

Revisiting Classifier Two-Sample Tests for GAN Evaluation and Causal Discovery

David Lopez-Paz, Maxime Oquab
Facebook AI Research
{dlp,qas}@fb.com

October 16, 2016

Abstract

The goal of two-sample tests is to decide whether two probability distributions, denoted by P and Q , are equal. One alternative to construct flexible two-sample tests is to use binary classifiers. More specifically, pair n random samples drawn from P with a positive label, and pair n random samples drawn from Q with a negative label. Then, the test accuracy of a binary classifier on these data should remain near chance-level if the null hypothesis “ $P = Q$ ” is true. Furthermore, such test accuracy is an average of independent random variables, and thus approaches a Gaussian null distribution. Furthermore, the prediction uncertainty of our binary classifier can be used to interpret the particular differences between P and Q . In particular, analyze which samples were correctly or incorrectly labeled by the classifier, with the least or most confidence.

In this paper, we aim to revive interest in the use of binary classifiers for two-sample testing. To this end, we review their fundamentals, previous literature on their use, compare their performance against alternative state-of-the-art two-sample tests, and propose them to evaluate generative adversarial network models applied to image synthesis.

As a by-product of our research, we propose the application of conditional generative adversarial networks, together with classifier two-sample tests, as an alternative to achieve state-of-the-art causal discovery.

1 Introduction

Generative models are a fundamental component in a variety of important machine learning tasks. These include feature compression, image synthesis and completion, semi-supervised learning, un-supervised learning, and density estimation, to name a few. Due to their many uses, evaluating and comparing generative models is a problem-specific task (Theis et al., 2015).

In this paper, we are interested in evaluating the quality of the samples synthesized by generative models with intractable likelihood, such as Generative Adversarial Networks or GANs (Goodfellow et al., 2014). Formally, evaluating sample quality is a *two-sample test*, that is, measuring the dissimilarities between the data distribution being modeled and the samples synthesized by our generative model. This paper aims at reviving the interest in using binary

classifiers as two-sample tests. In particular, since good generative models will produce samples barely indistinguishable from real data, the test accuracy of a binary classifier tasked with distinguishing real data from synthesized samples should remain at chance level.

The rest of this article is **organized** as follows. Section 2 introduces the fundamentals of two-sample tests, as well as their most common uses. Section 3 reviews classifier two-sample tests or, said differently, the use of binary classifiers as two sample tests. Section 4 provides a series of experiments to evaluate the performance of classifier two-sample tests. In particular, we i) compare their performance against alternative state-of-the-art two-sample tests, ii) propose and describe their use to evaluate generative models with intractable likelihoods, such as generative adversarial networks, and iii) propose their use, in conjunction with conditional generative adversarial network, to achieve state-of-the-art cause-effect discovery from observational data.

2 Two-sample testing

The goal of *two-sample tests* is to decide whether two probability distributions, denoted by P and Q , are equal (Lehmann & Romano, 2006). To this end, two-sample tests analyze the independently and identically distributed (iid) samples

$$\begin{aligned} x_1, \dots, x_n &\sim P(X), \\ y_1, \dots, y_m &\sim Q(Y), \end{aligned} \tag{1}$$

and summarize the differences between $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^m$ into a statistic $\hat{t} \in \mathbb{R}$. Then, for small values of \hat{t} , the two-sample test will accept the *null hypothesis* H_0 , which stands for “ P is equal to Q ”. Conversely, for large values of \hat{t} , the two-sample test will reject H_0 in favour of the *alternative hypothesis* H_1 , which stands for “ P is not equal to Q ”.

Formally, the statistician performs a two-sample test in four steps. First, the statistician chooses a *significance level* $\alpha \in [0, 1]$. Second, the statistician computes the two-sample test statistic \hat{t} . Third, the statistician computes the *p-value* $\hat{p} = P(T \geq \hat{t} | H_0)$, which is the probability of the two-sample test returning an statistic larger or equal than \hat{t} when the null hypothesis H_0 is true. Fourth, the statistician accepts the null hypothesis H_0 if the $\hat{p} < \alpha$, and accepts the alternative hypothesis H_1 otherwise. As a mandatory cautionary note, we remind that the *p-values* is not the probability of the null hypothesis being true, and that the results of statistical testing depend on both the significance level and the particular two-sample test under use (Johnson, 1999).

Inevitably, two-sample tests can fail in two different ways. First, to make a Type I error is to reject the null hypothesis when it is true (a “false positive”). Second, to make a Type II error is to accept the null hypothesis when it is false (a “false negative”). The probability of making a Type II error is denoted by β , and we refer to the quantity $\pi = 1 - \beta$ as the *power* of a test. Usually, the statistician uses domain-specific knowledge to upper-bound the probability of making a Type I error by the significance level α . Within the significance level α , a statistician would prefer the two-sample test minimizing the probability of making a Type II error, that is, maximizing power π .

The literature has evolved a variety of two-sample tests to vast to enumerate here. These include the t -test (Student, 1908) which tests for the difference in means of two samples; the Wilcoxon-Mann-Whitney test (Wilcoxon, 1945; Mann & Whitney, 1947), which tests for the difference in rank means of two samples; the Kolmogorov-Smirnov test (Kolmogorov, 1933; Smirnov, 1939), which tests for the difference in the empirical cumulative distributions of two samples; and the Maximum Mean Discrepancy or MMD (Gretton et al., 2012a), which tests for the difference in the empirical kernel mean embeddings of two samples. Notably, the MMD test is the only one from this list applicable to data supported in arbitrary domains, thanks to the use of kernels.

Apart from their straightforward application, two-sample tests offer two additional uses:

1. Two-sample tests can be used to *test for independence*, as pointed out by (Gretton et al., 2012a). In particular, testing the independence null hypothesis “the random variables X and Y are independent” translates into testing the two-sample null hypothesis “ $P(X, Y)$ is equal to $P(X)P(Y)$ ”.
2. Two-sample tests can be used to *evaluate generative models* with intractable likelihoods but tractable sampling procedures. Intuitively, good generative models should produce samples $\hat{S} = \{\hat{x}_i\}_{i=1}^n$ indistinguishable from data $S = \{x_i\}_{i=1}^n$ that they model. Thus, two-sample tests between \hat{S} and S can be used as metrics to evaluate the fidelity of the samples \hat{S} . Examples of such use of two-sample tests include the pioneering work of Early examples of such use of two-sample tests include (Box, 1980) or, more recently, the use of the MMD two-sample test to evaluate the quality of complex generative models Dziugaite et al. (2015); Lloyd & Ghahramani (2015), an idea also mentioned in (Bengio et al., 2013).

3 Classifier two-sample tests

Next, we discuss a general and flexible way to build powerful two-sample test: the simple use of binary classifiers. In particular, we assume access to the iid samples (1) where $x_i, y_j \in \mathcal{X}$, for all $i = 1, \dots, n$ and $j = 1, \dots, m$. To test whether $P = Q$, we proceed in four steps. First, construct the dataset

$$\mathcal{D} = \{(x_i, 0)\}_{i=1}^n \cup \{(y_i, 1)\}_{i=1}^m := \{(z_i, l_i)\}_{i=1}^{n+m}.$$

Second, shuffle \mathcal{D} at random, and split it into the disjoint subsets \mathcal{D}_{tr} and \mathcal{D}_{te} , such that $\mathcal{D} = \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{te}}$ and $n_{\text{te}} := |\mathcal{D}_{\text{te}}|$. Third, train a binary classifier $f : \mathcal{X} \rightarrow [0, 1]$ on \mathcal{D}_{tr} . In the sequel, we assume that $f(z_i)$ is an estimation of the conditional probability distribution $p(l = 1|z_i)$. Fourth, return the binary classifier classification accuracy on \mathcal{D}_{te} :

$$\hat{t} = \frac{1}{n_{\text{te}}} \sum_{(z_i, l_i) \in \mathcal{D}_{\text{te}}} \mathbb{I} \left[\left(f(z_i) > \frac{1}{2} \right) = l_i \right] \quad (2)$$

as our two-sample test statistic. The intuition here is that if $P = Q$, the test accuracy (2) should remain near chance-level, that is, one half. In opposition, if $P \neq Q$, the differences between the samples of the two distributions would

be unveiled by the binary classifier, which will in turn translate into a test classification accuracy (2) greater than one half. We call tests based on (2) *Classifier Two-Sample Tests* (C2ST), and review some of their properties next.

3.1 Null distribution

Under the null hypothesis, the samples drawn from P and Q are indistinguishable from each other, rendering an impossible binary classification problem. Thus, under the null hypothesis and regardless of the shape of the binary classifier f , each term $\mathbb{I}[(f(z_i) > 1/2) = l_i]$ appearing in (2) is an independent random variable following a Bernoulli(0.5) distribution. Therefore, the null distribution of $n_{te}\hat{t}$ will follow a Binomial($n_{te}, 0.5$) distribution. Alternatively, by applying the central limit theorem, it follows that the null distribution of $\hat{t} \xrightarrow{d} \mathcal{N}(0.5, 0.25/n_{te})$, as $n_{te} \rightarrow \infty$.

3.2 Power

The power of a test is its probability of rejecting false null hypotheses. Since the null distribution does not depend on the architecture of the classifier, maximizing the power of neural two-sample tests is a trade-off between i) maximizing the test accuracy of the classifier (bias), and ii) maximizing the size of the test set n_{te} (variance). This is of course a well known trade-off in machine learning. On the one hand, simple (underfitting) classifiers will miss some nonlinear patterns, leading to type-II errors and low power. However, simple classifiers call for less training data, leading to larger test sets. On the other hand, flexible (overfitting) classifiers may hallucinate patterns from noise, leading to type-I errors. However, flexible classifiers will minimize type-II errors at the expense of more data.

The power of classifier two-sample tests is minimax-optimal under some conditions (Ramdas et al., 2016). More specifically, the power of a classifier two-sample test is directly linked to its generalization error, which has a sample complexity upper-upper bounded as $O(n^{-1/2})$. On the other hand, the sample complexity of some simple two-sample tests, such as the difference between the means of two multivariate Gaussians, has a sample complexity lower-bounded as $O(n^{-1/2})$ (Ramdas et al., 2015).

3.3 Interpretability

We have assumed that $f(z_i)$ estimates $p(l = 1|z_i)$ for each of the samples z_i on the test set. Inspecting these probabilities, together with the true labels l_i , allow us to determine which samples were correctly or wrongly labeled by the classifier, with the least or the most confidence. This analysis provides insight about which specific samples make the probability distributions P and Q similar or dissimilar. Therefore, the statistic (2) does not only measure the dissimilarity between two probability distributions; it also explains *where* the two distributions are similar or different.

3.4 Prior uses

The reduction of two-sample testing to binary classification follows from Friedman (2003), is studied in great detail by Reid & Williamson (2011), and very recently reviewed by (Mohamed & Lakshminarayanan, 2016). The same discussion and further references appear in (Gretton et al., 2012a, Remark 20). In practice, the use of binary classifiers for two-sample testing is increasingly common in neuroscience (see Pereira et al. (2009); Olivetti et al. (2012) and the references therein). Implicitly, binary classifiers also perform two-sample testing in algorithms that aim at discriminating data from noise, such as noise contrastive estimation (Gutmann & Hyvärinen, 2012), negative sampling (Mikolov et al., 2013), and generative adversarial networks (Goodfellow et al., 2014).

4 Numerical simulations

In our experiments, we study samples $\mathcal{Z} = \{z_1, \dots, z_n\}$, where $z_i = (x_i, y_i)$, $x_i \sim P(X)$, $y_i \sim Q(X)$, and $x_i, y_i \in \mathbb{R}^d$, for all $1 \leq i \leq n$. We study two variants of C2ST: one based on neural network binary classifiers (C2ST-NN), and one based on k -nearest neighbour binary classifiers (C2ST-KNN). C2ST-NN uses a binary classifier with one hidden layers of 128 ReLU neurons, and is trained using the Adam optimizer (Kingma & Ba, 2015) with $\beta_1 = 0.5$. C2ST-KNN uses $k = \lfloor n_{\text{tr}}^{-1/2} \rfloor$ nearest neighbours for classification. When analyzing one-dimensional samples, we compare the performance of C2ST-NN and C2ST-KNN against the Wilcoxon-Mann-Whitney test (Wilcoxon, 1945; Mann & Whitney, 1947) and the Kolmogorov-Smirnov test (Kolmogorov, 1933; Smirnov, 1939). In all cases, we compare the performance of C2ST-NN and C2ST-KNN against the linear-time unbiased estimate of the Maximum Mean Discrepancy (MMD) criterion (Gretton et al., 2012a).

$$\text{MMD}(\mathcal{D}) = \frac{1}{\lfloor n/2 \rfloor} \sum_{i=1}^{\lfloor n/2 \rfloor} k(x_{2i-1}, x_{2i}) + k(y_{2i-1}, y_{2i}) - k(x_{2i-1}, y_{2i}) - k(x_{2i}, y_{2i-1}),$$

where $k : \mathcal{X} \times \mathcal{X}$ is a positive-definite kernel. We use a Gaussian kernel $k(x, y) = \exp(-\gamma \|x - y\|_2^2)$, where the bandwidth hyper-parameter $\gamma > 0$ is chosen to maximize test, using the “max-rat” rule from (Gretton et al., 2012b). Since the Gaussian kernel is a characteristic kernel, the MMD statistic approaches zero if and only if the null hypothesis (“ $P = Q$ ”) is true, as the sample size tends to infinity (Gretton et al., 2012a). We use a significance level $\alpha = 0.05$ across all experiments and tests.

Our code is available at https://github.com/lopezpaz/classifier_tests.

4.1 Two-sample testing

We deploy two synthetic experiments to evaluate the performance of C2ST when used for two-sample testing. First, we evaluate the correctness of all the considered two-sample tests (MMD, C2ST-KNN, C2ST-NN, Wilcoxon-Mann-Whitney, Kolmogorov-Smirnov) by examining if the specified significant level of each test correctly upper-bounds its Type I error. To do so, we draw $x_1, \dots, x_n, y_1, \dots, y_n \sim \mathcal{N}(0, 1)$, and run each two-sample test. In this setup,

a Type I error is to reject the null hypothesis, since the samples $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$ are drawn from the same distribution. Figure 1 f) shows that the Type I error of all tests is upper bounded by the pre-specified significance level, for all $n \in \{25, 50, 100, 500, 1000, 5000, 10000\}$, and across 100 random repetitions. Thus, all tests show the expected behaviour, in terms of Type I error control.

Our second experiment, considers a two sample test between a Normal distribution and a Student’s t distribution with ν degrees of freedom, based on n samples. Recall that the Student’s t distribution approaches the Normal distribution as ν increases. Therefore, two-sample tests must focus on the tails of the distribution to distinguish between Gaussian and Student’s samples. Figure 1 d,e) shows the test power of all tests as we vary $n \in \{100, 500, 1000, 5000, 10000\}$, and $\nu \in \{1, 2, 5, 10, 15, 20\}$. The Wilcoxon-Mann-Whitney exhibits the worst performance, as expected (since the ranks mean of the Gaussian and Student’s t distributions coincide) in this experiment. Kolmogorov-Smirnov, C2ST-NN, and C2ST-KNN tests exhibit the best performance, followed by the MMD test.

4.2 Independence testing

As mentioned in Section 2, we can use two-sample tests to measure statistical dependence, by defining the null distribution “ $P(X, Y)$ is equal to $P(X)P(Y)$ ”. We here compare the performance of the C2ST-NN, C2ST-KNN, and MMD tests in this task. Since the distribution $P(X, Y)$ is bivariate, we do not compare against the Wilcoxon-Mann-Whitney and Kolmogorov-Smirnov tests.

In particular, we will setup a generative model

$$\begin{aligned} x_i &\sim \mathcal{N}(0, 1), \\ \epsilon_i &\sim \mathcal{N}(0, \sigma^2), \\ y_i &\sim \cos(\nu x_i) + \epsilon_i, \end{aligned}$$

where we let x_i be iid samples from some random variable X , and y_i be iid samples from some random variable Y . Thus, the pair of random variables (X, Y) are statistically dependent, but the observable effect of such dependence weakens as we either i) increase the frequency ν of the sinusoid, or ii) increase the variance σ^2 of the additive noise. Figure 1 a,b,c) shows the test power of the C2ST-NN, C2ST-KNN, and MMD tests as we vary $n \in \{100, 500, 1000, 5000, 10000\}$, $\nu \in \{2, 4, 6, 8, 10\}$, and $\sigma \in \{0.1, 0.25, 0.5, 1, 2, 3\}$. The figure reveals that, in this experiment, the classifier two-sample tests have a better performance than the MMD test. For fairness, the MMD test is much faster to run than the C2ST tests. Moreover, the performance of MMD could be improved by using more sophisticated kernel functions.

4.3 Evaluation of GANs for image generation

Generative Adversarial Networks, or GANs (Goodfellow et al., 2014), are generative models implementing the adversarial game

$$\min_g \max_d \mathbb{E}_x \log(d(x)) + \mathbb{E}_z \log(1 - d(g(z))), \quad (3)$$

In the previous, $d(x)$ depicts the probability of the sample x being drawn from the data distribution, instead of synthesized by the generator. This is according

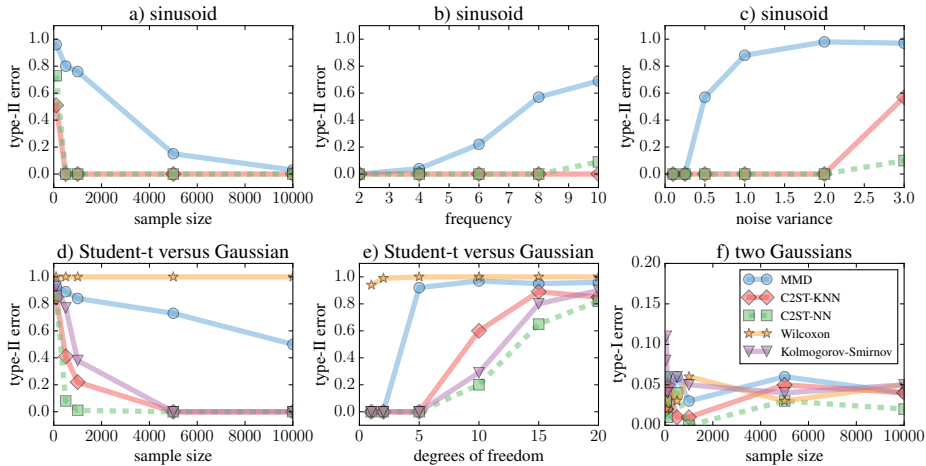


Figure 1: Results of the two-sample test experiments.

to a *discriminator* function d , which is also to be trained. In the adversarial game, the generator g plays to fool the discriminator d by synthesizing samples that look as real as possible, by transforming noise vectors $z \sim P(Z)$, $z \in \mathbb{R}^q$, into real-looking samples $g(x)$. On the other hand, the discriminator plays to distinguish between real samples x and synthesized samples $g(z)$ as best as possible. The adversarial game in GANs can be written in terms of two risk minimizations:

$$\begin{aligned}
 L_d(d) &= \mathbb{E}_x \ell(d(x), 1) + \mathbb{E}_z \ell(d(g(z)), 0), \\
 L_g(g) &= \mathbb{E}_x \ell(d(x), 0) + \mathbb{E}_z \ell(d(g(z)), 1) \\
 &= \mathbb{E}_z \ell(d(g(z)), 1).
 \end{aligned} \tag{4}$$

Under the formalization (6), the adversarial game is then reduced to the sequential minimization of $L_d(d)$ and $L_g(g)$, and reveals the true goal of the discriminator: to be the classifier two-sample test that best distinguishes data samples $x \sim P$ and synthesized samples $\hat{x} \sim \hat{P}$, where \hat{P} is the probability distribution induced by sampling $z \sim Q$ and computing $\hat{x} = g(z)$. The formalization (??) highlights the underlying existence of an arbitrary binary classification loss function ℓ . (Nowozin et al., 2016) explores the relationship between the shape of this loss function and the f -divergence minimized by the generator function g .

Unfortunately, GANs do not allow the tractable evaluation of their log-likelihood with respect to some data. Therefore, we will employ a two-sample test to evaluate the quality of the samples $\hat{x} = g(z)$ synthesized by the generator. In simple terms, evaluating a GAN in this manner amounts to withhold some original data from the training process, and then use it to perform a two sample test against the same amount of synthesized data. When the two-sample test is a binary classifier (as discussed in Section 3), this procedure can be seen as simply *training a fresh discriminator on a fresh set of data*.

We evaluate the usefulness of two-sample tests to perform model selection in generative adversarial networks. To this end, we train a number of Deep Convolutional Generative Adversarial Networks, or DCGANs (Radford et al.,

2016) on the LSUN (Yu et al., 2015, bedroom class) and the Labeled Faces in the Wild (LFW) dataset (Huang et al., 2007). We reused the torch code of Radford et al. (2016) to train a collection of DCGANs for $\{1, 10, 50, 100, 200\}$ epochs, where the generator and discriminator networks were convolutional neural networks (LeCun et al., 1998) with $\{32, 64, 96\}$ filters per layer. We then evaluated the quality of each DCGAN by using the MMD, C2ST-NN, and C2ST-KNN tests.

Our first experiments in this dataset revealed an interesting result. When performing two-sample tests directly on pixels, all tests would obtain near-perfect test accuracy when distinguishing between real and synthesized (fake) samples. Such near-perfect accuracy happened consistently across DCGANs, regardless of the visual quality of their samples. This is because, albeit visually indistinguishable, the fake samples contain a variety of pixel-level artifacts which are sufficient for the tests to consistently differentiate between real and fake. In a second series of experiments, we featurized all images (both real and fake) using a deep convolutional residual network (He et al., 2015) pre-trained on ImageNET, a dataset of natural images (Russakovsky et al., 2015). In particular, we use the `resnet-34` model from Gross & Wilber (2016). Reusing a model pre-trained on natural images ensures that the test will distinguish between real and fake samples based only on natural image statistics, such as gabor filters, edge detectors, and so forth. Such a strategy is similar in spirit to perceptual losses (Johnson et al., 2016) and the “inception score” from Salimans et al. (2016). The intuition here is that, in order to evaluate how natural do the images synthesized by a DCGAN look, one must employ a “natural discriminator” for this task.

Tables 1 and 2, included in the Appendix, show samples for each DCGAN, together with the two-sample test statistics provided by MMD, C2ST-NN, and C2ST-KNN. Although it is challenging to provide with an absolutely objective evaluation of our results, we believe that the two-sample tests provide rank sensibly the trained DCGAN models, and that this ranking can be used for efficient early stopping and model selection.

5 Conditional GANs for causal discovery

To conclude our exposition, we propose the novel use of conditional GANs (Mirza & Osindero, 2014) and classifier two-sample tests to perform causal discovery.

In causal discovery, we study the causal structure underlying a set of d random variables X_1, \dots, X_d . In particular, we assume that the random variables X_1, \dots, X_d are related by means of a causal structure, described by a Structural Equation Model, or SEM (Pearl, 2009). More specifically, we assume that the random variables X_i take values as described by the structural equations

$$X_i = f_i(\text{Pa}(X_i, \mathcal{G}), N_i),$$

for all $i = 1, \dots, d$. In the previous, \mathcal{G} is a Directed Acyclic Graph (DAG) with vertices associated to each of the random variables X_1, \dots, X_d . Also in the same equation, $\text{Pa}(X_i, \mathcal{G})$ denotes the set of parents of the random variable X_i in the graph \mathcal{G} , and N_i is an independent noise random variable that follows the probability distribution $P(N_i)$.

Now, let us assume that the graph \mathcal{G} captures the causal structure describing the set of random variables X_1, \dots, X_d . Then, the edges $X_i \rightarrow X_j$ gain the

meaning “ X_i causes X_j ”. The causal interpretation of SEMs becomes clear when stated in terms of interventions: if X_j is a parent of X_i in \mathcal{G} , then intervening on the value of X_j will have an effect on the value of X_i , and such effect will be described correctly by the graph and the equations in our SEM.

The goal of causal discovery is to infer the causal graph \mathcal{G} given samples from the joint probability distribution $P(X_1, \dots, X_d)$. For the sake of simplicity, we here focus on the discovery of causal relations between two random variables, X and Y . That is, given samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \sim P^n(X, Y)$, we are interested in devising algorithms able to conclude whether “ X causes Y ”, or “ Y causes X ”. This problem is known as cause-effect discovery (Mooij et al., 2016). In the case where $X \rightarrow Y$, we can write the cause-effect relationship as:

$$\begin{aligned} x &\sim P(X), \\ n &\sim P(N), \\ y &\leftarrow f(x, n). \end{aligned} \tag{5}$$

The current state-of-the-art in the cause-effect discovery is the family of Additive Noise Models, or ANM (Mooij et al., 2016). These methods assume that the structural equation (5) can be written as $y \leftarrow f(x) + n$, and exploit the independence between cause X and noise N to infer the causal relationship from data.

However, the additive noise model assumption can be limiting in some cases. Because of this reason, we propose to use conditional generative adversarial networks to address the problem of cause-effect discovery. The use of conditional GANs is motivated by their shockingly resemblance to the structural equation model (5). In particular, conditional GANs bypass the additive noise assumption by allowing arbitrary interactions $f(X, N)$ between the cause variable X and the noise variable N . Moreover, GANs respect the independence between cause, noise, and mechanism by definition, since the noise is sampled from a simple distribution a priori.

Following the formalizations from Equation 6, training a conditional GAN from X to Y is to minimize, in turns, the following two objectives:

$$\begin{aligned} L_d(d) &= \mathbb{E}_x \ell(d(x, y), 1) + \mathbb{E}_z \ell(d(x, g(x, z)), 0), \\ L_g(g) &= \mathbb{E}_z \ell(d(x, g(x, z)), 1). \end{aligned} \tag{6}$$

Therefore, our recipe for cause-effect discovery using conditional GANs is to:

1. Learn a conditional GAN from X to Y and generate $\mathcal{D}_{X \rightarrow Y} = \{(x_i, g_y(x_i, z_i))\}_{i=1}^n$.
2. Learn a conditional GAN from Y to X and generate $\mathcal{D}_{X \leftarrow Y} = \{(g_x(y_i, z_i), y_i)\}_{i=1}^n$.
3. Denote by $\hat{t}_{X \rightarrow Y}$ the two-sample statistic on \mathcal{D} versus $\mathcal{D}_{X \rightarrow Y}$.
4. Denote by $\hat{t}_{X \leftarrow Y}$ the two-sample statistic on \mathcal{D} versus $\mathcal{D}_{X \leftarrow Y}$.
5. If $\hat{t}_{X \rightarrow Y} < \hat{t}_{X \leftarrow Y}$, return “ X causes Y ”.
6. Else if $\hat{t}_{X \rightarrow Y} > \hat{t}_{X \leftarrow Y}$, return “ Y causes X ”.
7. Else, return “test inconclusive”.

Using a conditional GAN together with the C2ST-KNN test for cause-effect discovery yields 73% accuracy on the 99 scalar Tübingen cause-effect pairs dataset, version August 2016 (Mooij et al., 2016). Running 100 conditional GANs from different random initializations and preferring the top 1% for each cause-effect pair increases the performance to 82% classification accuracy. This result highlights the promise of GANs for causal discovery. Evaluating the same ensembles using the C2ST-NN test yielded 73%, and 65% when using the MMD test. Overall, our results are a significant improvement with respect to ANM: our implementation yields 66% accuracy. Learning-based methods, which require constructing a large dataset of cause-effect pairs, obtain near 79% accuracy (Lopez-Paz et al., 2015).

References

- Y. Bengio, L. Yao, and K. Cho. Bounding the test log-likelihood of generative models. *arXiv preprint arXiv:1311.6184*, 2013.
- G. E. P. Box. Sampling and bayes' inference in scientific modelling and robustness. *Journal of the Royal Statistical Society. Series A (General)*, 1980.
- K. G. Dziugaite, D. M. Roy, and Z. Ghahramani. Training generative neural networks via Maximum Mean Discrepancy optimization. *ArXiv*, 2015.
- J. H. Friedman. On multivariate goodness of fit and two sample testing. *eConf*, 2003.
- I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NIPS*, 2014.
- A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *JMLR*, 2012a.
- A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. *NIPS*, 2012b.
- S. Gross and M. Wilber. Training and investigating residual nets, 2016. URL <http://torch.ch/blog/2016/02/04/resnets.html>.
- M. U. Gutmann and A. Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *JMLR*, 2012.
- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2015.
- G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst, 2007.
- D. H. Johnson. The insignificance of statistical significance testing. *The journal of wildlife management*, 1999.
- J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *ArXiv*, 2016.

- D. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
- A. N. Kolmogorov. Sulla determinazione empirica di una legge di distribuzione, 1933.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.
- E. L. Lehmann and J. P. Romano. *Testing statistical hypotheses*. Springer Science & Business Media, 2006.
- J. R. Lloyd and Z. Ghahramani. Statistical model criticism using kernel two sample tests. In *Advances in Neural Information Processing Systems*, 2015.
- D. Lopez-Paz, K. Muandet, B. Schölkopf, and I. Tolstikhin. Towards a learning theory of cause-effect inference. *ICML*, 2015.
- H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 1947.
- T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 2013.
- M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- S. Mohamed and B. Lakshminarayanan. Learning in Implicit Generative Models. *ArXiv*, 2016.
- J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *JMLR*, 2016.
- S. Nowozin, B. Cseke, and R. Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *arXiv*, 2016.
- E. Olivetti, S. Greiner, and P. Avesani. Induction in neuroscience with classification: issues and solutions. In *Machine Learning and Interpretation in Neuroimaging*. 2012.
- J. Pearl. *Causality*. Cambridge University Press, 2009.
- F. Pereira, T. Mitchell, and M. Botvinick. Machine learning classifiers and fMRI: a tutorial overview. *Neuroimage*, 2009.
- A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2016.
- A. Ramdas, S. J. Reddi, B. Poczos, A. Singh, and L. Wasserman. Adaptivity and Computation-Statistics Tradeoffs for Kernel and Distance based High Dimensional Two Sample Testing. *ArXiv*, 2015.

- A. Ramdas, A. Singh, and L. Wasserman. Classification accuracy as a proxy for two sample testing. *ArXiv*, 2016.
- M. D. Reid and R. C. Williamson. Information, divergence and risk for binary experiments. *JMLR*, 2011.
- O. Russakovsky, J. Deng, H Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 2015.
- T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. *ArXiv*, 2016.
- N. V. Smirnov. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscow*, 1939.
- Student. The probable error of a mean. *Biometrika*, 1908.
- L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. *ArXiv*, 2015.
- F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1945.
- F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *ArXiv*, 2015.

A Results of image experiments

gf	df	ep	random sample	MMD	KNN	NN
-	-	-		-	-	-
32	32	1		0.154	0.994	1.000
32	32	10		0.024	0.831	0.996
32	32	50		0.026	0.758	0.983
32	32	100		0.014	0.797	0.974
32	32	200		0.012	0.798	0.964
32	64	1		0.330	0.984	1.000
32	64	10		0.035	0.897	0.997
32	64	50		0.020	0.804	0.989
32	64	100		0.032	0.936	0.998
32	64	200		0.048	0.962	1.000
32	96	1		0.915	0.997	1.000
32	96	10		0.927	0.991	1.000
32	96	50		0.924	0.991	1.000
32	96	100		0.928	0.991	1.000
32	96	200		0.928	0.991	1.000
64	32	1		0.389	0.987	1.000
64	32	10		0.023	0.842	0.979
64	32	50		0.018	0.788	0.977
64	32	100		0.017	0.753	0.959
64	32	200		0.018	0.736	0.963
64	64	1		0.313	0.964	1.000
64	64	10		0.021	0.825	0.988
64	64	50		0.014	0.864	0.978
64	64	100		0.019	0.685	0.978
64	64	200		0.021	0.775	0.980
64	96	1		0.891	0.996	1.000
64	96	10		0.158	0.830	0.999
64	96	50		0.015	0.801	0.980
64	96	100		0.016	0.866	0.976
64	96	200		0.020	0.755	0.983
96	32	1		0.356	0.986	1.000
96	32	10		0.022	0.770	0.991
96	32	50		0.024	0.748	0.949
96	32	100		0.022	0.745	0.965
96	32	200		0.024	0.689	0.981
96	64	1		0.287	0.978	1.000
96	64	10		0.012	0.825	0.966
96	64	50		0.017	0.812	0.962
96	64	100		0.019	0.670	0.983
96	64	200		0.020	0.711	0.972
96	96	1		0.672	0.999	1.000
96	96	10		0.671	0.999	1.000
96	96	50		0.829	0.999	1.000
96	96	100		0.668	0.999	1.000
96	96	200		0.849	0.999	1.000

Table 1: GAN evaluation experiments on the LSUN dataset.





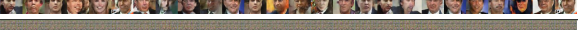
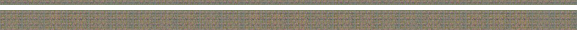


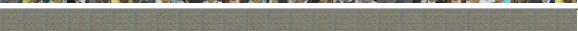

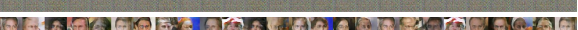




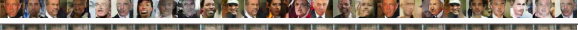



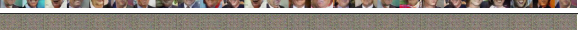










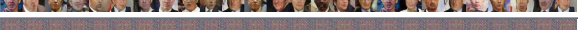

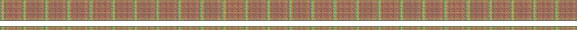



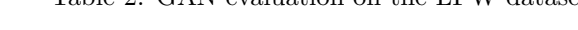









gf	df	ep	random sample	MMD	KNN	NN
-	-	-		-	-	-
32	32	1		0.806	1.000	1.000
32	32	10		0.152	0.940	1.000
32	32	50		0.042	0.788	0.993
32	32	100		0.029	0.808	0.982
32	32	200		0.022	0.776	0.970
32	64	1		0.994	1.000	1.000
32	64	10		0.989	1.000	1.000
32	64	50		0.050	0.808	0.985
32	64	100		0.036	0.766	0.972
32	64	200		0.015	0.817	0.987
32	96	1		0.995	1.000	1.000
32	96	10		0.992	1.000	1.000
32	96	50		0.995	1.000	1.000
32	96	100		0.053	0.778	0.987
64	96	200		0.037	0.779	0.995
64	32	1		1.041	1.000	1.000
64	32	10		0.086	0.971	1.000
64	32	50		0.043	0.756	0.988
64	32	100		0.018	0.746	0.973
64	32	200		0.025	0.757	0.972
64	64	1		0.836	1.000	1.000
64	64	10		0.103	0.910	0.998
64	64	50		0.018	0.712	0.973
64	64	100		0.020	0.784	0.950
64	64	200		0.022	0.719	0.974
64	96	1		1.003	1.000	1.000
64	96	10		1.015	1.000	1.000
64	96	50		1.002	1.000	1.000
64	96	100		1.063	1.000	1.000
64	96	200		1.061	1.000	1.000
96	32	1		1.022	1.000	1.000
96	32	10		0.222	0.978	1.000
96	32	50		0.026	0.734	0.965
96	32	100		0.016	0.735	0.964
96	32	200		0.021	0.780	0.973
96	64	1		0.715	1.000	1.000
96	64	10		0.042	0.904	0.999
96	64	50		0.024	0.697	0.971
96	64	100		0.028	0.744	0.983
96	64	200		0.020	0.697	0.976
96	96	1		0.969	1.000	1.000
96	96	10		0.920	1.000	1.000
96	96	50		0.926	1.000	1.000
96	96	100		0.920	1.000	1.000
96	96	200		0.923	1.000	1.000

Table 2: GAN evaluation on the LFW dataset.