
Generative Adversarial Nets from a Density Ratio Estimation Perspective

Masatoshi Uehara

Univeristy of Tokyo

uehara-masatoshi136@g.ecc.u-tokyo.ac.jp

Issei Sato

The Univeristy of Tokyo

sato@k.u-tokyo.ac.jp

Masahiro Suzuki

The Univeristy of Tokyo

masa@weblab.t.u-tokyo.ac.jp

Kotaro Nakayama

The Univeristy of Tokyo

nakayama@weblab.t.u-tokyo.ac.jp

Yutaka Matsuo

The Univeristy of Tokyo

matsuo@weblab.t.u-tokyo.ac.jp

Abstract

Generative adversarial networks (GANs) are successful deep generative models. GANs are based on a two-player minimax game. However, the objective function derived in the original motivation is changed to obtain stronger gradients when learning the generator. We propose a novel algorithm that repeats the density ratio estimation and f-divergence minimization. Our algorithm offers a new perspective toward the understanding of GANs and is able to make use of multiple viewpoints obtained in the research of density ratio estimation, e.g. what divergence is stable and relative density ratio is useful.

1 Introduction

There have been many recent studies about deep generative models. Generative adversarial networks (GAN) [5] is the variant of these models that has attracted the most attention. Generating vivid, realistic images from uniform distribution is possible [18, 3]. GANs are formulated as a two-player minimax game. However, the objective function derived in the original motivation is changed to obtain stronger gradients when learning the generator. Despite the fact that GANs have been applied in various studies, few have attempted to reveal their mechanism [6, 9].

Recently, f-GAN, which minimizes the variational estimate of f-divergence, has been proposed [17]. The original GAN is a special case of f-GAN.

In this study, we propose a novel algorithm inspired by GAN from the perspective of density ratio estimation based on the Bregman divergence, which we call b-GAN. It iterates the density ratio estimation and f-divergence minimization based on the obtained density ratio. In this study, we present two contributions.

1. We derive a novel algorithm that employs the well-studied results regarding density ratio estimation [10, 21].
2. In the original GANs, the value function derived from the two-player minimax game does not match the objective function that is actually used for learning generative model. In our

Table 1: Relation among GAN, f-GAN and b-GAN.

Name	D-step (updating θ_D)	G-step (updating θ_G)
GAN	estimate $\frac{p}{p+q}$	adversarial update
f-GAN	estimate $f'(\frac{p}{q})$ when $f = x \log x - (x+1) \log(x+1)$, it is GAN	minimize a part of variational estimate of f-divergence
b-GAN (this work)	estimate $\frac{p}{q} = r(x)$ dual relation with f-GAN	$\min_{\theta} E_{x \sim q(x; \theta)}[f(r(x))]$ minimize f-divergence directly

algorithm, the objective function derived from the original motivation is not changed for learning generative model.

The remaining sections of this paper are organized as follows. Section 2 describes related work. Section 3 introduces our novel algorithm and analyze that algorithm in detail. Section 4 explains our algorithm for specific cases. Section 5 reports the result of experiments. Section 6 sums up our statement and discusses future directions.

2 Related work

In this study, we denote an input space as X and a hidden space as Z . Let $p(x)$ be the distribution of training data over X and $q(x)$ be the generated distribution over X .

GANs [5] were developed based on a game theory scenario, where two model – a generator network and a discriminator network – are simultaneously trained. The generator network $G_{\theta_G}(z)$ produces samples with a probability density function of $q(x; \theta_G)$. The discriminator network $T_{\theta_D}(x)$ attempts to distinguish the samples from the training samples and that from the generator. GANs are described as a zero-sum game, where a function $v(G, T)$ determines the payoff of the discriminator and function $-v(G, T)$ determines the payoff of the generator. The discriminator $T_{\theta_D}(x)$ and generator $G_{\theta_G}(z)$ play the following two-player minimax game $\min_{\theta_G} \max_{\theta_D} v(G, T)$, where $v(G, T)$ is given by

$$E_{x \sim p(x)}[\log T_{\theta_D}(x)] + E_{x \sim q(x; \theta_G)}[\log(1 - T_{\theta_D}(x))]. \quad (1)$$

The discriminator and generator are iteratively trained by turns. For fixed G, the optimal $T(x)$ is $\frac{p(x)}{p(x)+q(x)}$. This suggests that training the discriminator can be formulated as a density ratio estimation. The generator is trained to minimize $v(G, T)$ adversarially. In fact, maximizing $E_{x \sim q(x; \theta_G)}[\log T_{\theta_D}(x)]$ is preferred to minimizing $E_{x \sim q(x; \theta_G)}[\log(1 - T_{\theta_D}(x))]$. Although this does not match the theoretical motivation, this heuristic is the key to successful learning. We analyze this heuristic in Sec 3.5.

f-GAN [17] generalizes the GAN concept. Function $v(G, T)$ of f-GAN is given by

$$E_{x \sim p(x)}[T_{\theta_D}(x)] - E_{x \sim q(x; \theta_G)}[f^*(T_{\theta_D}(x))], \quad (2)$$

where f^* is a Fenchel conjugate of f [16]. In Eq.2, $v(G, T)$ comes from,

$$\int q(x) f\left(\frac{p(x)}{q(x)}\right) dx = \sup_T E_{x \sim p}[T(x)] - E_{x \sim q}[f^*(T(x))]. \quad (3)$$

Following GANs, θ_D is trained to maximize Eq.2 to estimate the f-divergence. In contrast, θ_G is trained to adversarially minimize Eq.2 to minimize f-divergence estimate. However, as in GANs, maximizing $E_{x \sim q}[T(x)]$ is used instead of minimizing $E_{x \sim q}[-f^*(T(x))]$, where the latter optimization is theoretically valid in their formulation, but they used the former heuristically. f-GAN also formulates the training discriminator as a density ratio estimation, same as that in GANs. For fixed G, the optimal $T(x)$ is $f'(\frac{p}{q})$, where f' denotes the first order derivative of f . When $f(x)$ is $x \log x - (x+1) \log(1+x)$, f-GANs are equivalent to GANs. Table 1 summarizes GAN and f-GAN.

3 Method

As described in Section 2, training the discriminators in the D-step of GANs and f-GANs is regarded as density ratio estimation. In this section, we further extend this idea. We first review the density ratio estimation method based on the Bregman divergence. We then explain and analyze a novel algorithm, b-GAN.

3.1 f-divergence

The f-divergence measures the difference between two probability distributions p and q and is defined by

$$D_f(p||q) = \int q(x) f\left(\frac{p(x)}{q(x)}\right) dx = \int q(x) f(r(x)) dx, \quad (4)$$

where $f(x)$ is a convex function satisfying $f(1) = 0$. Noted in the space of positive measures (not satisfying normalized conditions), f-divergence must meet $f'(1) = 0$ because of its invariance [2].

3.2 Density ratio matching under the Bregman divergence

There have been many studies on direct density ratio estimation, where a density ratio model is fitted to a true density ratio model under the Bregman divergence [21]. We briefly review this method.

Suppose that there are two distributions $p(x)$ and $q(x)$. Our aim is to directly estimate the true density ratio $r(x) = \frac{p(x)}{q(x)}$ without independently estimating $p(x)$ and $q(x)$. Let $r_\theta(x)$ be a density ratio model. The integration of the Bregman divergence $d_f[r(x)||r_\theta(x)]$ between the density ratio model and true density ratio with respect to measure $q(x)dx$ is

$$\begin{aligned} BD_f(r||r_\theta) &= \int d_f[r(x)||r_\theta(x)] q(x) dx \\ &= \int (f(r(x)) - f(r_\theta(x)) - f'(r_\theta(x))(r(x) - r_\theta(x))) q(x) dx. \end{aligned} \quad (5)$$

We define the terms related to r_θ in $BD_f(r||r_\theta)$ as

$$BR_f(r_\theta) = \int (f'(r_\theta(x))r_\theta(x) - f(r_\theta(x))) q(x) dx - \int f'(r_\theta(x)) p(x) dx \quad (6)$$

$$= \int f'(r_\theta(x)) (r_\theta(x)q(x) - p(x)) dx - D_f(qr_\theta||q). \quad (7)$$

Thus, estimating the density ratio problem turns out to be the minimization of Eq.7 with respect to θ .

3.3 Motivation

In this section, we introduce important props needed to derive b-GAN. The following prop suggests that the supremum of the negative of Eq.6 is equal to the f-divergence between $p(x)$ and $q(x)$.

Prop 3.1. *The following equation holds.*

$$E_q\left[f\left(\frac{p(x)}{q(x)}\right)\right] = \sup_{r_\theta} E_{x \sim p}[f'(r_\theta(x))] - E_{x \sim q}[(f'(r_\theta(x))r_\theta(x) - f(r_\theta(x)))]. \quad (8)$$

The right side of Eq.8 reaches the supremum when $r_\theta(x) = r(x)$ is satisfied.

Proof. The following equation holds, from Eq. 6:

$$E_{x \sim p}[f'(r_\theta(x))] - E_{x \sim q}[(f'(r_\theta(x))r_\theta(x) - f(r_\theta(x)))] = -BD_f(r||r_\theta) + E_q\left[f\left(\frac{p(x)}{q(x)}\right)\right]. \quad (9)$$

Using $BD_f(r||r_\theta) \geq 0$ yields Eq. 8. We have $BD_f(r||r_\theta) = 0$ when r is equal to r_θ . Hence, the equality holds if and only if r is equal to r_θ . \square

It has been shown that the supremum of negative of Eq. 6 is equivalent to the supremum of Eq. 2. Interestingly, the negative Eq. 6 has a dual relation with the objective function of f-GAN, i.e., Eq. 2.

Prop 3.2. *Introducing dual coordinates $T_{\theta_D} = f'(r_{\theta})[2]$ yields the right side of Eq. 6 from Eq. 2.*

Proof. The steps to prove this are as follows.

$$\begin{aligned} \text{r.h.s of Eq.2} &= E_{x \sim p(x)}[f'(r_{\theta}(x))] - E_{x \sim q(x)}[f^*(f'(r_{\theta}(x)))] \\ &= E_{x \sim p(x)}[f'(r_{\theta}(x))] - E_{x \sim q(x)}[f(r_{\theta}(x))r_{\theta}(x) - f(r_{\theta}(x))] \\ &= E_{x \sim p(x)}[f'(r_{\theta}(x))] - E_{x \sim q(x)}[(f'(r_{\theta}(x))r_{\theta}(x) - f(r_{\theta}(x)))], \end{aligned} \quad (10)$$

In the derivation, we used the following equation $f^*(f'(r_{\theta}(x))) = f(r_{\theta}(x))r_{\theta}(x) - f(r_{\theta}(x))$. \square

Prop 3.2 shows the D-step of f-GAN can be regarded as the density ratio estimation because Eq. 6 is an equation of the density ratio estimation and Eq. 2 is a value function of f-GAN. As for the G-step, minimizing the variational estimate of the f-divergence indicates that the value of Eq. 2 decreases, i.e., the value of Eq. 7 increases. Hence, updating G means minimizing a part of the variational estimate of the f-divergence, $E_{x \sim q}[f(r_{\theta}(x)) - (f'(r_{\theta}(x))r_{\theta}(x))]$. However, this estimation ignores the term $E_{x \sim p}[f'(r_{\theta}(x))]$. Using importance sampling, the ignored term can be estimated as $E_{x \sim q}[r_{\theta}f'(r_{\theta})]$. Combining $E_{x \sim q}[f(r_{\theta}(x)) - (f'(r_{\theta}(x))r_{\theta}(x))]$ and $E_{x \sim q}[r_{\theta}f'(r_{\theta})]$, we get $E_{x \sim q}[f(r_{\theta})]$, i.e., the f-divergence. Rather than minimizing $E_{x \sim q}[f(r_{\theta}(x)) - (f'(r_{\theta}(x))r_{\theta}(x))]$, directly minimizing f-divergence seems to be more natural. We present the method that uses this more natural update as the G-step in the next section.

3.4 b-GAN

Our objective is to minimize the f-divergence between the distribution of the training data $p(x)$ and generated distribution $q(x)$. We introduce two functions constructed using neural networks: $r_{\theta_D}(x) : X \rightarrow \mathcal{R}$, which is parameterized by θ_D , and $G_{\theta_G}(z) : Z \rightarrow X$, which is parameterized by θ_G . Measure $q(x; \theta_G)dx$ is a probability measure induced from the uniform distribution by $G_{\theta_G}(z)$. In this case, $r_{\theta_D}(x)$ is regarded as a density-ratio estimation network and $G_{\theta_G}(z)$ is regarded of as a generator network for minimizing the f-divergence between $p(x)$ and $q(x)$.

Motivated by Section 3.3, we construct a b-GAN using the following two steps.

1. Update θ_D to estimate the density ratio between $p(x)$ and $q(x; \theta_G)$. To achieve this, we minimize Eq. 6 with respect to $r_{\theta}(x)$. The density ratio model $r_{\theta}(x)$ in Eq. 6 can be considered as $r_{\theta_D}(x)$ in this step.
2. Update θ_G to minimize the f-divergence $D_f(p||q)$ between $p(x)$ and $q(x; \theta_G)$ using the obtained density-ratio. We are able to suppose that $q(x; \theta_G)r_{\theta}(x)$ is close to $p(x)$. Instead of $D_f(p||q)$, we update θ_G to minimize $D_f(qr_{\theta}||q)$ by assuming the empirical approximation.

We denote the step for updating θ_D as D-step and the step for updating θ_G as G-step. The algorithm of b-GAN is summarized in Algorithm 1, where B is a batch size. In this study, a single-step gradient method[5, 17] is adopted.

Algorithm 1: b-GAN

for number of training iterations **do**

sample $\hat{X} = \{x_1, \dots, x_B\}$ from $p(x)$ and $\hat{Z} = \{z_1, \dots, z_B\}$ from an uniform distribution.

D-step: Update θ_D :

$$\theta_D^{t+1} = \theta_D^t - \nabla_{\theta_D} \left(\frac{1}{B} \sum_{i=1}^B f'(r_{\theta_D}(G(z_i)))r_{\theta_D}(G(z_i)) - f(r_{\theta_D}(G(z_i))) - f'(r_{\theta_D}(x_i)) \right).$$

G-step: Update θ_G :

$$\theta_G^{t+1} = \theta_G^t - \nabla_{\theta_G} \left(\frac{1}{B} \sum_{i=1}^B f(r(G_{\theta_G}(z_i))) \right).$$

end for

In the D-step, our algorithm estimates p/q towards any divergence; hence, it is slightly different from the D-step of f-GAN because the estimated values, i.e., $f'(p/q)$, are dependent on the divergences. We also introduce an f-GAN-like update as follows. As mentioned in Section 3, we have two options in the G step.

1. D-step: minimize $E_{x \sim p(x)}[-f'(r_{\theta_D}(x))] + E_{x \sim q(x)}[f'(r_{\theta_D}(x))r_{\theta_D}(x) - f(r_{\theta_D}(x))]$ w.r.t θ_D .
2. G-step: minimize $E_{x \sim q(x; \theta_G)}[-f'(r(x))]$ or $E_{x \sim q(x; \theta_G)}[-f'(r(x))r(x) + f(r(x))]$ w.r.t θ_G .

3.5 Analysis

Following Goodfellow et al. [2014], we explain the validity of the G-step and D-step. We then explain the meaning of b-GAN. Finally, we analyze differences between b-GAN and f-GAN.

In the D-step, the density ratio is estimated. The estimator of $r(x)$ is asymptotically consistent under the proper normal conditions [10]. In the G-step, we update the generator as minimizing $D_f(p||q)$ by replacing $p(x)$ with $r_{\theta}(x)q(x)$. We assume that $q(x; \theta_G)$ is equivalent to $p(x)$ in the case of $\theta_G = \theta^*$, $q(x; \theta_G)$ is identifiable, and the $q(x)$ obtained by $r(x)$ in the G-step is optimal. By our assumption, the acquired value in the G-step is $\hat{\theta}$, which minimizes

$$\frac{1}{B} \sum_{i=1}^B f(r(x_i)) = \frac{1}{B} \sum_{i=1}^B f\left(\frac{q(x_i; \theta^*)}{q(x_i; \hat{\theta})}\right). \quad (11)$$

This value is equal to θ^* because $f(r)$ has a minimum value 0 at $r = 1$. Usually, we cannot conduct only a G-step because we do not know the form of $p(x)$ and $q(x)$. In b-GAN, $D_f(p||q)$ can be minimized by estimating density ratio $r(x)$ without estimating densities directly.

In fact, the $r(x)$ obtained at each time is different and not optimal because we adopt a single-step gradient method. Hence, b-GAN dynamically updates the generator to minimize the f-divergence between $p(x)$ and $q(x)$. As mentioned earlier, $f(x)$ must satisfy $f'(1) = 0$ in this case because we cannot guarantee that $r_{\theta}(x)q(x)$ is normalized.

The D-step and G-step work adversarially, just as in GANs. In the D-step, $r_{\theta}(x)$ is updated to fit the ratio between $p(x)$ and $q(x)$. In the G-step, $q(x)$ changes, which means $r_{\theta}(x)$ becomes inaccurate in terms of the density ratio estimator. Next, $r_{\theta}(x)$ is updated in the D-step so that it fits the density ratio of $p(x)$ and new $q(x)$. This learning situation is derived from Eq. 7, which shows that θ_D is updated to increase $D_f(qr_{\theta}||q)$ in the D-step. In contrast, θ_G is updated to decrease $D_f(qr_{\theta}||q)$ in the G-step.

In Section 3.4, we also introduced an f-GAN-like update. Three choices can be considered for the G-step:

$$(1) E_{x \sim q(x; \theta_G)}[f(r(x))], (2) E_{x \sim q(x; \theta_G)}[-f'(r(x))], (3) E_{x \sim q(x; \theta_G)}[-f'(r(x))r(x) + f(r(x))].$$

Note that f is a convex function, $f(1) = 0$, and $f'(1) = 0$. It is noted in [17] that the case (2) works better than case (3) in practice. We also confirm this. The complete reason for this is unclear. However, we can find a partial reasons by differentiating objective functions with respect to r . The derivatives of the objective functions are

$$(1) f'(r), (2) -f''(r), (3) -rf''(r).$$

All signs are negative when $r(x)$ is below 1. Usually, when x is sampled from $q(x)$, $r(x)$ is below 1. Therefore, we speculate that $r(x)$ is below 1 during most of the process of learning when x is sampled from $q(x)$. small in (3) because term $r(x)$ is multiplied. Therefore, the derivative tends to be small in (3). The

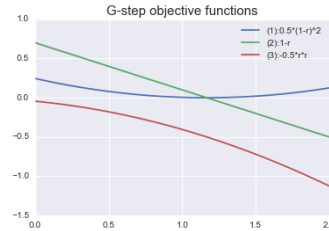


Figure 1: The graph of (1), (2), (3) when f is Pearson divergence

mechanism pulling $r(x)$ to 1 does not work when $r(x)$ is small. Hence, the case of (3) does not work well. A similar argument was proposed by Goodfellow et al., [2014] and Nowozin et al., [2016].

In our experiment cases of (1) and (2) work properly (see Section 5 for details). The reason case (2) works is that function $-f'(r)$ behaves like f-divergence, i.e., $-f'(1) = 0$. However, we cannot guarantee that $-f'(r)$ satisfies the conditions of f-divergence between positive measures, i.e., $-f'(r)$ is a convex function and $-f''(1) = 0$. If the derivatives in case (2) are negative when $r(x)$ is above 1, there is a possibility that the mechanism pulling $r(x)$ to 1 does not occur. In contrast, in case (1), when $r(x)$ is above 1, the derivatives are positive, therefore, the mechanism pulling $r(x)$ to 1 occurs. This prevents generators from emitting the same points. We can expect the same effects as "minibatch discrimination" [19].

4 Explanation of the algorithms for specific cases

We adopt α -divergence in the space of positive measure as the f-divergence [2]. In this case, $f(r)$ in Eq. 4 is

$$f_\alpha(r) = \begin{cases} \frac{4}{1-\alpha^2}(1 - r^{\frac{1+\alpha}{2}}) + \frac{2}{1-\alpha}(r - 1) & (\alpha \neq \pm 1) \\ r \log r - r + 1 & (\alpha = 1) \\ -\log r + r - 1 & (\alpha = -1) \end{cases} \quad (12)$$

4.1 Kullback Leibler(KL) divergence ($\alpha = 1$)

In this case, the form of $f(r)$ is $r \log r - r + 1$.

- D-step: minimize $E_{x \sim q(x)}[r_{\theta_D}(x)] - E_{x \sim p(x)}[\log r_{\theta_D}(x)]$ w.r.t. θ_D .
- G-step: minimize $E_{x \sim q(x; \theta_G)}[r(x) \log r(x) - r(x) + 1]$ w.r.t. θ_G .

Density ratio estimation via the KL divergence corresponds to the KLEIP (Kullback-Leibler Importance Estimation Procedure) [21]. In the G-step of f-GAN-like update, the objective function is $E_{x \sim q(x; \theta_G)}[-\log r(x)]$ or $E_{x \sim q(x; \theta_G)}[-r(x)]$.

4.2 Pearson divergence ($\alpha = 3$)

In this case, the form of $f(r)$ is $\frac{(r-1)^2}{2}$.

- D-step: minimize $E_{x \sim q(x)}[\frac{1}{2}r_{\theta_D}(x)^2] - E_{x \sim p(x)}[1 - r_{\theta_D}(x)]$ w.r.t. θ_D .
- G-step: minimize $E_{x \sim q(x; \theta_G)}[\frac{1}{2}(r(x) - 1)^2]$ w.r.t. θ_G .

Density ratio estimation via the Pearson divergence corresponds to the LSIF (Least-Squares Importance Fitting)[23]. It is more robust than under KL divergence [23]. This is because Pearson divergence does not include the log term. Hence the algorithm using Pearson divergence should be more stable. In the G-step of f-GAN-like update, the objective function is $E_{x \sim q(x; \theta_G)}[1 - r(x)]$ or $E_{x \sim q(x; \theta_G)}[-0.5r(x)^2]$.

4.3 Reversed KL divergence ($\alpha = -1$)

In this case, the form of $f(r)$ is $-\log(r) + r - 1$.

- D-step: minimize $E_{x \sim q(x)}[\log r_{\theta_D}(x)] - E_{x \sim p(x)}[-\frac{1}{r_{\theta_D}(x)}]$ w.r.t. θ_D .
- G-step: minimize $E_{x \sim q(x; \theta_G)}[-\log(r) + r - 1]$ w.r.t. θ_G .

Estimating density ratio with reversed KL divergence seems to be unstable because reversed KL-divergence is mode seeking. However, it is preferable to use reversed KL divergence when generating realistic images [9]. In the G-step of f-GAN-like update, the objective function is $E_{q(x; \theta_G)}[-\frac{1}{r_{\theta_D}(x)}]$ or $E_{q(x; \theta_G)}[\log r_{\theta_D}(x)]$.

4.4 Some heuristics

We describe some heuristic methods that work for our experiments. In the initial learning process, empirical distribution p and generated distribution q are totally different. Therefore, the estimated density ratio $r(x) = \frac{p(x)}{q(x)}$ is enormous when x is taken from p and tiny when x is taken from q . It seems that the learning does not succeed in this case. In fact, in our settings, when the final activation function of $r_{\theta_D}(x)$ is taken from functions in the range $(0, \infty)$, b-GAN does not properly work. Therefore, we use a scaled sigmoid function such as a 2-times sigmoid function.

As mentioned, density ratio $\frac{p(x)}{q(x)}$ is extremely sensitive. To avoid this problem, in the D-step of the KL-divergence, we also conducted experiments wherein we estimated not $\frac{p(x)}{q(x)}$ but $\frac{p}{\alpha p + (1-\alpha)q}$ (where α is small). The same idea is introduced in the covariant shift situation [22]. A similar idea has been also used for GAN learning [19].

5 Experiments

We ascertain that our algorithm works properly by experiments, and have successfully generated natural images. Our algorithm is based on density ratio estimation. Therefore, knowledge regarding the density ratio estimation can be utilized. In the experiment, using the Pearson divergence and estimating the relative density ratio is shown to be useful for stable learning. We also empirically confirm our statement in Section 3.5, i.e., that f-divergence is increased when learning θ_D and decreased when learning θ_G .

5.1 Settings

We have applied our algorithm to the CIFAR-10 data set [12] and Celeb A data set [14] because it is often used in GAN research [19, 5]. The images size is 32×32 pixels. All results in this section is analyzed based on the results of CIFAR-10 data set. See the appendix for results of the other dataset. Our network architecture is almost equivalent to that of existing study [18] (see the appendix for details). Note that unless otherwise noted, the last layer function of $r_{\theta_D}(x)$ is a sigmoid function multiplied by two. We used TensorFlow for automatic differentiation [1]. For stochastic optimization, ADAM was adopted [11].

5.2 Results

Figure 2 shows the density ratio estimate $r_{\theta_D}(x)$ and loss values of the generators. For each divergence, we conducted four experiments with 40,000 epochs, fixing the initial learning rate value (5×10^{-5}) except for reversed KL divergence. These results show that the b-GANs using the Pearson divergence are stable because the learning did not stop. The same results have been reported in the research on density ratio estimation [23]. In contrast, b-GANs using the KL divergence are unstable. In fact, the learning stopped between the 20,000th and 37,000th epoch when the learning rate is not as small. When we use a heuristic method, i.e., estimating the relative density ratio as described in Section 4.4, this problem is solved. For reversed KL divergence, the learning stopped too soon if the initial learning rate value was 5×10^{-5} . If the learning rate was 1×10^{-6} , the learning did not stop. However, it was still unstable.

In Figure 2, the last layer activation function of b-GANs is a twofold sigmoid function. In Figure 3, we use a sigmoid function multiplied by five. The results indicate that the estimated density ratio values approach one. They also confirm that our algorithm works with sigmoid functions at other scales.

Figure 4 shows the estimated f-divergence $D_f(qr_{\theta}||q)$ before the G-step subtracted by $D_f(qr_{\theta}||q)$ after the G-step. Most of the values are above zero, which suggests f-divergence decreases at every G-step iteration. This observation matches our analysis in Section 3.5.

Figure 5 shows samples that were randomly generated by using b-GANs. These results indicate that b-GANs can successfully create natural images. We did not conduct a Parzen window density estimation for the evaluations because of Theis et al., [2016].

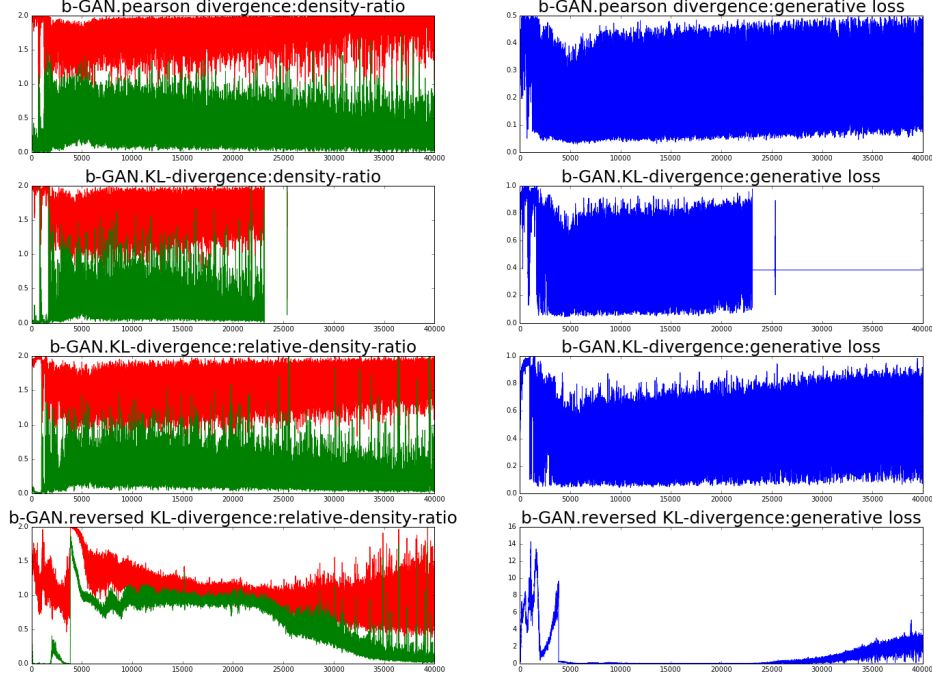


Figure 2: The estimated density ratio values $r_{\theta_D}(x)$ from the training data are showed in red. The estimated density-ratio values $r_{\theta_D}(x)$ from the generated distribution are show in green. Generator losses taken in the G-step are shown in blue. The top, second, and bottom rows show $r_{\theta_D}(x)$ and the losses of b-GAN with the Pearson divergence, KL divergence, modified KL divergence (relative-density-ratio estimation version, $\alpha = 0.2$), and reversed KL divergence, respectively.

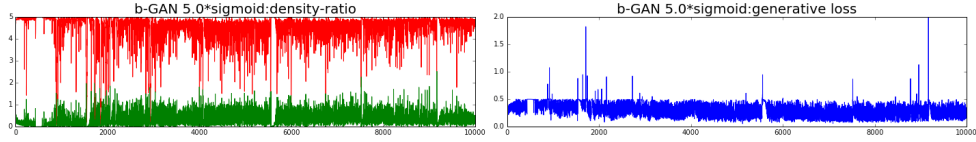


Figure 3: Density ratio value $r_{\theta_D}(x)$ and generator losses of b-GAN when the last output function is a sigmoid function multiplied by 5.

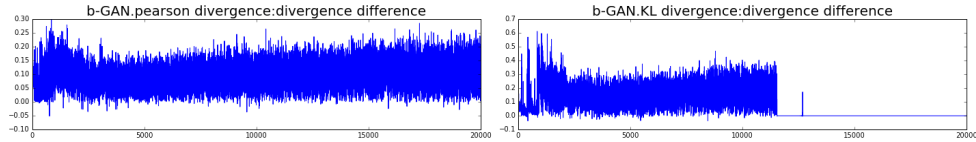


Figure 4: Divergence differences between D-step and G-step (Left) b-GAN with Pearson divergence, (Right) b-GAN with KL divergence.

Note that the learning is successful with an f-GAN-like update when minimizing $E_q[-f'(r)]$. However, the learning f-GAN-like update when minimizing $E_q[f(r) - rf'(r)]$ did not work well for our network architecture and dataset.

6 Conclusions and future work

We proposed a novel algorithm to learn a deep generative model from a density ratio estimation perspective. Our algorithm provides the experimental insights that Pearson divergence and estimating relative density ratio are useful for improving the stability of GAN learning. Other insights regarding

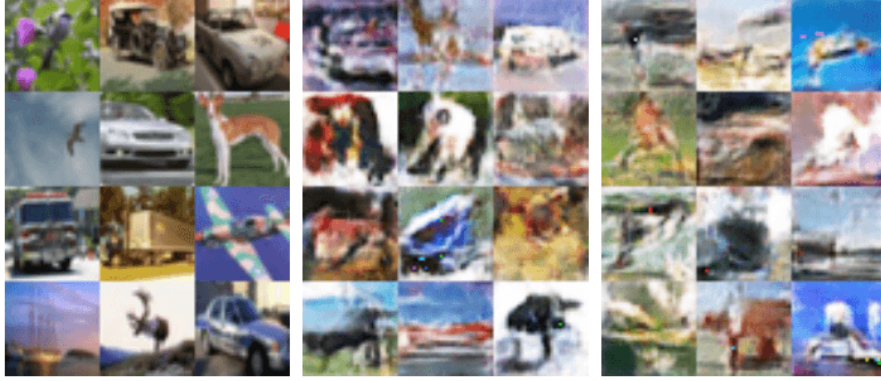


Figure 5: (Left) original images set, (middle) a set of images generated based on the Pearson divergence, and (right) a set of images based on the KL divergence.

density ratio estimation would also be also useful. GANs are sensitive to datasets, the form of network and hyper parameters. Therefore, providing methods for improving GAN learning is meaningful.

In the research regarding density-ratio estimation, the Pearson divergence ($\alpha = 3$) is considered to be robust [15]. We empirically confirmed the same property when learning deep generative models. For generating realistic images, the reversed KL-divergence ($\alpha = -1$) is preferred because it is mode-seeking [9]. However, if α is small, the density ratio estimation becomes inaccurate. Thus, the determination of the optimal divergence is still a persisting problem. In addition, we can think of algorithms such that used divergences are different in G-step and in D-step. Original GANs are described such algorithms as mentioned in Section 3.5. In such a case, choices of divergences are more diverse.

Kernel mean matching is also commonly used in the research regarding density ratio estimation [7]. Methods of learning deep generative models using maximum mean discrepancy has been proposed recently [13, 4]. In such a case, a distance between $p(x)$ and $q(x)$ is minimized by not using density ratios. We are able to apply such kernel methods to f-GAN by using kernel-mean matching as density ratio estimation.

For future work, using the discriminator for inlier-based outlier detection could be considered [8]. Usually, when we use a density ratio estimation for outlier detection, we estimate the divergence between a normal data set, called the inliers, and a test data set containing outliers. In a GAN situation, we artificially generate the test data sets several times and train the discriminator. Almost all researches on GANs till date make use of generators solely. However, if the density ratio estimation is successful, we could use the discriminator for outlier detection based on the density ratio estimation.

Acknowledgement

The authors would like to thank Masanori Misono for technical assistance with the experiments.

A Setup

We describe the network architecture of $r_{\theta_D}(x)$ and $G_{\theta_G}(z)$ used in b-GAN. Here, BN is the batch normalization layer [20].

A.1 $r_{\theta_D}(x)$

$x \rightarrow \text{Conv}(3, 64) \rightarrow \text{lRelu} \rightarrow \text{Conv}(64, 256) \rightarrow \text{BN} \rightarrow \text{lRelu} \rightarrow \text{Conv}(256, 512) \rightarrow \text{BN} \rightarrow \text{lRelu} \rightarrow \text{Reshape}(4 \times 4 \times 512) \rightarrow \text{Linear}(4 \times 4 \times 512, 1) \rightarrow 2 \times \text{Sigmoid}$

A.2 $G_{\theta_G}(z)$

$z \rightarrow \text{Linear}(100, 4 \times 4 \times 512) \rightarrow \text{BN} \rightarrow \text{Relu} \rightarrow \text{Reshape}(4, 4, 512) \rightarrow \text{Conv}(512, 256) \rightarrow \text{BN} \rightarrow \text{Relu} \rightarrow \text{Conv}(256, 64) \rightarrow \text{BN} \rightarrow \text{Relu} \rightarrow \text{Conv}(64, 3) \rightarrow \tanh$

B Celeb A dataset

We have also applied our algorithm to Celeb A dataset. The images are resized and cropped to 64×64 . Figure 6 and 7 show samples that were randomly generated by using b-GANs.

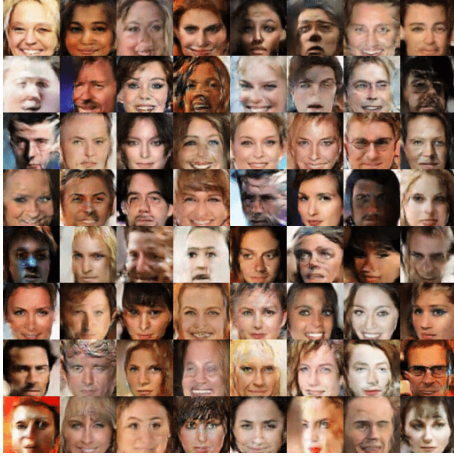


Figure 6: Pearson divergence

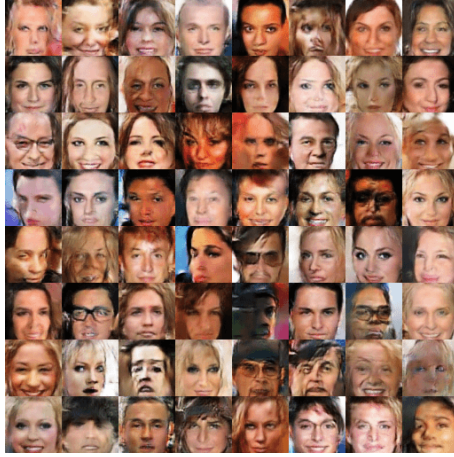


Figure 7: KL divergence

Network architecture is as follows.

B.1 $r_{\theta_D}(x)$

$x \rightarrow \text{Conv}(3, 64) \rightarrow \text{lRelu} \rightarrow \text{Conv}(64, 128) \rightarrow \text{BN} \rightarrow \text{lRelu} \rightarrow \text{Conv}(128, 256) \rightarrow \text{BN} \rightarrow \text{lRelu} \rightarrow \text{Conv}(256, 512) \rightarrow \text{BN} \rightarrow \text{lRelu} \rightarrow \text{Reshape}(4 \times 4 \times 512) \rightarrow \text{Linear}(4 \times 4 \times 512, 1) \rightarrow 2 \times \text{Sigmoid}$

B.2 $G_{\theta_G}(z)$

$z \rightarrow \text{Linear}(64, 4 \times 4 \times 512) \rightarrow \text{BN} \rightarrow \text{Relu} \rightarrow \text{Reshape}(4, 4, 512) \rightarrow \text{Conv}(512, 256) \rightarrow \text{BN} \rightarrow \text{Relu} \rightarrow \text{Conv}(256, 128) \rightarrow \text{BN} \rightarrow \text{Relu} \rightarrow \text{Conv}(128, 64) \rightarrow \text{BN} \rightarrow \text{Relu} \rightarrow \text{Conv}(64, 3) \rightarrow \tanh$

References

- [1] M. Abadi, A. Agarwal, and P Barham. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <http://tensorflow.org/>. Software available from tensorflow.org.
- [2] S. Amari and A. Cichoki. Information geometry of divergence functions. *Bull. Polish. Acad. Sci.*, 58:183–195, 2010.
- [3] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. *ArXiv e-prints*, June 2015.
- [4] Gintare K. Dziugaite, Daniel M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Uncertainty in Artificial Intelligence (UAI)*, 2015.
- [5] I. Goodfellow, M. Pouget-Abadie, J. and Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680. 2014.

- [6] I. J. Goodfellow. On distinguishability criteria for estimating generative models. *ArXiv e-prints*, 2014. URL <https://arxiv.org/pdf/1412.6515v4.pdf>.
- [7] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. *Covariate shift by kernel mean matching*. URL http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/attachments/shift-book-for-LeEtAl-webversion_5376%5b0%5d.pdf.
- [8] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Statistical outlier detection using direct density ratio estimation. *Knowl. Inf. Syst.*, 26 (2):309–336, 2011.
- [9] F. Huszar. How (not) to Train your Generative Model: Scheduled Sampling, Likelihood, Adversary? *ArXiv e-prints*, 2015. URL <http://arxiv.org/pdf/1511.05101v1.pdf>.
- [10] T. Kanamori, T. Suzuki, and M. Sugiyama. Divergence estimation and two-sample homogeneity test under semiparametric density-ratio models. *IEEE Transactions on Information Theory*, 58: 708–720, 2012.
- [11] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2014.
- [12] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.
- [13] Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *International Conference on Machine Learning (ICML)*, 2015.
- [14] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.
- [15] H. Nam and M. Sugiyama. Direct density ratio estimation with convolutional neural networks with application in outlier detection. *IEICE Transactions*, 98:1073–1079, 2015.
- [16] X. Nguyen, M. Wainwright, and M. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 2010.
- [17] S. Nowozin, B. Cseke, and R. Tomioka. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. *ArXiv e-prints*, 2016. URL <https://arxiv.org/pdf/1606.00709v1.pdf>.
- [18] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*, 2015.
- [19] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training GANs. *ArXiv e-prints*, June 2016.
- [20] I. Sergey and S. Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015. URL <https://arxiv.org/pdf/1502.03167v3.pdf>.
- [21] M. Sugiyama, T. Suzuki, and T. Kanamori. Density ratio matching under the bregman divergence: a unified framework of density ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64:1009–1044, 2012.
- [22] M. Sugiyama, M. Yamada, and M.C. du Plessis. Learning under non-stationarity:covariate shift and class-balance change. *WIREs Computational Statistics*, 5:465–477, 2013.
- [23] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama. Relative density-ratio estimation for robust distribution comparison. In *Advances in Neural Information Processing Systems*, pages 594–602. 2011.