Lecture notes on generative adversarial networks (GANs)

1.1 Prerequisites

Binary classification

Given a data distribution $p_{\text{data}}(\boldsymbol{x}, y)$ with $y \in \{0, 1\}$, we would like to fit a binary classifier $p_{\boldsymbol{\phi}}(y|\boldsymbol{x})$ to the conditional distribution $p_{\text{data}}(y|\boldsymbol{x})$. A maximum likelihood estimate of the parameters $\boldsymbol{\phi}$ is obtained by solving the following optimisation task:

$$\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \mathbb{E}_{p_{\text{data}}(\boldsymbol{x},y)}[\log p_{\boldsymbol{\phi}}(y|\boldsymbol{x})], \tag{1}$$

Assume the dataset is balanced, i.e. $p_{\text{data}}(y) = \text{Bern}(0.5)$, then the above objective is equivalent to

$$\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|y=1)}[\log p_{\boldsymbol{\phi}}(y=1|\boldsymbol{x})] + \mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|y=0)}[\log(1 - p_{\boldsymbol{\phi}}(y=1|\boldsymbol{x}))]. \tag{2}$$

The negation of the above maximum likelihood objective is also known as the cross-entropy loss.

1.2 Generative adversarial networks (GANs)

Original GAN formulation as binary classification

The generative adversarial network (GAN) approach [Goodfellow et al., 2014] constructs a binary classification task to assist the learning of the generative model distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ to fit the data distribution $p_{\text{data}}(\boldsymbol{x})$. This is done by labelling all the datapoints sampled from the data distribution as "real" data and those sampled from the model as "fake" data. In other words, a joint distribution $\tilde{p}(\boldsymbol{x}, y)$ is constructed as follows for the binary classification task:

$$\tilde{p}(\boldsymbol{x}, y) = \tilde{p}(\boldsymbol{x}|y)\tilde{p}(y), \quad \tilde{p}(y) = \text{Bern}(0.5), \quad \tilde{p}(\boldsymbol{x}|y) = \begin{cases} p_{\text{data}}(\boldsymbol{x}), & y = 1 \\ p_{\boldsymbol{\theta}}(\boldsymbol{x}), & y = 0 \end{cases}. \tag{3}$$

Fitting a binary classifier ("discriminator") with $p_{\boldsymbol{\phi}}(y = 1|\boldsymbol{x}) = D_{\boldsymbol{\phi}}(\boldsymbol{x})$ to $\tilde{p}(y|\boldsymbol{x})$ can be done by maximising the maximum likelihood objective (see eq. (2)):

$$\boldsymbol{\phi}^*(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}), \quad \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) := \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}[\log D_{\boldsymbol{\phi}}(\boldsymbol{x})] + \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}[\log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}))]. \tag{4}$$

Notice the dependence of the objective (4) on the generative model parameter $\boldsymbol{\theta}$, since the "data distribution" $\tilde{p}(\boldsymbol{x}, y)$ of the binary classification task depends on $p_{\boldsymbol{\theta}}(\boldsymbol{x})$. Then the training of the generative model $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ aims at fooling the discriminator, by *minimising* the log probability of making the right decisions:

$$\boldsymbol{\theta}^*(\boldsymbol{\phi}) = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}[\log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}))]. \tag{5}$$

In summary, the two-player game objective for training the GAN generator and discriminator is

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}). \tag{6}$$

Importantly, the terms in the objective related to $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is $\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}[\log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}))]$ which in practice is approximated by Monte Carlo:

$$\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}[\log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}))] \approx \log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x})), \quad \boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}). \tag{7}$$

Therefore the evaluation of the objective does not require computation of the distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, instead one can directly define the sampling process of $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, which also defines the distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ in an *implicit* way:

$$\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}) \quad \Leftrightarrow \quad \boldsymbol{z} \sim p(\boldsymbol{z}), \quad \boldsymbol{x} = G_{\boldsymbol{\theta}}(\boldsymbol{z}), \tag{8}$$

and often we set $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \mathrm{I})$.

Equivalence to Jensen-Shannon divergence minimisation

In order to justify the two-player game objective (6), in the following we will show that with infinite capacity for both the generator and the discriminator, the global optimum of the generator is $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})$. For a fixed generator $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, we compute the gradient of the GAN objective w.r.t. $\boldsymbol{\phi}$:

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \int \left(\frac{p_{\text{data}}(\boldsymbol{x})}{D_{\boldsymbol{\phi}}(\boldsymbol{x})} - \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{1 - D_{\boldsymbol{\phi}}(\boldsymbol{x})} \right) \nabla_{\boldsymbol{\phi}} D_{\boldsymbol{\phi}}(\boldsymbol{x}) d\boldsymbol{x} \tag{9}$$

Given infinite capacity of the discriminator, setting $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = 0$ results in

$$\frac{p_{\text{data}}(\boldsymbol{x})}{D_{\boldsymbol{\phi}}(\boldsymbol{x})} = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{1 - D_{\boldsymbol{\phi}}(\boldsymbol{x})} \quad \Rightarrow \quad D_{\boldsymbol{\phi}^*(\boldsymbol{\theta})}(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}) + p_{\text{data}}(\boldsymbol{x})}. \tag{10}$$

Pluggin in the optimal discriminator to the GAN objective:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}^*(\boldsymbol{\theta})) &= \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})} \left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}) + p_{\text{data}}(\boldsymbol{x})} \right] + \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}) + p_{\text{data}}(\boldsymbol{x})} \right] \\ &= \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})} \left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{\frac{1}{2}(p_{\boldsymbol{\theta}}(\boldsymbol{x}) + p_{\text{data}}(\boldsymbol{x}))} \right] + \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{\frac{1}{2}(p_{\boldsymbol{\theta}}(\boldsymbol{x}) + p_{\text{data}}(\boldsymbol{x}))} \right] - 2 \log 2 \\ &= 2 \underbrace{(\frac{1}{2} \text{KL} \left[p_{\text{data}}(\boldsymbol{x}) || \frac{1}{2}(p_{\text{data}}(\boldsymbol{x}) + p_{\boldsymbol{\theta}}(\boldsymbol{x})) \right] + \frac{1}{2} \text{KL} \left[p_{\boldsymbol{\theta}}(\boldsymbol{x}) || \frac{1}{2}(p_{\text{data}}(\boldsymbol{x}) + p_{\boldsymbol{\theta}}(\boldsymbol{x})) \right])}_{:= \text{JS}[p_{\text{data}}(\boldsymbol{x}) || p_{\boldsymbol{\theta}}(\boldsymbol{x})]} - 2 \log 2, \end{aligned} \tag{11}$$

where $\text{JS}[p_{\text{data}}(\boldsymbol{x}) || p_{\boldsymbol{\theta}}(\boldsymbol{x})]$ is the Jensen-Shannon divergence between $p_{\text{data}}(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x})$. Since Jensen-Shannon divergence is a valid divergence measure, this means with infinite capacity for the generator, $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}^*(\boldsymbol{\theta}))$ is minimised iff. $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})$.

Alternative loss for the generator

In the original GAN paper [Goodfellow et al., 2014] the authors proposed to optimise an alternative "non-saturated" objective for the generator, given a fixed discriminator:

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})} [\log D_{\boldsymbol{\phi}}(\boldsymbol{x})]. \tag{12}$$

Compared with the original objective (5) which minimises the log probability of making correct predictions, the alternative objective maximises the log probability of making *wrong* predictions. To see how this approach helps, notice that the discriminator often has near-perfect classification performance at the beginning of GAN training (since at this stage the "fake" data quality is bad). In this case $D_{\boldsymbol{\phi}}(\boldsymbol{x}) \approx 0$ for $\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x})$. Also assume the generative model is implicitly defined by $\boldsymbol{z} \sim p(\boldsymbol{z}), \boldsymbol{x} = G_{\boldsymbol{\theta}}(\boldsymbol{z})$. Note that $D_{\boldsymbol{\phi}}(\boldsymbol{x})$ is often defined using sigmoid activation $\text{sigmoid}(t) = (1 + exp[-t])^{-1}$ at the last layer, i.e. $D_{\boldsymbol{\phi}}(\boldsymbol{x}) = \text{sigmoid}(d_{\boldsymbol{\phi}}(\boldsymbol{x}))$ with $d_{\boldsymbol{\phi}}(\boldsymbol{x})$ parameterised by a neural network. This means $D_{\boldsymbol{\phi}}(\boldsymbol{x}) \approx 0$ when $d_{\boldsymbol{\phi}}(\boldsymbol{x}) \to -\infty$ (so at the beginning of GAN training $d_{\boldsymbol{\phi}}(\boldsymbol{x}) \to -\infty$ for $\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x})$). Therefore the gradients of the two objectives w.r.t. $\boldsymbol{\theta}$ are

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})} [\log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}))] = -\nabla_{\boldsymbol{\theta}} \mathbb{E}_{p(\boldsymbol{z})} [\log(1 + \exp[d_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\boldsymbol{z}))])] = -\mathbb{E}_{p(\boldsymbol{z})} [\underbrace{D_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\boldsymbol{z}))}_{\approx 0} \nabla_{\boldsymbol{\theta}} d_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\boldsymbol{z}))], \tag{13}$$

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})} [\log D_{\boldsymbol{\phi}}(\boldsymbol{x})] = -\nabla_{\boldsymbol{\theta}} \mathbb{E}_{p(\boldsymbol{z})} [\log(1 + \exp[-d_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\boldsymbol{z}))])] = \mathbb{E}_{p(\boldsymbol{z})} [\underbrace{(1 - D_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\boldsymbol{z})))}_{\approx 1} \nabla_{\boldsymbol{\theta}} d_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\boldsymbol{z}))]. \tag{14}$$

It is clear that the alternative objective addresses the vanishing gradient problem of the original one (5) at the beginning of training, hence the name "non-saturated objective".

Another justification of the alternative objective is provided by deriving the optimal solution of the generator, given the optimal discriminator. Define $f(t) = \log(1 + t^{-1}) - \log 2$, in which $f(t)$ is convex and $f(1) = 0$. Then we can define an f-divergence [Csiszár, 1963; Morimoto, 1963; Ali and Silvey, 1966] as

$$\begin{aligned} \mathrm{D}_f[p_{\boldsymbol{\theta}}(\boldsymbol{x}) || p_{\text{data}}(\boldsymbol{x})] &:= \int p_{\boldsymbol{\theta}}(\boldsymbol{x}) f\left(\frac{p_{\text{data}}(\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})}\right) d\boldsymbol{x} \\ &= \int p_{\boldsymbol{\theta}}(\boldsymbol{x}) \log\left(1 + \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})}\right) d\boldsymbol{x} - \log 2 \\ &= -\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}[\log D_{\boldsymbol{\phi}^*(\boldsymbol{\theta})}(\boldsymbol{x})] - \log 2. \end{aligned} \tag{15}$$

This shows that maximising the alternative "non-saturated objective" is equivalent to minimising an f-divergence between the model and the data distribution. Therefore again with infinite capacity of the generator, the optimal solution of the generative model is $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})$.

1.3 Conditional GAN

Similar to conditional VAEs, conditional GAN is a particular parameterisation of $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{y})$ which approximates the data distribution $p_{\text{data}}(\boldsymbol{x}|\mathbf{y})$. Again $\boldsymbol{x}$ is the random variable for data and $\mathbf{y}$ is the random variable corresponding to the additional information. For the design of the generative model $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{y})$, we use a conditional LVM as follows:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{y}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \mathbf{y}) p(\boldsymbol{z}) d\boldsymbol{z}, \tag{16}$$

and often we set $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathrm{I})$. But different from conditional VAE which explicitly specifies the distribution form of $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \mathbf{y})$, for conditional GAN it is defined implicitly by the following sampling process:

$$\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \mathbf{y}) \quad \Leftrightarrow \quad \boldsymbol{z} \sim p(\boldsymbol{z}), \boldsymbol{x} = G_{\boldsymbol{\theta}}(\boldsymbol{z}, \mathbf{y}), \tag{17}$$

with $G_{\boldsymbol{\theta}}(\boldsymbol{z}, \mathbf{y})$ defined by a neural network that takes both $\boldsymbol{z}$ and $\mathbf{y}$ as inputs. Similar to GANs, learning is done by optimising an adversarial objective:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \; \mathbb{E}_{p_{\text{data}}(\boldsymbol{x}, \mathbf{y})}[\log D_{\boldsymbol{\phi}}(\boldsymbol{x}, \mathbf{y})] + \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{y}) p_{\text{data}}(\mathbf{y})}[\log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}, \mathbf{y}))]. \tag{18}$$

In practice the component related to the generator parameters $\boldsymbol{\theta}$ is computed by

$$\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{y}) p_{\text{data}}(\mathbf{y})}[\log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}, \mathbf{y}))] \approx \log(1 - D_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\boldsymbol{z}, \mathbf{y}), \mathbf{y})), \quad \boldsymbol{z} \sim p(\boldsymbol{z}), \mathbf{y} \sim p_{\text{data}}(\mathbf{y}). \tag{19}$$

Using similar techniques, one can derive the optimal discriminator for a fixed generative model with parameter $\boldsymbol{\theta}$:

$$D_{\boldsymbol{\phi}^*(\boldsymbol{\theta})}(\boldsymbol{x}, \mathbf{y}) = \frac{p_{\text{data}}(\boldsymbol{x}, \mathbf{y})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{y}) p_{\text{data}}(\mathbf{y}) + p_{\text{data}}(\boldsymbol{x}, \mathbf{y})}, \tag{20}$$

and with the optimal discriminator, maximising $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ w.r.t. $\boldsymbol{\theta}$ is equivalent to minimising the Jensen-Shannon divergence $\mathrm{JS}[p_{\text{data}}(\boldsymbol{x}, \mathbf{y}) || p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{y}) p_{\text{data}}(\mathbf{y})]$.

1.4 *Wasserstein GAN

Wasserstein distance

Wasserstein distance is a key concept developed in optimal transport, which aims at finding the lowest cost approach to transform a distribution to another [Villani, 2008]. The dual form of the Wasserstein distance is defined by taking the optimal *test functions* from $\mathcal{F} = \{f : ||f||_L \leq 1\}$, the set of 1-Lipschitz functions:

$$W_2[p, q] = \sup_{||f||_L \leq 1} \mathbb{E}_p[f(\boldsymbol{x})] - \mathbb{E}_q[f(\boldsymbol{x})]. \tag{21}$$

As a reminder, a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is said to be l-Lipschitz (denoted as $||f||_L \leq l$) if

$$|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)| \leq l||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2, \quad \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d. \tag{22}$$

If f is differentiable everywhere, then

$$||f||_L \leq 1 \quad \Leftrightarrow \quad ||\nabla_{\boldsymbol{x}} f(\boldsymbol{x})||_2 \leq 1, \forall \boldsymbol{x} \in \mathbb{R}^d. \tag{23}$$

Using Wasserstein distance in GANs

In Wasserstein GANs [Arjovsky et al., 2017], the discriminator is used to parameterise the test function $f := D_{\boldsymbol{\phi}}$, and the Wasserstein distance is used as the loss function for adversarial training:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}[D_{\boldsymbol{\phi}}(\boldsymbol{x})] - \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}[D_{\boldsymbol{\phi}}(\boldsymbol{x})], \quad \text{subject to } ||\nabla_{\boldsymbol{x}} D_{\boldsymbol{\phi}}(\boldsymbol{x})||_2 \leq 1, \forall \boldsymbol{x} \in \mathbb{R}^d. \tag{24}$$

However, it is impractical to compute the constraint for every $\boldsymbol{x} \in \mathbb{R}^d$. Instead, the point-wise constraint is replaced by the following alternative [Gulrajani et al., 2017]:

$$\mathbb{E}_{\hat{p}(\boldsymbol{x})}[(||\nabla_{\boldsymbol{x}} D_{\boldsymbol{\phi}}(\boldsymbol{x})||_2 - 1)^2] = 0, \tag{25}$$

with the auxiliary "interpolation" distribution $\hat{p}(\boldsymbol{x})$ defined by the following generative process:

$$\boldsymbol{x} \sim \hat{p}(\boldsymbol{x}) \quad \Leftrightarrow \quad \boldsymbol{x}_d \sim p_{\text{data}}(\boldsymbol{x}), \boldsymbol{x}_g \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}), \alpha \sim \text{Uniform}([0, 1]), \boldsymbol{x} = \alpha \boldsymbol{x}_d + (1 - \alpha)\boldsymbol{x}_g. \tag{26}$$

This alternative constraint (25) is justified as follows. Since the original Wasserstein distance objective (24) requires evaluating the discriminator within the supports of $p_{\text{data}}(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ only, it requires to enforce the $||\nabla_{\boldsymbol{x}} D_{\boldsymbol{\phi}}(\boldsymbol{x})||_2 \leq 1$ constraint for $\boldsymbol{x} \in \text{supp}(p_{\text{data}}(\boldsymbol{x})) \cup \text{supp}(p_{\boldsymbol{\theta}}(\boldsymbol{x}))$. Also it can be shown that the optimal discriminator of the objective (24) satisfies $||\nabla_{\boldsymbol{x}} D_{\boldsymbol{\phi}}(\boldsymbol{x})||_2 = 1$ for $\boldsymbol{x} \in \text{supp}(p_{\text{data}}(\boldsymbol{x})) \cup \text{supp}(p_{\boldsymbol{\theta}}(\boldsymbol{x}))$. Now the alternative constraint (25) is satisfied iff. $||\nabla_{\boldsymbol{x}} D_{\boldsymbol{\phi}}(\boldsymbol{x})||_2 = 1$ for $\boldsymbol{x} \in \text{supp}(\hat{p}(\boldsymbol{x}))$. Given that $\text{supp}(p_{\text{data}}(\boldsymbol{x})) \cup \text{supp}(p_{\boldsymbol{\theta}}(\boldsymbol{x})) \subset \text{supp}(\hat{p}(\boldsymbol{x}))$ by construction, this indicates that the constraint in the Wasserstein distance object (24) is satisfied if the constraint (25) is satisfied. The optimisation of the objective (24) with alternative constraint (25) can be solved by the Lagrange multiplier method, resulting in the WGAN-GP ("Wasserstein GAN with gradient penalty") objective [Gulrajani et al., 2017]:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}[D_{\boldsymbol{\phi}}(\boldsymbol{x})] - \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}[D_{\boldsymbol{\phi}}(\boldsymbol{x})] + \lambda \mathbb{E}_{\hat{p}(\boldsymbol{x})}[(||\nabla_{\boldsymbol{x}} D_{\boldsymbol{\phi}}(\boldsymbol{x})||_2 - 1)^2]. \tag{27}$$

Integral probability metrics (IPMs)

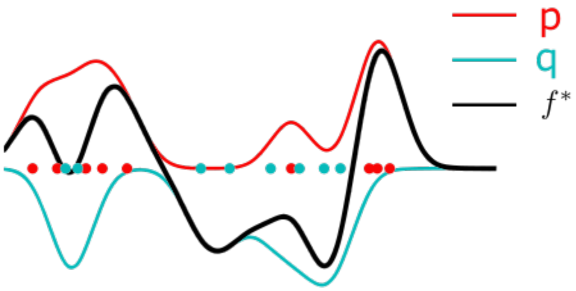
Wasserstein distance is an instance of a family of distance measures between distributions, named integral probability metrics (IPMs).

Definition 1. *(Integral probability metric (IPM)) Given a set of test functions $\mathcal{F}$, consider the following quantity:*

$$\mathrm{D}[p, q] = \sup_{f \in \mathcal{F}} |\mathbb{E}_p[f(\boldsymbol{x})] - \mathbb{E}_q[f(\boldsymbol{x})]|, \tag{28}$$

where $|\cdot|$ denotes a norm in the output space of f. If $\mathcal{F}$ is sufficiently large such that $\mathrm{D}[p, q] = 0$ iff. $p = q$, then $\mathrm{D}[p, q]$ is said to be an integral probability metric defined by the test functions in $\mathcal{F}$.

To provide an intuition of IPMs, consider a strategy of comparing distributions by comparing their *moments*, e.g. mean, variance, kurtosis, etc. Loosely speaking, if two distributions p and q have the same moments for all orders then p and q should be identical.[1] Therefore, to check whether p and q are identical or not, one can find the best moment, or in a broader sense the best test function



f that can distinguish p from q the most, and if such optimal test function still fails to distinguish between p and q, then the two distributions p and q are identical.[2]

The intuition is further visualised in the above figure.[3] We see from the visualisation that the optimal test function f^* takes positive values in the region where $p(\boldsymbol{x}) > q(\boldsymbol{x})$ and vise versa. In other words, the optimal test function tells us more than whether $p = q$ or not; it also provides information on *how* p and q differ from each other. This is a useful property for IPMs for applications in adversarial learning: as f^* describes in detail the difference between p and q, we can optimise the q distribution in a guided way towards approximating the target distribution p. Indeed various versions of IPMs have been used as optimisation objectives in the GAN literature, e.g. see Li et al. [2017]; Mroueh and Sercu [2017]; Mroueh et al. [2018].

References

Ali, S. M. and Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 131–142.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 214–223.

Csiszár, I. (1963). Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der ergodizitat von markoffschen ketten. *Magyar. Tud. Akad. Mat. Kutató Int. Közl*, 8:85–108.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5769–5779.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., and Póczos, B. (2017). MMD GAN: Towards deeper understanding of moment matching network. In *Advances in neural information processing systems*, pages 2203–2213.

Morimoto, T. (1963). Markov processes and the H-theorem. *Journal of the Physical Society of Japan*, 18(3):328–331.

Mroueh, Y., Li, C.-L., Sercu, T., Raj, A., and Cheng, Y. (2018). Sobolev GAN. In *International Conference on Learning Representations*.

[1] This is subject to some conditions, e.g. the moment generating functions exist for p & q.

[2] Again under some assumptions of the form for p & q.

[3] Figure adapted from Dougal Sutherland's slides: http://www.gatsby.ucl.ac.uk/~dougals/slides/dali/#/38

Mroueh, Y. and Sercu, T. (2017). Fisher GAN. In *Advances in Neural Information Processing Systems*, pages 2513–2523.

Villani, C. (2008). *Optimal transport: old and new*. Springer Science & Business Media.