Spread Divergence

Mingtian Zhang [1] **Peter Hayes** [1] **Tom Bird** [1] **Raza Habib** [1] **David Barber** [1]

Abstract

For distributions $\mathbb{P}$ and $\mathbb{Q}$ with different supports or undefined densities, the divergence $\mathrm{D}(\mathbb{P}||\mathbb{Q})$ may not exist. We define a Spread Divergence $\tilde{\mathrm{D}}(\mathbb{P}||\mathbb{Q})$ on modified $\mathbb{P}$ and $\mathbb{Q}$ and describe sufficient conditions for the existence of such a divergence. We demonstrate how to maximize the discriminatory power of a given divergence by parameterizing and learning the spread. We also give examples of using a Spread Divergence to train implicit generative models, including linear models (Independent Components Analysis) and non-linear models (Deep Generative Networks).

1. Introduction

For distributions $\mathbb{P}$ and $\mathbb{Q}$, a divergence $\mathrm{D}(\mathbb{P}||\mathbb{Q})$ is a measure of their difference (Dragomir, 2005) provided it satisfies the properties[1]

$$\mathrm{D}(\mathbb{P}||\mathbb{Q}) \geq 0 \;\text{ and }\; \mathrm{D}(\mathbb{P}||\mathbb{Q}) = 0 \;\Leftrightarrow\; \mathbb{P} = \mathbb{Q}. \tag{1}$$

For absolutely continuous distributions[2] $\mathbb{P}$, $\mathbb{Q}$ and probability density functions $p(x)$, $q(x)$, the f-divergence is

$$\mathrm{D}_f(\mathbb{P}||\mathbb{Q}) \equiv \mathrm{D}_f(p||q) = \mathbb{E}_{q(x)}\left[f\left(\frac{p(x)}{q(x)}\right)\right], \tag{2}$$

where $f(x)$ is a convex function with $f(1) = 0$. When p and q have the same support[3],

$$\mathrm{D}_f(p||q) = 0 \Leftrightarrow p = q \; a.e. \Leftrightarrow \mathbb{P} = \mathbb{Q}, \tag{3}$$

see (Csiszár, 1967; Ali & Silvey, 1966). A well-known f-divergence is the Kullback-Leibler (KL) divergence

$$\mathrm{KL}(p||q) = \mathbb{E}_{p(x)}\left[\log \frac{p(x)}{q(x)}\right]. \tag{4}$$

The KL divergence plays a key role in fitting a model $q(x)$ to data. For training data $x_1, \ldots, x_N$ generated iid from a distribution $p(x)$, a sample based approximation of eq(4) is

$$\mathrm{KL}(p||q) = -\frac{1}{N}\sum_{n=1}^{N} \log q(x_n) + const. \tag{5}$$

where $\mathcal{L}(\theta) \equiv \sum_{n=1}^{N} \log q(x_n)$ is the data log likelihood and θ are the parameters of the model $q(x)$. Hence, fitting a model q to data using maximum likelihood can be viewed as minimizing the KL divergence between the model and the data generating process.

Using an f-divergence $\mathrm{D}_f(\mathbb{P}||\mathbb{Q})$ for model training therefore requires (i) $\mathbb{P}$ and $\mathbb{Q}$ to have valid probability densities p, q and (ii) p and q to have common support. However, these requirements are not satisfied in some important machine learning applications, in particular in implicit generative models, as we discuss below.

1.1. Implicit Generative Models

Implicit generative models are of considerable recent interest in machine learning. These take the form of a latent variable model $q(x) = \int q(x|z)q(z)dz$, but with a deterministic output distribution $q(x|z) = \delta(x - g_\theta(z))$. We can define the generalised density[4], for the model $\mathbb{Q}$ as

$$q(x) = \int \delta\left(x - g_\theta(z)\right) q(z) dz, \tag{6}$$

where z represents the value of latent variable Z, x the value of observation variable X and $\delta(x)$ is the Dirac delta function. In the common setting where the latent dimension is lower than the observation dimension $\dim(Z) < \dim(X)$,

[1]Department of Computer Science, University College London, UK. Correspondence to: Mingtian Zhang <mingtian.zhang.17@ucl.ac.uk>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

[1]Two distributions being equal $\mathbb{P} = \mathbb{Q}$ can be interpreted to mean that the cdfs (cumulative distribution functions) of the two distributions match.

[2]A distribution is absolutely continuous if its cdf is absolutely continuous. In this case, it has a density function. See also Tao (2011, p. 172).

[3]We use $a.e.$ to represent 'almost everywhere', see Durrett (2019, p. 18) for a definition.

[4]We use 'generalised density' to include implicit generative models. This includes also the special case $p(x) = \delta(x - \mu)$, where $p(x)$ represents a (Dirac) delta distribution with support $\{\mu\}$. In general we cannot define an f-divergence based on generalised densities, but can do so if the distribution is absolutely continuous - see section(2.1.2) for details.

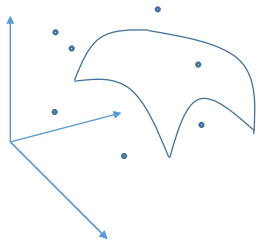


Figure 1. The figure shows an implicit generative model with data $\dim(X) = 3$ and latent $\dim(Z) = 2$. The model can only generate data (dots) on a lower dimensional manifold in X space. The likelihood of generating a data point off the manifold is zero meaning that the likelihood will in general be a non-continuous function of the parameters defining the manifold.

the model $q(x)$ only has limited support, see figure(1). Strictly speaking, this does not define a density over the real space $\mathbb{R}^{\dim(X)}$ since X is not absolutely continuous. Nevertheless, this does define a distribution $\mathbb{Q}$. To generate a sample from $\mathbb{Q}$, one first samples z from $q(z)$ and then passes this through the 'generator' function $g_\theta(z)$.

In this case, gradient based maximum likelihood learning is problematic since $\mathcal{L}(\theta)$ is typically not a continuous function of θ[5]. Furthermore, the Expectation Maximisation algorithm (Dempster et al., 1977) is not available for models of the form in eq(6) since EM assumes that $\log q(x|z)$ is well defined, which is not the case when $q(x|z) = \delta(x - g_\theta(z))$ (Bermond & Cardoso, 1999). Equally, in eq(4) the ratio $p(x)/q(x)$ may represent a division by zero; the KL divergence between the model and the data generating process is thus ill-defined.

It is natural to consider transforming two distributions $\mathbb{P}$ and $\mathbb{Q}$ to have common support by using a mixture model $\tilde{\mathbb{P}} = \alpha\mathbb{P} + (1-\alpha)\mathbb{N}$, $\tilde{\mathbb{Q}} = \alpha\mathbb{Q} + (1-\alpha)\mathbb{N}$, where $\mathbb{N}$ is an absolutely continuous noise distribution with full support. However, as we explain in appendix(E) this approach is not useful in the context of training implicit generative models since $\tilde{\mathbb{Q}}$ does not have a density which can be numerically evaluated.

1.1.1. Model noise is not enough

A common approach to enable maximum likelihood to be used to train implicit generative models is to simply add noise to the model so that it has full support (and a valid density), see for example (Wu et al., 2016). However, this approach does not guarantee a consistent estimator. To see this, consider the simple implicit generative model $\mathbb{Q}$ with generalised density

$$q(x) = \int \delta(x - z\theta_q)\,\mathcal{N}(z|0,1)\,dz, \tag{7}$$

where the latent Z is univariate and $\dim(X) > 1$. Here the vector θ_q defines a one-dimensional line in the X-space. For D-dimensional X, adding independent Gaussian noise with mean zero and isotropic covariance $\sigma^2 I_D$ to X results in the noised distribution with density

$$\tilde{q}(x) = \mathcal{N}(x|0_D, \Sigma), \qquad \Sigma \equiv \theta_q\theta_q^\intercal + \sigma^2 I_D. \tag{8}$$

For observed training data $x_1, \ldots, x_N$ the log likelihood under this model is

$$\frac{1}{N}\mathcal{L}(\theta_q) = -\frac{1}{N}\sum_{n=1}^{N} x_n^\intercal \Sigma^{-1} x_n - \log\det\Sigma + const. \tag{9}$$

We assume that the training data x_n is iid sampled from the distribution $\mathbb{P}$ with generalised density

$$p(x) = \int \delta(x - z\theta_p)\,\mathcal{N}(z|0,1)\,dz. \tag{10}$$

Hence $\mathbb{P}$ and $\mathbb{Q}$ are from the same parametric distribution but with different parameters. By the law of large numbers, in the large N limit, the log likelihood eq(9) tends to

$$-\theta_p^\intercal\Sigma^{-1}\theta_p - \log\det\Sigma + const. \tag{11}$$

which has an optimum when[6] (see appendix(A))

$$\theta_q = \sqrt{\frac{\theta_p^2 - \sigma^2}{\theta_p^2}}\,\theta_p. \tag{12}$$

Thus adding noise to the model $\mathbb{Q}$ and training using maximum likelihood does not form a consistent estimator; it has an optimum at $\theta_q \neq \theta_p$, resulting in an incorrect estimate of the data generating process. In appendix(A) we explain why annealing the noise σ^2 towards zero during numerical optimisation will not directly heal this problem.

Another well-known failure case of trying to learn an implicit generative model by adding noise only to the model is deterministic ICA, which we discuss at length in section(4.1).

For this reason alternative (non-likelihood, non-KL) approaches to measure the difference between distributions are commonly used in training implicit generative models (see for example Mohamed & Lakshminarayanan (2016)), such as Maximum Mean Discrepancy (Gretton et al., 2012) and Wasserstein distance (Arjovsky et al., 2017; Peyré et al., 2019). In the next section we introduce the Spread Divergence which defines a valid divergence even when the

[5] For a point x_n that is not on the model manifold, $q(x_n) = 0$. As we adjust θ such that x_n becomes on the manifold, $q(x_n)$ will typically increase to a finite non-zero value, meaning that the (log) likelihood is discontinuous in θ.

[6] θ_p^2 is shorthand for the squared length $\theta_p^\intercal\theta_p = ||\theta_p||_2^2$.

supports of the distributions do not match or the distributions do not have a valid density. As we will demonstrate, the Spread Divergence allows one to use maximum likelihood based approaches to train implicit generative models, whilst resulting in a consistent estimator.

2. Spread Divergence

For distributions $\mathbb{Q}$ and $\mathbb{P}$ with generalised densities $q(x)$ and $p(x)$ we first need to define $\tilde{q}(y)$ and $\tilde{p}(y)$ that (i) are valid probability density functions and (ii) have the same support. In contrast to simply noising $\mathbb{Q}$ we define 'noisy' densities for both distributions

$$\tilde{p}(y) = \int p(y|x)p(x)dx, \quad \tilde{q}(y) = \int p(y|x)q(x)dx \quad (13)$$

The 'noise' $p(y|x)$ must 'spread' $\mathbb{P}$ and $\mathbb{Q}$ such that $\tilde{p}(y)$ and $\tilde{q}(y)$ satisfy the above two reauirements. The choice of $p(y|x)$ must also ensure that $\mathrm{D}(\tilde{p}||\tilde{q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$. If we can define the noise appropriately, this would allow us to define the Spread Divergence

$$\tilde{\mathrm{D}}(\mathbb{P}||\mathbb{Q}) \equiv \mathrm{D}(\tilde{p}||\tilde{q}), \quad (14)$$

which satisfies the divergence requirement $\tilde{\mathrm{D}}(\mathbb{P}||\mathbb{Q}) \geq 0$ and $\tilde{\mathrm{D}}(\mathbb{P}||\mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$. In the following we discuss appropriate choices for the noise distribution $p(y|x)$. We focus on stationary spread noise $p(y|x) = k(y - x)$ since this is simple to implement by adding independent noise to a variable. Non-stationary spread distributions can be easily constructed using a mixture of stationary noise distributions, or through Mercer kernels – these are left for future study. The case of discrete X is discussed in appendix(B).

2.1. Stationary Spread Divergence

For a random variable X we define a new 'stationary spread' random variable Y by adding to X random noise K. In order to ensure that Y has a valid probability density function and the required support, we assume that K is absolutely continuous with density $k(x)>0$, $x \in \mathbb{R}$. We then define a random variable $Y = X + K$. In the context of eq(13) this corresponds to using noise of the form $p(y|x) = k(y - x)$.

2.1.1. Absolutely Continuous Distributions

For two absolutely continuous distributions $\mathbb{P}$ and $\mathbb{Q}$ with densities $p(x)$ and $q(x)$. We define $\tilde{p}$ and $\tilde{q}$ as a convolution

$$\tilde{p}(y) = \int k(y - x)p(x)dx = (k * p)(y), \quad (15)$$

$$\tilde{q}(y) = \int k(y - x)q(x)dx = (k * q)(y). \quad (16)$$

Since $k>0$, $\tilde{p}$ and $\tilde{q}$ have common support $\mathbb{R}$.

Since all densities $p(x)$ are absolutely integrable, the Fourier transforms $\mathcal{F}\{p\}$ and $\mathcal{F}\{q\}$ exist. Assuming $\mathcal{F}\{k\}$ also exists, we can then use the convolution theorem to write

$$\mathcal{F}\{\tilde{p}\} = \mathcal{F}\{k\}\mathcal{F}\{p\}, \quad \mathcal{F}\{\tilde{q}\} = \mathcal{F}\{k\}\mathcal{F}\{q\}. \quad (17)$$

Let[7] $\mathcal{F}\{k\} \neq 0$. Then

$$\mathcal{F}\{k\}\mathcal{F}\{p\} = \mathcal{F}\{k\}\mathcal{F}\{q\} \Rightarrow \mathcal{F}\{p\} = \mathcal{F}\{q\}. \quad (18)$$

Using this we can show that the stationary Spread Divergence is valid. A sketch of the proof is as follows:

$$\tilde{\mathrm{D}}(p||q) = 0 \Leftrightarrow \mathrm{D}(\tilde{p}||\tilde{q}) = 0 \quad (19)$$

$$\Leftrightarrow \tilde{p} = \tilde{q} \ a.e. \quad (20)$$

$$\Leftrightarrow \mathcal{F}\{k\}\mathcal{F}\{p\} = \mathcal{F}\{k\}\mathcal{F}\{q\} \quad (21)$$

$$\Leftrightarrow \mathcal{F}\{p\} = \mathcal{F}\{q\} \quad (22)$$

$$\Leftrightarrow p = q \ a.e. \Leftrightarrow \mathbb{P} = \mathbb{Q}, \quad (23)$$

where we used the invertibility of the Fourier transform.

Hence we can define a valid Spread Divergence provided (i) $k(x)$ is a positive probability density function with support $\mathbb{R}$ and (ii) $\mathcal{F}\{k\} \neq 0$.

As an example consider Gaussian additive spread noise $k(x) = \mathcal{N}(x|0, \sigma^2)$. This has Fourier transform

$$\mathcal{F}\{k\}(\omega) = e^{-\frac{\sigma^2\omega^2}{2}} > 0. \quad (24)$$

Similarly, for Laplace noise $k(x) = \frac{1}{2b}e^{-\frac{1}{b}|x|}$,

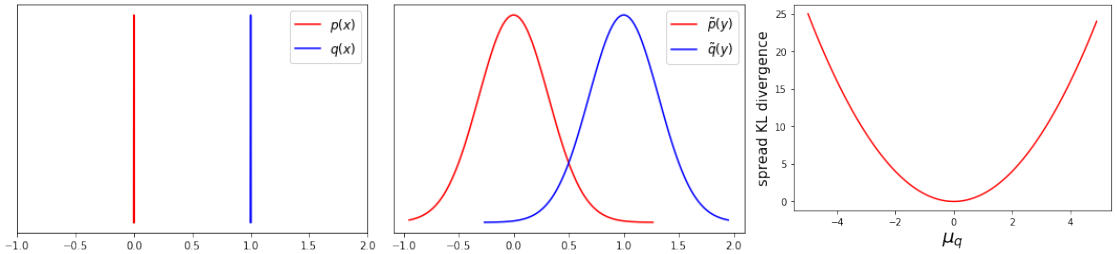
$$\mathcal{F}\{k\}(\omega) = \sqrt{\frac{2}{\pi}}\frac{b^{-1}}{b^{-2} + \omega^2} > 0. \quad (25)$$

In both cases $k>0$ and $\mathcal{F}\{k\}>0$. Additive Gaussian and Laplace noise can therefore be used to define a valid Spread Divergence. That is, if the divergence between the spreaded distributions is zero, then the original distributions are equal.

2.1.2. General Stationary Case

The previous additive noise setting assumed that X is absolutely continuous. In appendix(C) we show how to extend this to all distributions, including implicit generative models. We show that adding absolutely continuous noise K to X defines an absolutely continuous random variable $X + K$, even if X is itself not absolutely continuous (which is the case for implicit generative models). Applying the same additive noise process to both $\mathbb{P}$ and $\mathbb{Q}$ then results in absolutely continuous distributions that have densities $\tilde{p}$ and $\tilde{q}$ and common support. We further show that, provided the characteristic function of the additive random noise variable

[7] In fact the more general proof in appendix(C) shows that only the weaker condition that $\mathcal{F}\{k\} \neq 0$ on a countable set is needed.



(a) Two delta distributions. (b) Spreaded delta distributions. (c) Spread KL divergence

Figure 2. (a) Delta distributions $p(x) = \delta(x - \mu_p)$, $q(x) = \delta(x - \mu_q)$ where $\mu_p = 0$, $\mu_q = 1$. (b) Spreaded distributions $\tilde{p}(y) = \int p(y|x)p(x)dx$, $\tilde{q}(y) = \int p(y|x)q(x)dx$, where $p(y|x) = \mathcal{N}\left(y|x, \sigma^2 = 0.5\right)$. (c) The spread KL divergence as a function of μ_q.

K is non-zero[8], the noise K can be used to define a valid Spread Divergence between any two distributions with the property that $\tilde{\mathrm{D}}_f(\mathbb{P}||\mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$. This non-zero requirement for the characteristic function is analogous to the characteristic condition on kernels such that the Maximum Mean Discrepancy $\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$, see (Sriperumbudur et al., 2011; 2012; Gretton et al., 2012).

As an illustration, consider the extreme case of two delta distributions $\mathbb{P}$ and $\mathbb{Q}$ with generalised densities

$$p(x) = \delta(x - \mu_p), \quad q(x) = \delta(x - \mu_q). \tag{26}$$

In this case $\mathrm{KL}(p||q)$ is not well defined. For stationary Gaussian noise $p(y|x) = \mathcal{N}\left(y|x, \sigma^2\right)$, the 'spreaded' distributions are:

$$\tilde{p}(y) = \int \delta(x - \mu_p)\mathcal{N}\left(y|x, \sigma^2\right) dx = \mathcal{N}\left(y|\mu_p, \sigma^2\right),$$
$$\tilde{q}(y) = \int \delta(x - \mu_q)\mathcal{N}\left(y|x, \sigma^2\right) dx = \mathcal{N}\left(y|\mu_q, \sigma^2\right).$$

For noise variance $\sigma^2 = 0.5$ this gives:

$$\mathrm{KL}(\tilde{p}||\tilde{q}) = ||\mu_p - \mu_q||_2^2. \tag{27}$$

Hence $\mathrm{KL}(\tilde{p}||\tilde{q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$. It is also worth noting that the spread KL divergence in this case is equal to the squared 2-Wasserstein distance (Peyré et al., 2019; Gelbrich, 1990). Treating μ_q as a variable, figure(2) illustrates the spread KL divergence converging to 0 as μ_q tends to $\mu_p = 0$.

This treatment of generalised densities allows us to define a divergence for implicit generative models and, by extension, an associated training algorithm, as we describe below.

2.2. Spread Maximum Likelihood Estimation

In section(1.1.1) we noted that in the context of fitting an implicit generative model $\mathbb{Q}$ to training data, simply adding noise to the model distribution $\mathbb{Q}$ and using maximum likelihood does not result in a consistent estimator. In the Spread Divergence case, for data $x_1, \ldots, x_N$ we define the empirical (generalised) density as

$$p(x) = \frac{1}{N}\sum_{n=1}^{N} \delta\left(x - x_n\right). \tag{28}$$

For spread noise $p(y|x)$ we then spread both the model $\mathbb{Q}$ and empirical density using

$$\tilde{q}(y) = \int p(y|x)q(x)dx, \quad \tilde{p}(y) = \int p(y|x)p(x)dx. \tag{29}$$

We then define the spread log likelihood using

$$\tilde{\mathcal{L}} \equiv \int \tilde{p}(y) \log \tilde{q}(y) dy. \tag{30}$$

We consider that the data $x_1, \ldots, x_N$ is generated from the same underlying parametric implicit distribution as the model $\mathbb{Q}$

$$m(x; \theta_q) = \int \delta(x - g_{\theta_q}(z))q(z)dz, \tag{31}$$

but with a different parameters θ_p. Then as $N \to \infty$ (using the law of large numbers) the spread log likelihood becomes

$$\lim_{N\to\infty} \mathcal{L} = \int \tilde{m}(y; \theta_p) \log \tilde{m}(y; \theta_q) dy, \tag{32}$$

where

$$\tilde{m}(y; \theta) = \int p(y|x)m(x; \theta)dx. \tag{33}$$

Up to an additive constant this is $-\mathrm{KL}(\tilde{m}(y; \theta_p)||\tilde{m}(y; \theta_q))$. The spread log likelihood eq(32) therefore has a maximum when the spread KL divergence has a minimum. This occurs when the two spread distributions match $\tilde{m}(y; \theta_p) = \tilde{m}(y; \theta_q)$. Furthermore, by the property of the Spread Divergence, this means that the spread log likelihood has a maximum when the distributions match $m(y; \theta_p) = m(y; \theta_q)$,

[8] See appendix(C) for a weaker condition.

which occurs when $\theta_q = \theta_p$. Hence, the spread log likelihood defines a consistent estimator. In practice we typically cannot carry out the integral in eq(30) exactly, and resort to a sample approximation, sampling L noisy versions $y_{n,l}$, $l = 1, \ldots, L$ of each datapoint x_n to give

$$\tilde{\mathcal{L}} \approx \frac{1}{LN} \sum_{n=1}^{N} \sum_{l=1}^{L} \log \tilde{q}(y_{n,l}). \tag{34}$$

The Maximum Likelihood Estimator (MLE) is a cherished approach due to its consistency (convergence to the true parameters in the large data limit) and asymptotic efficiency (achieves the Cramér-Rao lower bound on the variance of any unbiased estimator) - see Casella & Berger (2002) for an introduction. An interesting question for future study is whether these properties also carry over to the spread MLE. In appendix(F), we demonstrate that spread MLE (for a certain family of spread noise) has weaker sufficient conditions than MLE for both consistency and asymptotic efficiency. Furthermore, a sufficient condition for the existence of the MLE is that the likelihood function is continuous over a compact parameter space Θ. We provide an example in appendix(F.1) where the maximum likelihood estimator may not exist (since it violates the compactness requirement), but the spread maximum likelihood estimator still exists.

3. Maximising Discriminatory Power

Intuitively, spreading out distributions makes them more similar. More formally, from the data processing inequality (see appendix(D)), using spread noise will always decrease the f-divergence $\mathrm{D}_f(\tilde{p}||\tilde{q}) \leq \mathrm{D}_f(p||q)$ (when $\mathrm{D}_f(p||q)$ is well defined). If we use spread MLE to train a model, too much noise may make the spreaded empirical and spreaded model distributions so similar that it becomes difficult to numerically distinguish them. It is useful therefore to learn spread noise $p_\psi(y|x)$ (parameterised by ψ) to maximally discriminate between the distributions $\max_\psi \mathrm{D}(\tilde{p}||\tilde{q})$. In general we need to constrain the spread noise to ensure that the divergence remains bounded. The learned noise will discourage overlap between the two spreaded distributions.

We discuss below two complementary approaches to adjust $p_\psi(y|x)$. The first adjusts the covariance for Gaussian $p(y|x)$ and the second uses a mean transformation. In principle, both approaches can be combined and easily generalised to other noise distributions, such as the Laplace distribution. In section(4.2), we empirically investigate the benefit of these approaches when scaling the application of Spread Divergence to complex high dimensional problems.

3.1. Learning the Gaussian Noise Covariance

In learning Gaussian stationary spread noise $p(y|x) = \mathcal{N}(y|x, \Sigma)$, the number of parameters in the covariance matrix Σ scales quadratically with the data dimension D. We therefore define $\Sigma = \sigma^2 I + UU^\mathsf{T}$ where $\sigma^2 > 0$ is fixed (to ensure bounded Spread Divergence) and U is a learnable $D \times R$ matrix with $R \ll D$.

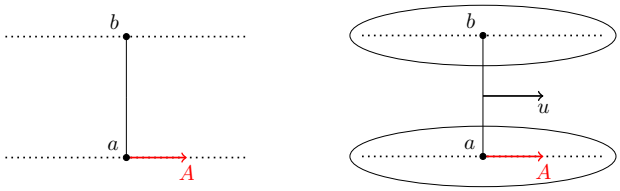


Figure 3. Left: The lower dotted line denotes Gaussian distributed data $p(x)$ with support only along the linear subspace defined by the origin a and direction A. The upper dotted line denotes Gaussian distributed data $q(x)$ with support different from $p(x)$. Optimally, to maximise the Spread Divergence between the two distributions, for fixed noise entropy, we should add noise that preferentially spreads out along the directions defined by p and q, as denoted by the ellipses.

As a simple example that can be computed exactly, we consider a implicit generative models with generalised densities p and q that generate data in separated linear subspaces,

$$p(x) = \int \delta(x - a - Az)\, p(z) dz \tag{35}$$

$$q(x) = \int \delta(x - b - Bz)\, p(z) dz, \tag{36}$$

with $p(z) = \mathcal{N}(z|0, I_Z)$. The spreaded densities are then

$$\tilde{p}(y) = \mathcal{N}\left(y|a, AA^\mathsf{T} + \Sigma\right), \quad \tilde{q}(y) = \mathcal{N}\left(y|b, BB^\mathsf{T} + \Sigma\right).$$

As Σ tends to zero, $\mathrm{KL}(\tilde{p}||\tilde{q})$ tends to infinity. We therefore constrain $\Sigma = \sigma^2 I + uu^\mathsf{T}$, where σ^2 is fixed and $u^\mathsf{T}u = 1$. Also, for calculational simplicity, we assume $A = B$. The Spread Divergence $\mathrm{KL}(\tilde{p}||\tilde{q})$ is then maximised for the noise direction u pointing orthogonal to the vector $\left(AA^\mathsf{T} + \sigma^2 I\right)^{-1}(b - a)$. The noise thus concentrates along directions defined by p and q, see figure(3).

3.2. Learning a Mean Transformation

Consider spread noise $p(y|x) = k(y - f(x))$ for injective[9] f and stationary k. Then, we define

$$\tilde{p}(y) = \int k(y - f(x)) p_x(x) dx. \tag{37}$$

Using a change of variables,

$$\tilde{p}(y) = \int k(y - z) p_z(z) dz, \tag{38}$$

$$p_z(z) = p_x(f^{-1}(z))/J\left(x = f^{-1}(z)\right), \tag{39}$$

[9]Since the co-domain of f is determined by its range, injective indicates invertible in this case.

where J is the absolute Jacobian of f. This is a valid Spread Divergence since eq(38) is in the form of standard stationary spread noise, but on a transformed variable z. Each injective f gives a different noise $p(y|x)$ and hence we can search for the best noise implicitly by learning f.

4. Applications

As an application of the spread MLE, we use it to train implicit generative models with generalised density

$$p_\theta(x) = \int \delta\left(x - g_\theta(z)\right) p(z) dz, \tag{40}$$

where θ are the parameters of the encoder g. We show that, despite the likelihood gradient not being available, we can nevertheless successfully train such models using slightly modified versions of standard likelihood based training approaches, such as variational algorithms (Barber, 2012). In particular we discuss learning a low dimensional linear ICA model and a high dimensional deep generative model.

4.1. Implicit Linear Models: Deterministic ICA

ICA (Independent Components Analysis) corresponds to the model $p(x,z) = p(x|z)\prod_i p(z_i)$, where the independent components z_i follow a non-Gaussian distribution. For Gaussian noise ICA an observation x is assumed to be generated by the process $p(x|z) = \prod_j \mathcal{N}\left(x_j|g_j(z), \gamma^2\right)$ where $g_i(z)$ mixes the independent latent process z. In linear ICA, $g_j(z) = a_j^\mathsf{T} z$ where a_j is the j^{th} column on the mixing matrix A. For zero observation noise $\gamma^2 = 0$ this corresponds to a linear implicit generative model. For training data $x_1, \ldots, x_N$ a standard maximum likelihood approach to learning A is to maximise

$$\mathbb{E}_{\hat{p}(x)}\left[\log p(x)\right], \tag{41}$$

where $p(x)$ is the marginal of the joint $p(x,z)$ and we define the empirical distribution

$$\hat{p}(x) = \frac{1}{N}\sum_{n=1}^{N} \delta\left(x - x_n\right). \tag{42}$$

Since this is a latent variable model, it is natural to apply the EM algorithm (Dempster et al., 1977) to learn A. However, for zero, or even small observation noise γ^2, it is well known that EM is ineffective (Bermond & Cardoso, 1999; Winther & Petersen, 2007). To see this, consider $\dim(X) = \dim(Z)$ (where $\dim(X)$ and $\dim(Z)$ are the dimension of the data and latents respectively) and invertible A. At iteration k the EM algorithm has an estimate A_k of the mixing matrix. It is straightforward to show that the M-step updates A_k to

$$A_{k+1} = \mathbb{E}\left[xz^\mathsf{T}\right]\mathbb{E}\left[zz^\mathsf{T}\right]^{-1}, \tag{43}$$

where, for zero observation noise ($\gamma = 0$),

$$\mathbb{E}\left[xz^\mathsf{T}\right] = \frac{1}{N}\sum_n x_n\left(A_k^{-1}x_n^\mathsf{T}\right) = \hat{S}A_k^{-\mathsf{T}}, \tag{44}$$

$$\mathbb{E}\left[zz^\mathsf{T}\right] = A_k^{-1}\hat{S}A_k^{-\mathsf{T}}, \tag{45}$$

and $\hat{S} \equiv \frac{1}{N}\sum_n x_n x_n^\mathsf{T}$ is the moment matrix of the data. Thus, $A_{k+1} = \hat{S}A_k^{-\mathsf{T}}\left(A_k^{-1}\hat{S}A_k^{-\mathsf{T}}\right)^{-1} = A_k$ and the algorithm 'freezes'. Similarly, for low noise $\gamma \ll 1$, progress critically slows down. Thus trying to train a deterministic ICA model by simply adding noise to the model will not help directly (see also section(1.1.1)).

To deal with small noise and the limiting case of a deterministic model ($\gamma = 0$), we consider Gaussian spread noise $p(y|x) = \mathcal{N}\left(y|x, \sigma^2 I_X\right)$ to give

$$p(y,z) = \int p(y|x)p(x,z)dx \tag{46}$$

$$= \prod_j \mathcal{N}\left(y_j|g_j(z), \left(\gamma^2 + \sigma^2\right) I_X\right)\prod_i p(z_i). \tag{47}$$

Using spread noise, the empirical distribution is replaced by the spreaded empirical distribution

$$\hat{p}(y) = \frac{1}{N}\sum_n \mathcal{N}\left(y|x^n, \sigma^2 I_X\right). \tag{48}$$

We then learn the model parameters by maximising the spread log likelihood (see section(2.2))

$$\mathbb{E}_{\hat{p}(y)}\left[\log p(y)\right]. \tag{49}$$

Since this is of the form of a latent variable model, we can use an EM algorithm to maximise eq(49). The M-step to maximise the spread log likelihood has the same form as eq(43) but with modified statistics

$$\mathbb{E}\left[yz^\mathsf{T}\right] = \frac{1}{N}\sum_n \int p(y,z|n) yz^\mathsf{T} dzdy, \tag{50}$$

$$\mathbb{E}\left[zz^\mathsf{T}\right] = \frac{1}{N}\sum_n \int p(y,z|n) zz^\mathsf{T} dzdy. \tag{51}$$

where

$$p(y,z|n) \equiv \mathcal{N}\left(y|x^n, \sigma^2\right) p(z|y) \tag{52}$$

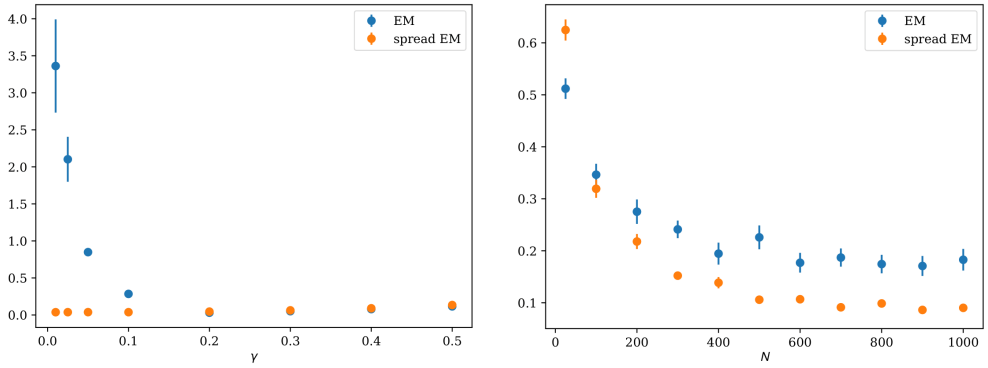
with

$$p(z|y) = \frac{1}{Z_q(y)}\mathcal{N}\left(z|\mu(y), \Sigma\right)\prod_i p(z_i), \tag{53}$$

$$Z_q(y) = \int \mathcal{N}\left(z|\mu(y), \Sigma\right)\prod_i p(z_i) dz, \tag{54}$$

Here $Z_q(y)$ is a normaliser and

$$\Sigma = \left(\gamma^2 + \sigma^2\right)\left(A^\mathsf{T}A\right)^{-1}, \quad \mu(y) = \left(A^\mathsf{T}A\right)^{-1}Ay. \tag{55}$$



(a) Error versus observation noise γ.

(b) Error versus number of training points.

Figure 4. Relative error $|A_{ij}^{est} - A_{ij}^{true}|/|A_{ij}^{true}|$ versus observation noise (a) and number of training points (b). (a) For $\dim(X) = 20$ dimensional observations and $\dim(Z)$=10 dimensional latent variables, we generate N=20000 data points from the model $x = Az + \gamma\mathcal{N}(0, I_X)$, for independent zero mean unit variance Laplace z. The elements of A are uniform random ± 1. We use S_y=1, S_z=1000 samples and 2000 EM iterations to estimate A. The error is averaged over all i, j and 10 experiments. We also plot standard errors around the mean relative error. In blue we show the error in learning A using the standard EM algorithm. As $\gamma \to 0$, the error blows up as the EM algorithm 'freezes'. In orange we plot the error for spread noise EM; no slowing down occurs as the observation noise γ decreases. In (b), apart from small N, the spread EM algorithm error is lower than for the standard EM algorithm. Here $\dim(Z)$=5, $\dim(X)$=10, S_y=1, S_z=1000, $\gamma = 0.2$, with 500 EM updates used. Results are averaged over 50 runs of randomly drawn A.

Since the posterior $p(z|y)$ peaks around $\mathcal{N}(z|\mu(y), \Sigma)$, we rewrite eq(50) as

$$\mathbb{E}\left[yz^\mathsf{T}\right] = \frac{1}{N}\sum_n \int \mathcal{N}\left(y|x^n, \sigma^2 I_X\right)\mathcal{N}\left(z|\mu(y), \Sigma\right) \times \frac{\prod_i p(z_i)}{Z_q(y)} yz^\mathsf{T} dzdy \tag{56}$$

and similarly for $\mathbb{E}\left[zz^\mathsf{T}\right]$. Writing the expectations with respect to $\mathcal{N}(z|\mu(y), \Sigma)$ allows for a simple but effective importance sampling approximation focused on regions of high probability. We implement this update by drawing S_y samples from $\mathcal{N}\left(y|x_n, \sigma^2 I_X\right)$ and, for each y sample, we draw S_z samples from $\mathcal{N}(z|\mu(y), \Sigma)$. This scheme has the advantage over more standard variational approaches, see for example (Winther & Petersen, 2007), in that we obtain a consistent estimator of the M-step update for A.

We show results for a toy experiment in figure(4), learning the mixing matrix A in a deterministic non-square setting. Note that standard algorithms such as FastICA (Hyvärinen, 1999) fail in this case. The spread noise is set to $\sigma = \max(0.001, 2.5 * \text{sqrt}(\text{mean}(AA^\mathsf{T})))$. This modified EM algorithm thus learns a good approximation of A, with no critical slowing down.

4.2. Non-linear Implicit Models: δ-VAE

A deep implicit generative model has generalised density

$$p_\theta(x) = \int \delta\left(x - g_\theta(z)\right) p(z) dz, \tag{57}$$

where g_θ is a deep neural network, see for example (Goodfellow et al., 2014). As discussed, for $\dim(Z) < \dim(X)$ we cannot use standard maximum likelihood approaches to train this model. To address this, we consider using the spread MLE approach section(2.2). For training data $x_1, \ldots, x_N$ the empirical distribution is

$$\hat{p}(x) = \frac{1}{N}\sum_{n=1}^N \delta\left(x - x_n\right). \tag{58}$$

For Gaussian spread noise $p(y|x) = \mathcal{N}\left(y|x, \sigma^2 I_X\right)$, the spreaded empirical distribution is

$$\tilde{p}(y) = \frac{1}{N}\sum_{n=1}^N \mathcal{N}\left(y|x_n, \sigma^2 I_X\right), \tag{59}$$

and the spreaded model is

$$\tilde{p}_\theta(y) = \int \mathcal{N}\left(y|g_\theta(z), \sigma^2 I_X\right) p(z) dz \tag{60}$$

$$= \int p_\theta(y|z) p(z) dz. \tag{61}$$

We then maximise the spread log likelihood

$$\int \tilde{p}(y) \log \tilde{p}_\theta(y) dy. \tag{62}$$

Typically, the integral over y is intractable, in which case we resort to a sampling estimation

$$\frac{1}{NS}\sum_{n=1}^N \sum_{s=1}^S \log \tilde{p}_\theta(y_s^n), \tag{63}$$

(a) Fixed Laplace Spread Noise

(b) Fixed Gaussian Spread Noise

(c) Learned Gaussian Spread Noise

Figure 5. Samples from a deep implicit generative model trained using δ-VAE with (a) Laplace spread noise with fixed covariance, (b) Gaussian spread noise with fixed covariance and (c) Gaussian spread noise with learned covariance. We first train with one epoch a standard VAE as initialization to all models and keep the latent code $z \sim \mathcal{N}(z|0, I_Z)$ fixed when sampling from these models thereafter, so we can more easily compare the sample quality. See also figure(8) for further samples.

(a) δ Fixed spread noise (b) δ Learned spread noise

Figure 6. Samples from a deep implicit generative model trained using δ-VAE with (a) fixed and (b) learned spread with the mean transformation method. See also figure(9) for more samples. We use a similar sampling strategy as in the MNIST experiment to facilitate sample comparison between the different models – see appendix(I).

where y_s^n is a sample from $p(y_n|x_n) = \mathcal{N}\left(y^n|x_n, \sigma^2 I_X\right)$.

For non-linear g, the distribution $\tilde{p}_\theta(y)$ is usually intractable and we therefore use the variational lower bound

$$\log \tilde{p}_\theta(y) \geq \int q_\phi(z|y)(-\log q_\phi(z|y) + \log\left(p_\theta(y|z)p(z)\right))dz. \quad (64)$$

This approach is a straightforward extension of the standard variational autoencoder (VAE) method (Kingma & Welling, 2013) and in appendix(G) we derive a lower variance objective and detail the learning procedure. We dub this model and associated Spread Divergence training the 'δ-VAE'.

For learning the spread noise we use the approach outlined in section(3). To learn the mean transformation function, we used an invertible residual network (Behrmann et al., 2018) $f_\psi : \mathbb{R}^D \to \mathbb{R}^D$ where $f_\psi = (f_\psi^1 \circ \ldots \circ f_\psi^T)$ denotes a ResNet with blocks $f_\psi^t = I(\cdot) + g_{\psi_t}(\cdot)$. Then f_ψ is invertible if the Lipschitz-constants $Lip(g_{\psi_t})<1$, for all $t \in \{1, \ldots, T\}$. Note that when using the Spread Divergence for training we only need samples from $\tilde{p}(y)$ which can be obtained from eq(37) by first sampling x from $p_x(x)$ and then y from $p(y|x) = k(y - f(x))$; this does not require computing the Jacobian or inverse ${f_\psi}^{-1}$.

MNIST Experiment: We trained a δ-VAE on MNIST (LeCun et al., 2010) with (i) fixed Laplace spread noise, as in eq(25), (ii) fixed Gaussian spread noise, as in eq(24) and (iii) Gaussian noise with learned covariance, as in section(3.1), with rank $R = 20$; $g_\theta(\cdot)$ is a neural network that contains 3 feed forward layers. See appendix(H) for further details.

Figures 5(a,b,c) show samples from $p_\theta(x)$ for these models. MNIST is a relatively easy problem in the sense that it is hard to distinguish between the quality of the fixed and learned noise samples. We speculate that Laplace noise appears to improve image sharpness since this noise focuses attention on discriminating between points close to the data manifold (since the Laplace distribution is leptokurtic and has a higher probability of generating points close to the data manifold than the Gaussian distribution).

CelebA Experiment: We trained a δ-VAE on the CelebA dataset (Liu et al., 2015) with (i) fixed and (ii) learned spread using the mean transformation method as discussed in section(3.2). We compare to results from a standard VAE with fixed Gaussian noise $p(x|z) = \mathcal{N}(x|g_\theta(z), 0.5 I_X)$ (Tolstikhin et al., 2017), where $g_\theta(\cdot)$ is a neural network contains 4 convolution layers.

For (i) the fixed Spread Divergence uses Gaussian noise $\mathcal{N}(y|x, 0.25 I_X)$. For (ii) we use Gaussian noise with a learned injective function in the form of a ResNet: $f_\psi(\cdot) = I(\cdot) + g_\psi(\cdot)$ - see appendix(I) for details. Figure 6 shows samples from our δ-VAE for (i) and (ii) (with $g_\theta(z)$ initialised to the fixed-noise setting). It is notable how the 'sharpness' of the image samples substantially increases when learning the spread noise. Table 1 shows Frechet Inception Distance (FID) (Heusel et al., 2017) score comparisons between different baseline algorithms for implicit generative model training[10]. The δ-VAE significantly improves on the standard VAE result. Learning the mean transformation improves on the fixed-noise δ-VAE. Indeed the mean transformation δ-VAE results are comparable to popular GAN and WAE models (Gulrajani et al., 2017; Berthelot et al., 2017; Arjovsky et al., 2017; Kodali et al., 2017; Mao et al., 2017; Fedus et al., 2017; Tolstikhin et al., 2017). Whilst the δ-VAE results are not state-of-the-art, we believe

[10]FID scores were computed using `github.com/bioinf-jku/TTUR` based on 10000 samples.

Encoder-Decoder Models	FID	GAN Models	FID
VAE	63.0	WGAN GP	30.0
δ-VAE with fixed spread	**52.7**	BEGAN	38.9
δ-VAE with learned spread	**46.5**	WGAN	41.3
		DRAGAN	42.3
WAE-MMD	55.0	LSGAN	53.9
WAE-GAN	42.0	NS GAN	55.0
		MM GAN	65.6

Table 1. CelebA FID Scores. The δ-VAE results are the average over 5 independent measurements. The scores of the GAN models are based on a large-scale hyperparameter search and take the best FID obtained (Lucic et al., 2018). The results of the VAE model and both WAE-based models are from (Tolstikhin et al., 2017).

it is the first time that implicit models have been trained using a principled maximum likelihood based approach. By increasing the complexity of the generative model g_θ and injective function f_ψ, or using better choices of noise, the results may become more competitive with state-of-the-art GAN models[11].

5. Related Work

Instance noise: The instance noise trick to stabilize GAN training (Roth et al., 2017; Sønderby et al., 2016) is a special case of Spread Divergence using fixed Gaussian noise. Whilst other similar tricks, e.g. (Furmston & Barber, 2009), have been proposed previously, we believe it is important to state the more general utility of the spread noise approach.

δ-VAE versus WAE: The Wasserstein Auto-Encoder (Tolstikhin et al., 2017) is another implicit generative model that uses an encoder-decoder architecture. The major difference to our work is that the δ-VAE is based on the KL divergence, which corresponds to MLE, whereas the WAE uses the Wasserstein distance.

δ-VAE versus denoising VAE: The denoising VAE (Im et al., 2017) uses a VAE with noise added to the data only. In contrast, for our δ-VAE, spread MLE adds noise to both the data and the model. Therefore, the denoising VAE cannot recover the true data distribution, whereas in principle the δ-VAE with spread MLE can.

MMD GAN with kernel learning: Learning a kernel to increase discrimination is also used in MMD GAN (Li et al., 2017). Similar to ours, the kernel in MMD GAN is constructed by $\tilde{k} = k \circ f_\psi$, where k is a fixed kernel and f_ψ is a neural network. To ensure the MMD distance $M_{k\circ f_\psi}(p, q) = 0 \Leftrightarrow p = q$, this requires f_ψ to be injective (Gretton et al., 2012). However, in this framework, $f_\psi(x)$ usually maps x to a lower dimensional space. This is crucial for MMD because the amount of data required to produce a reliable estimator grows with the data dimension (Ramdas et al., 2015) and the computational cost of MMD scales quadratically with the amount of data. Whilst using a lower-dimensional mapping makes MMD more practical it also makes it difficult to construct an injective function f. For this reason, heuristics such as the auto-encoder regularizer (Li et al., 2017) are considered. In contrast, for the δ-VAE with spread MLE, the cost of estimating the divergence is linear in the number of data points. Therefore, there is no need for f_ψ to be a lower-dimensional mapping; guaranteeing that f_ψ is injective is straightforward for the δ-VAE.

Flow-based generative models: Invertible flow-based functions (Rezende & Mohamed, 2015) have been used to boost the representation power of generative models. Our use of injective functions is quite distinct from the use of flow-based functions to boost generative model capacity. In our case, the injective function f does not change the model - it only changes the divergence. For this reason, the Spread Divergence doesn't require the log determinant of the Jacobian, which is required in (Rezende & Mohamed, 2015; Behrmann et al., 2018), meaning that more general invertible functions can be used to boost the discriminatory power of a Spread Divergence.

6. Summary

We described how to define a divergence even when two distributions don't have valid probability density functions or have the same support. We showed that defining divergences this way enables us to train implicit generative models using standard likelihood based approaches. In principle, we can learn the true data generating implicit generative model by using any valid Spread Divergence. In practice, however, the quality of the learned model can depend strongly on the choice of spread noise. We therefore investigated learning spread noise to maximally discriminate between two distributions. We found that the resulting training approach is numerically stable and that it can significantly improve the quality of the learned model.

There are several directions for further investigation: (i) the broader family of spread noise and their properties, including statistical efficiency; (ii) optimal noise selection for different tasks; (iii) the connections between Spread Divergence and other distance (or divergence) measure, e.g. MMD, Wasserstein Distance.

Acknowledgments

We would like to thank a reviewer for improving our consideration of non-absolutely continuous distributions and Li Zhu for useful discussions.

[11] We also tried learning the image generator using Laplace spread noise. However, the colour of the sampled images becomes overly intense and we leave it to future work to address this.

References

Ali, S. and Silvey, S. A General Class of Coefficients of Divergence of one Distribution from Another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. *arXiv:1701.07875*, 2017.

Barber, D. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

Behrmann, J., Duvenaud, D., and Jacobsen, J. Invertible Residual Networks. *arXiv:1811.00995*, 2018.

Bermond, O. and Cardoso, J. Approximate Likelihood for Noisy Mixtures. In *Proc. ICA '99*, pp. 325–330, 1999.

Berthelot, D., Schumm, T., and Metz, L. Began: Boundary Equilibrium Generative Adversarial Networks. *arXiv:1703.10717*, 2017.

Casella, G. and Berger, R. *Statistical Inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

Cramér, H. *Mathematical Methods of Statistics*, volume 9. Princeton University Press, 1999.

Csiszár, I. Information-type Measures of Difference of Probability Distributions and Indirect Observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

Csiszár, I. A Class of Measures of Informativity of Observation Channels. *Periodica Mathematica Hungarica*, 2 (1-4):191–213, 1972.

Dempster, A., Laird, N., and Rubin, D. Maximum Likelihood from Incomplete Data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Dragomir, S. Some General Divergence Measures for Probability Distributions. *Acta Mathematica Hungarica*, 109(4):331–345, Nov 2005. ISSN 1588-2632. doi: 10.1007/s10474-005-0251-6.

Durrett, R. *Probability: Theory and Examples*, volume 49. Cambridge university press, 2019.

Fedus, W., Rosca, M., Lakshminarayanan, B., Dai, A., Mohamed, S., and Goodfellow, I. Many Paths to Equilibrium: GANs Do Not Need to Decrease a Divergence at Every Step. *arXiv:1710.08446*, 2017.

Ferguson, T. An Inconsistent Maximum Likelihood Estimate. *Journal of the American Statistical Association*, 77 (380):831–834, 1982.

Furmston, T. and Barber, D. Solving Deterministic Policy (PO)MPDs using Expectation-Maximisation and Antifreeze. In *First international workshop on learning and data mining for robotics (LEMIR)*, pp. 56–70, 2009. In conjunction with ECML/PKDD-2009.

Gelbrich, M. On a Formula for the L2 Wasserstein Metric Between Measures on Euclidean and Hilbert Spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990.

Gerchinovitz, S., Ménard, P., and Stoltz, G. Fano's Inequality for Random Variables. *arXiv*, 2018. doi: arXiv:1702.05985v2.

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

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

Hyvärinen, A. Fast and Robust Fixed-point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, May 1999. ISSN 1045-9227. doi: 10.1109/72.761722.

Im, D., Ahn, S., Memisevic, R., and Bengio, Y. Denoising Criterion for Variational Auto-Encoding Framework. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Ioffe, S. and Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167*, 2015.

Kallenberg, O. *Foundations of Modern Probability*. Springer Science & Business Media, 2006.

Kingma, D. and Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2014.

Kingma, D. and Welling, M. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [stat.ML]*, 2013.

Kodali, N., Abernethy, J., Hays, J., and Kira, Z. On Convergence and Stability of GANs. *arXiv:1705.07215*, 2017.

LeCun, Y., Cortes, C., and Burges, C. MNIST Handwritten Digit Database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18, 2010.

Lehmann, E. *Elements of Large-sample Theory*. Springer Science & Business Media, 2004.

Li, C., Chang, W., Cheng, Y., Yang, Y., and Póczos, B. MMD GAN: Towards Deeper Understanding of Moment Matching Network. In *Advances in Neural Information Processing Systems*, pp. 2203–2213, 2017.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. Are GANs Created Equal? A Large-Scale Study. In *Advances in Neural Information Processing Systems*, pp. 700–709, 2018.

Mao, X., Li, Q., Xie, H., Lau, R., Wang, Z., and S., P. Least Squares Generative Adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, 2017.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral Normalization for Generative Adversarial Networks. *arXiv:1802.05957*, 2018.

Mohamed, S. and Lakshminarayanan, B. Learning in Implicit Generative Models. *arXiv*, 2016. doi: arXiv: 1610.03483.

Peyré, G., Cuturi, M., et al. Computational Optimal Transport. *Foundations and Trends® in Machine Learning*, 11 (5-6):355–607, 2019.

Ramdas, A., Reddi, S., Póczos, B., Singh, A., and Wasserman, L. On the Decreasing Power of Kernel and Distance Based Nonparametric Hypothesis Tests in High Dimensions. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Rezende, J. and Mohamed, S. Variational Inference with Normalizing Flows. *arXiv:1505.05770*, 2015.

Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. Stabilizing Training of Generative Adversarial Networks through Regularization. pp. 2018–2028, 2017.

Sønderby, C., Caballero, J., Theis, L., Shi, W., and Huszár, F. Amortised Map Inference for Image Super-Resolution. *arXiv:1610.04490*, 2016.

Sriperumbudur, B., Fukumizu, K., and Lanckriet, G. Universality, Characteristic Kernels and RKHS Embedding of Measures. *J. Mach. Learn. Res.*, 12:2389–2410, July 2011. ISSN 1532-4435.

Sriperumbudur, B., Fukumizu, K., Gretton, A., Schölkopf, B., and Lanckriet, G. On the Empirical Estimation of Integral Probability Metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

Tao, T. An Introduction to Measure Theory. 2011.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schölkopf, B. Wasserstein Auto-Encoders. *arXiv:1711.01558*, 2017.

Wald, A. Note on the Consistency of the Maximum Likelihood Estimate. *The Annals of Mathematical Statistics*, 20(4):595–601, 1949.

Winther, O. and Petersen, K. Bayesian Independent Component Analysis: Variational Methods and Non-negative Decompositions. *Digital Signal Processing*, 17(5):858 – 872, 2007. ISSN 1051-2004. Special Issue on Bayesian Source Separation.

Wu, Y., Burda, Y., Salakhutdinov, R., and Grosse, R. On the Quantitative Analysis of Decoder-based Generative Models. *arXiv:1611.04273*, 2016.

Zhang, M., Bird, T., Habib, R., Xu, T., and Barber, D. Variational f-Divergence Minimization. *arXiv:1907.11891*, 2018.