta_G) = \min_{\theta_G} \max_{\theta_D} \left(\mathbb{E}_\Omega \log\left(D_{\theta_D}(X)\right) + \mathbb{E}_\Lambda \log\left(1 - D_{\theta_D}(G_{\theta_G})\right)\right).$$

When probability met GANs

Suppose that X and G are continuous random variables with density functions f_X and f_G.

$$\mathcal{F}(D, G) = \int_{\mathbb{R}} \log\left(D(s)\right) f_X(s) + \log\left(1 - D(s)\right) f_G(s)\, ds.$$

Lemma

For fixed G the optimal generator is

$$D^*(s) = \frac{f_X(s)}{f_X(s) + f_G(x)}.$$

Proof: Maximize the function $D \mapsto \log\left(D\right) f_X(s) + \log\left(1 - D\right) f_G(s)$.

At this optimal value, we have

$$\begin{aligned}
\mathcal{F}(D^*, G) &= \int_{\mathbb{R}} \log\left(\frac{f_X(s)}{f_X(s)+f_G(x)}\right) f_X(s)\, ds + \int_{\mathbb{R}} \log\left(\frac{f_G(s)}{f_X(s)+f_G(x)}\right) f_G(s)\, ds \\
&= \int_{\mathbb{R}} \log\left(\frac{1}{2}\frac{f_X(s)}{\frac{f_X(s)+f_G(x)}{2}}\right) f_X(s)\, ds + \int_{\mathbb{R}} \log\left(\frac{1}{2}\frac{f_G(s)}{\frac{f_X(s)+f_G(x)}{2}}\right) f_G(s)\, ds \\
&= -\log(4) + \mathrm{KL}\left(X \middle\| \frac{X+G}{2}\right) + \mathrm{KL}\left(G \middle\| \frac{X+G}{2}\right) \\
&= -\log(4) + \underbrace{\mathrm{JSD}(X, G)}_{\text{Jensen-Shannon div.}} .
\end{aligned}$$

Recall: $\mathrm{KL}(X \,||\, Y) = \int_{\mathbb{R}} f_X(s) \log\left(\frac{f_X(s)}{f_Y(s)}\right) ds.$

Probabilistic interpretation of GANs (Goodfellow et al.)

For perfect discriminator D, the generator G aims to minimize the Jensen-Shannon divergence between the original distribution and the synthetic distribution.

Optimizing the GAN game

The universal ML trick

Optimize stuff by gradient descent.

$$\theta_D^{n+1} = \theta_D^n + \nabla_{\theta_D} \mathcal{F}(\theta_D^n, \theta_G^n)$$
$$\theta_G^{n+1} = \theta_G^n - \nabla_{\theta_G} \mathcal{F}(\theta_D^n, \theta_G^n)$$

In practice

- Sample minibatches to estimate the expectations

$$\nabla \mathcal{F}(\theta_D^n, \theta_G^n) \approx \frac{1}{N} \sum_{i=1}^{N} \nabla \log \left(D_{\theta_D^n}(x_i) \right) + \frac{1}{M} \sum_{j=1}^{M} \nabla \log \left(1 - D_{\theta_D^n}(G_{\theta_G^n}(\lambda_j)) \right).$$

- Apply backpropagation to optimize the weights of the neural network

$$\nabla_{\theta_D} \mathcal{F}(\theta_D^n, \theta_G^n) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{\nabla_{\theta_D} D(x_i)}{D_{\theta_D^n}(x_i)} - \frac{1}{M} \sum_{j=1}^{M} \frac{\nabla_{\theta_D} D(G_{\theta_G^n}(\lambda_j))}{1 - D_{\theta_D^n}(G_{\theta_G^n}(\lambda_j))}.$$

The Eden: convergence of the GAN flow

Theorem (Nagarajan-Kolter)

Suppose that the GAN satisfy the following assumptions:

- The perfect discriminator (D^*) and generator ($f_G = f_X$) are representable.
- Around any Nash equilibrium, the functions
$$\theta_D \mapsto \mathbb{E}_\Omega D^2_{\theta_D}(X) \text{ and } \theta_G \mapsto ||\mathbb{E}_\Omega \nabla_{\theta_D} D^*(X) - \mathbb{E}_\Lambda \nabla_{\theta_D} D^*(G_{\theta_G})||^2$$
are locally constant in any isotropic direction of the Hessian.
- $\operatorname{supp} f_{G_{\theta_G}} = \operatorname{supp} f_X$ nearby the equilibrium points.

$\Rightarrow$ Any Nash equilibrium of the GAN game is **locally stable** for the training through gradient descent.

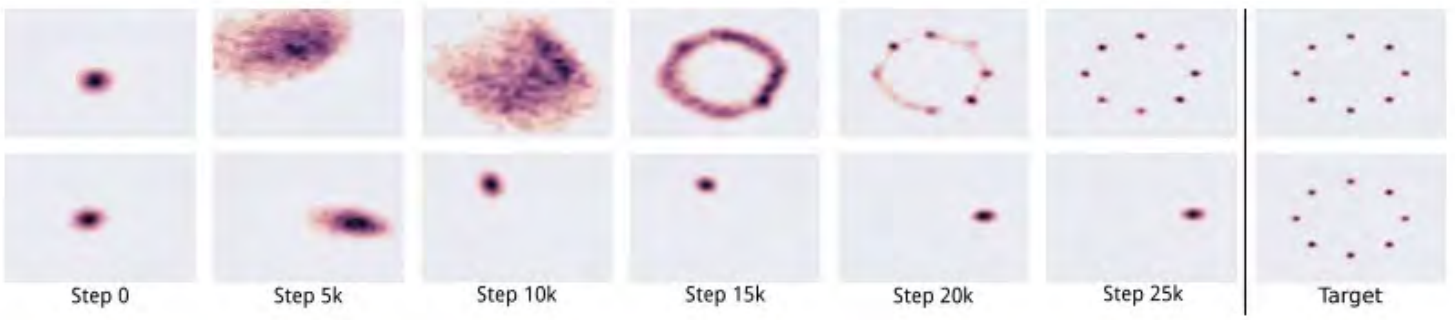
Proof: Linearize the system around the critical points (Nash equilibria) and study the Jacobian matrix.

Problems in the paradise: not so good convergence

In practice: Very bad convergence is observed in real trainings.

- Vanishing gradient: If D is too good, then G cannot be trained ($\nabla_{\theta_G} \mathcal{F} \approx 0$).
- Mode collapsing: If X is 'multimodal' then G only produces the 'most likely' mode

$$G^*(\lambda) = \text{argmax}_{x \in \mathbb{R}^d} D(x)$$



- Dirac GAN: Non-convergent GAN.

The guilty: The Nash flow

Let $\mathcal{F} : \Theta_D \times \Theta_G \to \mathbb{R}$ be a differentiable function.

Morse (gradient) flow

$$\begin{cases} \theta'_D &= \nabla_{\theta_D} \mathcal{F} \quad (\max), \\ \theta'_G &= \nabla_{\theta_G} \mathcal{F} \quad (\max). \end{cases}$$

Properties: 'Easy' to understand, generically non-degenerate critical points, good converge, tight relation with topology (Morse theory).

Nash flow

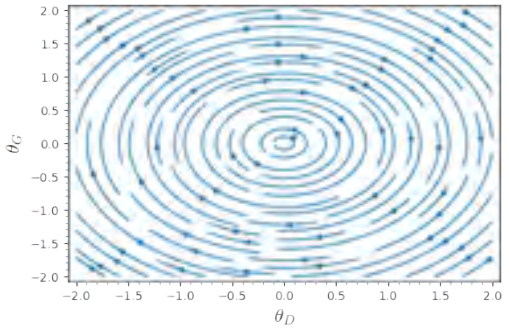
$$\begin{cases} \theta'_D &= \nabla_{\theta_D} \mathcal{F} \quad (\max), \\ \theta'_G &= -\nabla_{\theta_G} \mathcal{F} \quad (\min). \end{cases}$$

Properties: Hard to understand, admits centers and degenerate critical points, poor converge...

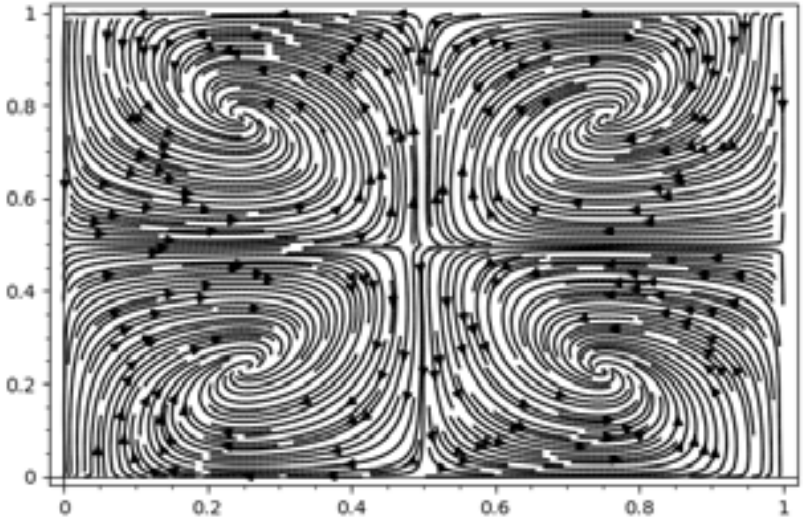
Dynamic of the Nash flow is a nightmare.

Example 1: $\mathcal{F}(\theta_D, \theta_G) = \theta_D \cdot \theta_G \Rightarrow (0,0)$ is a saddle point.

$$\begin{cases} \theta'_D &= \theta_G, \\ \theta'_G &= -\theta_D. \end{cases}$$



Example 2: Real (toy) GAN



Bend the curve

Empirical observation: The generator network tends to follow a 'normal-like' distribution

Key idea

To 'bend' this distribution to get a more similar shape to the real distribution.

Let $Z : \Omega \to \mathbb{R}$ be a random variable with continuous increasing cumulative probability function $F_Z : \mathbb{R} \to [0, 1]$.

Proposition (Probability Integral Transform)

The random variable

$$F_Z(Z) : \Omega \to \mathbb{R}$$

is uniformly distributed with support $[0, 1]$.

The holy Smirnov transform

Theorem

If Z is a random variable with continuous increasing cumulative distribution function and $F : \mathbb{R} \to [0, 1]$ is **any** cumulative probability function, then

$$\mathcal{S}(X) := F^{-1} \circ F_Z(Z) : \Omega \to \mathbb{R}$$

has distribution F.

Remark: The inverse F^{-1} must be interpreted in a quantilic sense

$$F^{-1}(p) = \inf\{z \in \mathbb{R} \,|\, F_X(z) \geq p\}.$$

Subtle point: For a random vector X, only marginal information is captured. The **joint** distribution must be inferred by the generator network.

Sampling the Smirnov transform

Definition

Take Z as a standard normal random variable, and let F be the **empirical** cumulative distribution function of X with samples $x_1, \ldots, x_N$

$$F_n^{\text{emp}}(s) = \frac{1}{N} \sum_{i=1}^{N} \chi_{[x_i,\infty)}(s).$$

Then, the Smirnov transform is the map

$$\mathcal{S} = \left(F_n^{\text{emp}}\right)^{-1} \circ F_{\mathcal{N}(0,1)} : \mathbb{R} \to \mathbb{R}.$$

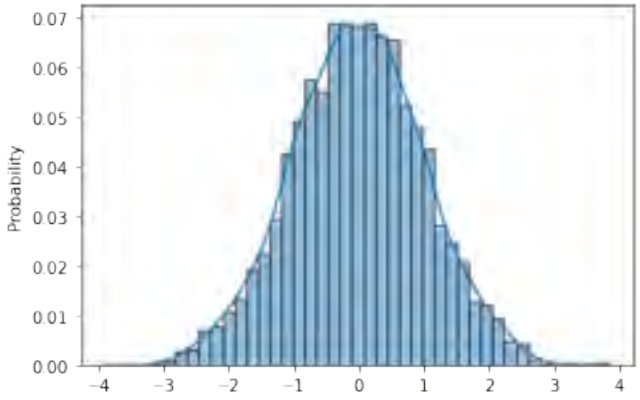
Theorem (Glivenko-Cantelli)

Almost sure we have

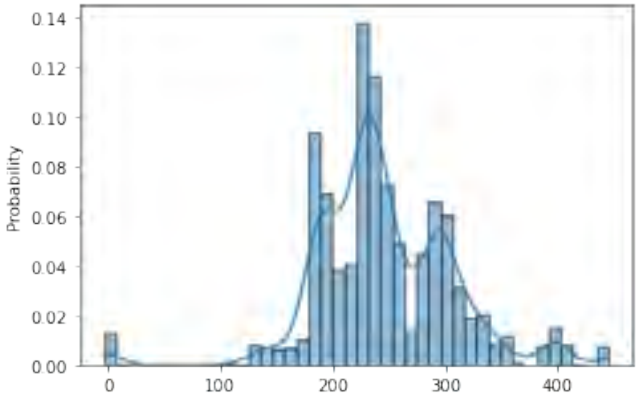
$$||F - F_n^{\text{emp}}||_{L^\infty} \overset{n\to\infty}{\longrightarrow} 0.$$

Glivenko-Cantelli and the consistence of the Smirnov transform

Interpretation: Asymptotically, the Smirnov transform converts a normal distribution into the target distribution X.



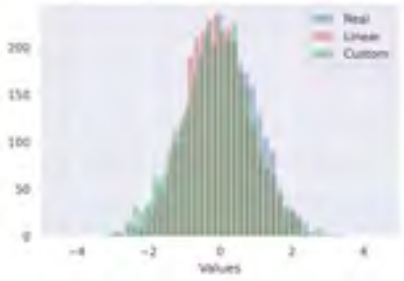
Smirnov trans. $\Longrightarrow$



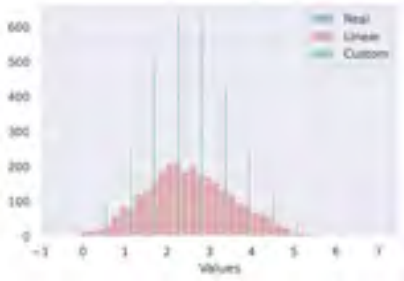
(a) Original distribution

(b) Transformed distribution

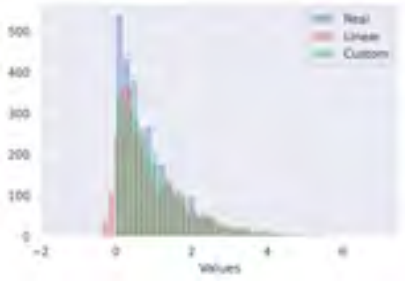
Good news: It works!



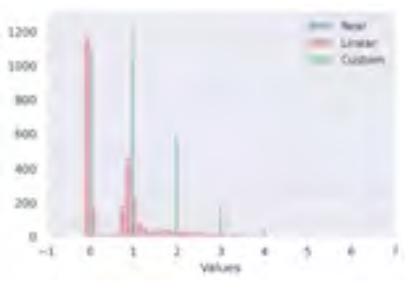
(a) Feature 0.



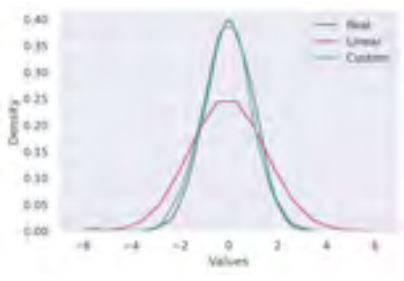
(b) Feature 1.



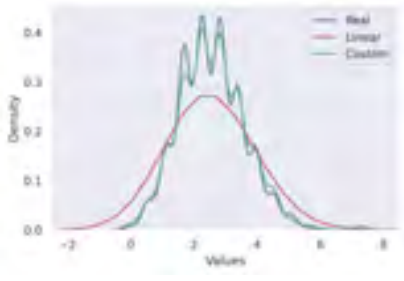
(c) Feature 2.



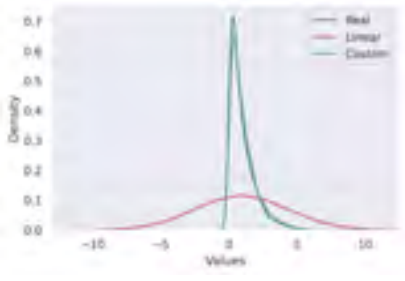
(d) Feature 3.



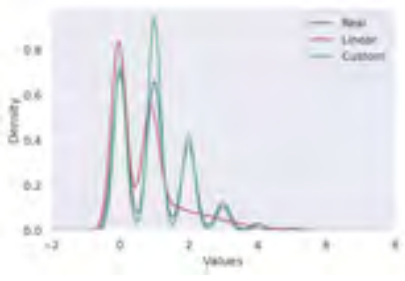
(e) Feature 0.



(f) Feature 1.



(g) Feature 2.



(h) Feature 3.

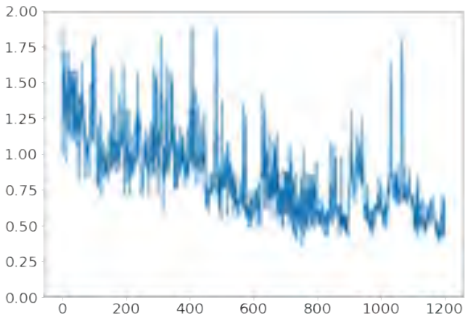
Other quality measures

1. L^p-distance between the empirical cumulative distribution functions of the real data X and the generated data G

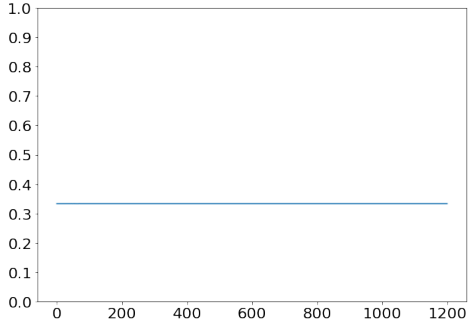
$$d_{L^p}(X, G) = \left(\int_{\mathbb{R}^d} |F^{\text{emp}}_{X,n} - F^{\text{emp}}_{G,n}|^p \, dx \right)^{1/p}$$

2. F_1-score of a nested ML model: Train a ML model to classify the the synthetic data into classes, and compare with the obtained performance when training with real data.

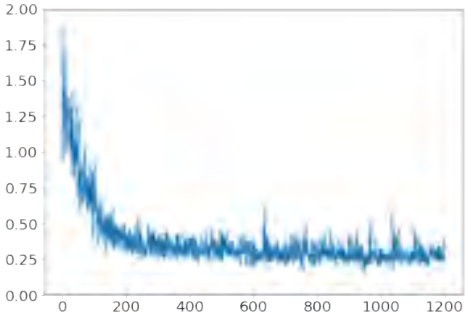
More good news!



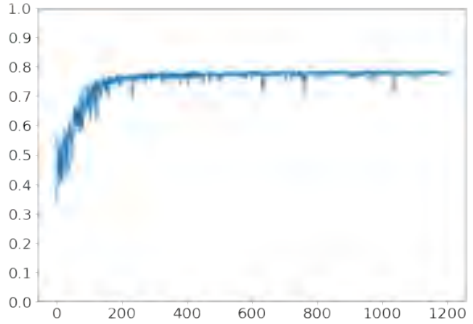
(a) L^1-distance (Classic GAN)



(b) F_1-score (Classic GAN)



(c) L^1-distance (Smirnov GAN)



(d) F_1-score (Smirnov GAN)

Future fun

- Analyze GAN training using dynamical systems.
- Beat the Nash flow: Perturbations and regularizations.
- Find a theoretical explanation for the empirical normal-like behavior ($\approx$ mode collapse).
- Smoothing the activation function.
- Analyze the effect of using copulas.

Thank you very much for your attention!