Boosted Density Estimation Remastered

Zac Cranko [1 2] **Richard Nock** [2 1 3]

Abstract

There has recently been a steady increase in the number iterative approaches to density estimation. However, an accompanying burst of formal convergence guarantees has not followed; all results pay the price of heavy assumptions which are often unrealistic or hard to check. The *Generative Adversarial Network (GAN)* literature — seemingly orthogonal to the aforementioned pursuit — has had the side effect of a renewed interest in variational divergence minimisation (notably f-GAN). We show how to combine this latter approach and the classical boosting theory in supervised learning to get the first density estimation algorithm that provably achieves geometric convergence under very weak assumptions. We do so by a trick allowing to combine *classifiers* as the sufficient statistics of an exponential family. Our analysis includes an improved variational characterisation of f-GAN.

1. Introduction

In the emerging area of *Generative Adversarial Networks (GAN's)* (Goodfellow et al., 2014) a binary classifier, called a *discriminator*, is used learn a highly efficient sampler for a data distribution P; combining what would traditionally be two steps — first learning the density function from a family of densities, then fine-tuning a sampler — into one. Interest in this field has sparked a series of formal inquiries and generalisations describing GAN's in terms of (among other things) divergence minimisation (Nowozin et al., 2016; Arjovsky et al., 2017). Using a similar framework to Nowozin et al. (2016), Grover & Ermon (2018) make a preliminary analysis of an algorithm that takes a series of iteratively trained discriminators to estimate a density function[1]. The cost here, insofar as we have been able to devise, is that one forgoes learning an efficient sampler (as with a GAN), and must make do with classical sampling techniques to sample from the learned density. We leave the issue of efficient sampling from these density as an open problem, and instead focus on analysing the densities learned with formal convergence guarantees. Previous formal results have established a range of guarantees, from qualitative convergence (Grover & Ermon, 2018), to geometric convergence rates (Tolstikhin et al., 2017), with numerous results in between (See §1.1).

Our starting point is fundamentally different, we learn a density from a sequence of binary classifiers. By using a similar weak learning assumptions in boosting, is shown to be able to fit arbitrarily closely the target density.

With the advent of deep learning, such an approach appears to be very promising, as the formal bounds we obtain yield *geometric convergence* under assumptions arguably much weaker than other similar works in this area

The rest of the paper and our contributions are as follows: in §2, to make explicit the connections between classification, density estimation, and divergence minimisation we re-introduce the variational f-divergence formulation, and in doing so are able to fully explain some of the underspecified components of f-GAN (Nowozin et al., 2016); in §3, we relax a number of the assumptions of Grover & Ermon (2018), and then give both more general, and much stronger bounds for their algorithm; in §4, we apply our algorithm to several toy datasets in order demonstrate convergence and compare directly with Tolstikhin et al. (2017); and finally, a final section §5 concludes.

The appendices that follow in the supplementary material are: §A, we compare our formal results with other related works; §B, a geometric account of the function class in the variational form of an f-divergence; §C, a further relaxation of the weak learning assumptions to some that could actually be estimated experimentally and a proof that the boosting rates are slightly worse but of essentially the same order; §D, proofs for the main formal results from the paper; and finally, §E, technical details for the settings of our experiments.

[1]The Australian National University [2]Data61 [3]The University of Sydney. Correspondence to: Zac Cranko <zac.cranko@anu.edu.au>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

[1] Grover & Ermon (2018) call this procedure "multiplicative discriminative boosting".

Table 1: Summary of related works and results in terms of (i) the components aggregated (”updates”) in the final density (this density can be implicit), (ii) the rate to a given KL/JS value and (iii) the assumption(s).

Approach	Updates	Rate	Assumption
Dai et al., 2016	kernel density estimate / particles	$\Omega(\mathrm{KL}(P, Q_0))$	smoothness, Lipschitz, measure concentration, etc.
Tolstikhin et al., 2017	density	$\Omega(\log \mathrm{JS}(P, Q_0))$	updates close to optimal
Grover & Ermon, 2018	density	none	none
This work	binary classifiers	$\Omega(\log \mathrm{KL}(P, Q_0))$	weak learning assumption on classifiers, weak dominance

1.1. Related work

In learning a density function iteratively, it is remarkable that most previous approaches (Guo et al., 2016; Li & Barron, 2000; Miller et al., 2017; Rosset & Segal, 2002; Tolstikhin et al., 2017; Zhang, 2003, and references therein) have investigated a single update rule, not unlike Frank–Wolfe optimisation:

$$q_t = h(\alpha_t j(d_t) + (1 - \alpha_t) j(q_{t-1})), \quad (1)$$

wherein h, j are some transformations that are in general (but not always) the identity, $(q_t)_{t\in\mathbb{N}}$ is the sequence of density functions learnt, and $(\alpha_t, d_t)_{t\in\mathbb{N}}$ are the step sizes and updates. The updates and step sizes are chosen so that for some measure of divergence $I(P, Q_t) \to 0$ as $t \to \infty$, where $I(P, Q_t)$ is the divergence of the true distribution P from Q_t. Grover & Ermon (2018) is one (recent) rare exception to (1) wherein alternative choices are explored. Few works in this area are accompanied by convergence proofs, and even less provide convergence rates (Guo et al., 2016; Li & Barron, 2000; Rosset & Segal, 2002; Tolstikhin et al., 2017; Zhang, 2003).

To establish convergence and/or bound convergence by a rate, *all* approaches necessarily make structural assumptions or approximations on the parameters involved in (1). These assumptions can be on the (local) variation of the divergence (Guo et al., 2016; Naito & Eguchi, 2013; Zhang, 2003), the true distribution or the updates (Dai et al., 2016; Grover & Ermon, 2018; Guo et al., 2016; Li & Barron, 2000),the step size (Miller et al., 2017; Tolstikhin et al., 2017), the previous updates, $(d_i)_{i\leq t}$,(Dai et al., 2016; Rosset & Segal, 2002), and so on. Often in order to produce the best geometric convergence bounds, the update is usually required required to be close to the optimal one (Tolstikhin et al., 2017, Cor. 2, 3). Table 1 compares the best results of the leading three to our approach. We give for each of them the updates aggregated, the assumptions on which rely the results and the *rate* to come close to a fixed value of KL divergence (Jensen-Shannon, JS, for (Tolstikhin et al., 2017)), which is just the order of the number of iterations necessary, hiding all other dependences for simplicity.

However, it must be kept in mind that for many of these works (viz. Tolstikhin et al., 2017) the primary objective is to develop an efficient black box sampler for P, in particular for large dimensions. Our objective however is to focus on furtive lack of formal results on the densities and convergence, instead leaving the problem of sampling from these densities as an open question.

2. Preliminaries

In the sequel $\mathcal{X}$ is a topological space. Unnormalised Borel measures on $\mathcal{X}$ are indicated by decorated capital letters, $\tilde{P}$, and Borel probability measures by capital letters without decoration, P. To a function $f : \mathcal{X} \to (-\infty, +\infty]$ we associate another function f^*, called the *Fenchel conjugate* with $f^*(x^*) \overset{\text{def}}{=} \sup_{x\in\mathcal{X}} \langle x^*, x\rangle - f(x)$. If f is convex, proper, and lower semi-continuous, $f = (f^*)^*$. If f is strictly convex and differentiable on $\mathrm{int}(\mathrm{dom}\, f)$ then $(f^*)' = (f')^{-1}$. Theorem-like formal statements are numbered to be consistent with their appearance in the appendix (§D) to which we defer all proofs.

An important tool of ours are the *f-divergences* of information theory (Ali & Silvey, 1966; Csiszár, 1967). The f-divergence of P from Q is $\mathrm{I}_f(P, Q) \overset{\text{def}}{=} \int f(\mathrm{d}P/\mathrm{d}Q)\,\mathrm{d}Q$, where it is assumed that $f : \mathbb{R} \to (-\infty, +\infty]$ is convex and lower semi-continuous, and Q dominates P.[2] Every f-divergence has a *variational representation* (Reid & Williamson, 2011) via the Fenchel conjugate:

$$\begin{aligned}
\mathrm{I}_f(P, Q) &= \int (f^*)^*\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right) \mathrm{d}Q \\
&= \int \sup_{t>0}\left(t \cdot \frac{\mathrm{d}P}{\mathrm{d}Q} - f^*(t)\right) \mathrm{d}Q \\
&= \sup_{u\in(\mathrm{dom}\, f^*)^{\mathcal{X}}} \int \left(u \cdot \frac{\mathrm{d}P}{\mathrm{d}Q} - f^* \circ u\right) \mathrm{d}Q \\
&= \sup_{u\in(\mathrm{dom}\, f^*)^{\mathcal{X}}} \Big(\mathrm{E}_P\, u - \mathrm{E}_Q\, f^* \circ u\Big), \quad (2)
\end{aligned}$$

where the supremum is implicitly restricted to measurable functions.

In contrast to the abstract family $(\mathrm{dom}\, f^*)^{\mathcal{X}}$, binary classification models tend to be specified in terms of *density*

[2] Common divergence measures such as Kullback–Liebler (KL) and total variation can easily be shown to be members of this family by picking f accordingly (Reid & Williamson, 2011). Several examples of these are listed in Table 2.

Table 2: Some common f-divergences and their variational components.

	I_f	$f(t)$	$f^*(t^*)$	$f'(t)$	$(f^* \circ f')(t)$
Kullback–Liebler	KL	$t \log t$	$\exp(t^* - 1)$	$\log t + 1$	t
Reverse KL	rKL	$\lvert t - 1\rvert$	$-\log(-t^*) - 1$	$-1/t$	$\log t - 1$
Hellinger	-	$(\sqrt{t} - 1)^2$	$3(t^* - 1)^{-1} - 1$	$1 - 1/t$	$\sqrt{t} - 1$
Pearson	χ^2	$(t - 1)^2$	$t^*(4 + t^*)/4$	$2(t - 1)$	$t^2 - 1$
GAN	GAN	$t \log t - (t + 1) \log(t + 1)$	$-\log\left(1 - \exp(t^*)\right)$	$-\log(t) - \log(t + 1)$	$\log(1 + t)$

Table 3: Classification decision rules.

Collection	$\mathcal{R}(\mathcal{X})$	$\mathcal{D}(\mathcal{X})$	$\mathcal{C}(\mathcal{X})$
Decision rule	$d \geq 1$	$D \geq 1 - D$	$c \geq 0$

$$\begin{tikzcd}
\mathcal{C}(\mathcal{X}) \arrow[rr, bend left, "\exp\circ"] \arrow[dr, "\sigma\circ"'] & & \mathcal{R}(\mathcal{X}) \arrow[r, "f'\circ"] & \operatorname{int}(\operatorname{dom} f^*)^{\mathcal{X}} \\
& \mathcal{D}(\mathcal{X}) \arrow[ur, "\varphi\circ"'] & &
\end{tikzcd}$$

Figure 4: Bijections for reparameterising (V).

ratios, *binary conditional distributions*, and *binary classifiers*, these are respectively[3]

$$\mathcal{R}(\mathcal{X}) \overset{\text{def}}{=} \{d : \mathcal{X} \to (0, \infty)\},\ \mathcal{D}(\mathcal{X}) \overset{\text{def}}{=} \{D : \mathcal{X} \to (0, 1)\},$$
$$\mathcal{C}(\mathcal{X}) \overset{\text{def}}{=} \{c : \mathcal{X} \to \mathbb{R}\}.$$

It is easy to see that these sets equivalent, with the commonly used connections

$$\varphi(D) \overset{\text{def}}{=} \frac{D}{1 - D}, \quad \sigma(c) \overset{\text{def}}{=} \frac{1}{1 + \exp(-c)},$$
$$(\varphi \circ \sigma) = \exp,$$

which are illustrated in Figure 4.

It is a common result (Nguyen et al., 2010; Grover & Ermon, 2018; Nowozin et al., 2016) that the supremum in (2) is achieved for $f' \circ \mathrm{d}P/\mathrm{d}Q$. It's convenient to define the *reparameterised variational problem*:

$$\underset{d \in \mathcal{F}}{\text{maximise}} \quad J(d) \overset{\text{def}}{=} \mathrm{E}_P\, f' \circ d - \mathrm{E}_Q\, f^* \circ f' \circ d, \tag{V}$$

where $\mathcal{F} \subseteq \mathcal{R}(\mathcal{X})$, so that the unconstrained maximum is achieved for $\mathrm{d}P/\mathrm{d}Q$. See §B for more details.

Example 1 (Neural classifier)**.** *A neural network with softmax layer corresponds to an element $D \in \mathcal{C}(\mathcal{X})$. To convert this to an element $d \in \mathcal{R}(\mathcal{X})$ simply substitute the softmax with an exponential activation function. This is just the arrow $\mathcal{C}(\mathcal{X}) \to \mathcal{R}(\mathcal{X})$ in Figure 4.*

Example 2 (*f*-GAN)**.** *The GAN objective (Goodfellow et al., 2014) is implicitly solving the reparameterised variational problem (V):*

$$\begin{aligned}
&\sup_{D \in \mathcal{D}(\mathcal{X})} \left(\mathrm{E}_P \log(D) + \mathrm{E}_Q \log(1 - D)\right) \\
&\quad = \sup_{D \in \mathcal{D}(\mathcal{X})} \left(\mathrm{E}_P (f' \circ \varphi) \circ D - \mathrm{E}_Q (f^* \circ f' \circ \varphi) \circ D\right) \\
&\quad = \mathrm{GAN}(P, Q),
\end{aligned}$$

where the function f is defined in Table 2, corresponding to the GAN f-divergence. In our derivation it's clear that (V) together with the isomorphisms in Figure 4 give a simple, principled choice for the "output activation function", g_f, of Nowozin et al. (2016).

3. Boosted density estimation

We fit distributions Q_t over the space $\mathcal{X}$ incrementally using the following update

$$\begin{aligned}
\tilde{Q}_t(\mathrm{d}x) &= d_t^{\alpha_t}(x) \cdot \tilde{Q}_{i-1}(\mathrm{d}x), \\
Q_t &= \frac{1}{Z_t} \tilde{Q}_t, \quad \text{where} \quad Z_t \overset{\text{def}}{=} \int \mathrm{d}\tilde{Q}_t,
\end{aligned} \tag{3}$$

where $\alpha_t \subseteq [0, 1]$ is the step size (for reasons that will be clear shortly), $d_t : \mathcal{X} \to \mathbb{R}_+$ is a density ratio, and we fix the initial distribution Q_0. After t updates we have the distribution

$$Q_t(\mathrm{d}x) \overset{\text{def}}{=} \frac{1}{\int \prod_{i=1}^t d_t^{\alpha_t}\, \mathrm{d}Q_0} \prod_{i=1}^t d_t^{\alpha_i}(x) Q_0(\mathrm{d}x). \tag{4}$$

Proposition 2. *The normalisation factors can be written recursively with $Z_t = Z_{t-1} \cdot \mathrm{E}_{Q_{t-1}} d_t^{\alpha_t}$.*

Proposition 3. *Let Q_t be defined via (4) with a sequence of binary classifiers $c_1, \ldots, c_t \in \mathcal{C}(\mathcal{X})$, where $c_i = \log d_i$ for $i \in [t]$. Then Q_t is an exponential family distribution with natural parameter $\alpha \overset{\text{def}}{=} (\alpha_1, \ldots, \alpha_t)$ and sufficient statistic $c(x) \overset{\text{def}}{=} (c_1(x), \ldots, c_t(x))$.*

The sufficient statistic of our distributions are classifiers that would hence be learned, along with the appropriate

[3] While it might seem like there are certain inclusions here (for example $\mathcal{D}(\mathcal{X}) \subseteq \mathcal{R}(\mathcal{X}) \subseteq \mathcal{C}(\mathcal{X})$), these categories of functions really are distinct objects when thought of with respect to their corresponding binary classification decision rules (listed in Table 3).

fitting of natural parameters. As explained in the proof, the representation may not be minimal; however, without further constraints on α, the exponential family is regular (Barndorff-Nielsen, 1978). A similar interpretation of a neural network in terms of parameterising the sufficient statistics of a *deformed exponential family* is given by Nock et al. (2017).

In the remainder of this section, we show how to learn the density ratios d_i and choose the step sizes α_i from observed data to ensure convergence $Q_t \to_t P$.

3.1. Helper results for the convergence of Q_t to P

The updates d_t learnt by solving (V) with Q replaced by Q_{t-1}. To make explicit the dependence of Q_t on α_t we will sometimes write $\mathrm{d}\tilde{Q}_t|_{\alpha_t} \overset{\text{def}}{=} d_t^{\alpha_t}\,\mathrm{d}\tilde{Q}_{t-1}$ and $\mathrm{d}Q_t|_{\alpha_t} \overset{\text{def}}{=} \frac{1}{Z_t}\,\mathrm{d}\tilde{Q}_t|_{\alpha_t}$. Since Q_t is an exponential family (Proposition 3), we measure the divergence between P and Q_t using the KL divergence (Table 2), which is the canonical divergence of exponential families (Amari & Nagaoka, 2000).

Notice that we can write any solution to (V) as $d_t = \mathrm{d}P/\mathrm{d}Q_{t-1} \cdot \varepsilon_t$, where we call $\varepsilon_t : \mathcal{X} \to \mathbb{R}_+$ the *error term* due to the fact that it is determined by the difference between the constrained and global solutions to (V). A more detailed analysis of the quantity ε_t is presented in §B.

Lemma 5. *For any $\alpha_t \in [0,1]$, letting Q_t, Q_{t-1} as in (3), we have:*

$$\begin{aligned}\mathrm{KL}(P, Q_t|_{\alpha_t}) \le{}& (1-\alpha_t)\,\mathrm{KL}(P, Q_{t-1}) \\ &+ \alpha_t(\log \mathrm{E}_P\,\varepsilon_t - \mathrm{E}_P \log \varepsilon_t)\end{aligned} \tag{5}$$

for all $d_t \in \mathfrak{R}(\mathcal{X})$, where $\varepsilon_t \overset{\text{def}}{=} (\mathrm{d}P/\mathrm{d}Q_{t-1})^{-1} d_t$.

Remark 3. *Grover & Ermon (2018) assume a uniform error term, $\varepsilon_t \equiv 1$. In this case Lemma 5 yields geometric convergence*

$$\forall \alpha_t \in [0,1]:\ \mathrm{KL}(P, Q_t|_{\alpha_t}) \le (1-\alpha_t)\,\mathrm{KL}(P, Q_{t-1}).$$

This result is significantly stronger than Grover & Ermon (2018, Thm. 2), who just show the non-increase of the KL divergence. If, in addition to achieving uniform error, we let $\alpha_t = 1$, then (5) guarantees $Q_t|_{\alpha_t=1} = P$.

We can express the update (4) and (5) in a way that more closely resembles Frank–Wolfe update (1). Since ε_t takes on positive values, we can identify it with a density ratio involving a (not necessarily normalised) measure $\tilde{R}_t$, as follows

$$\tilde{R}_t(\mathrm{d}x) \overset{\text{def}}{=} \varepsilon_t(x)\cdot P(\mathrm{d}x) \quad \text{and} \quad R_t \overset{\text{def}}{=} \frac{1}{\int \mathrm{d}\tilde{R}_t}\cdot \tilde{R}_t. \tag{6}$$

Introducing $\tilde{R}_t$ allows us to lend some interpretation to Lemma 5 in terms of the probability measure R_t. If we assume $\mathcal{X}$ is a measure space and that all measures have positive density functions (denoted by lowercase letters) with respect to the ambient measure then

$$q_t \propto d_t^{\alpha_t} q_{t-1} = \left(\frac{p}{q_{t-1}}\varepsilon_t\right)^{\alpha_t} q_{t-1} = \tilde{r}_t^{\alpha_t} q_{t-1}^{1-\alpha_t},$$

or equivalently

$$\exists C > 0:\ \log q_t = \alpha_t \log r_t + (1-\alpha_t) q_{t-1} - C.$$

This shows the manner in which (3) is a special case of (1).

Corollary 6. *For any $\alpha_t \in [0,1]$ and $\varepsilon_t \in [0,+\infty)^{\mathcal{X}}$, letting Q_t as in (4) and R_t from (6). If R_t satisfies*

$$\mathrm{KL}(P, R_t) \le \gamma\,\mathrm{KL}(P, Q_{t-1}) \tag{7}$$

for $\gamma \in [0,1]$, then

$$\mathrm{KL}(P, Q_t|_{\alpha_t}) \le (1-\alpha_t(1-\gamma))\,\mathrm{KL}(P, Q_{t-1}).$$

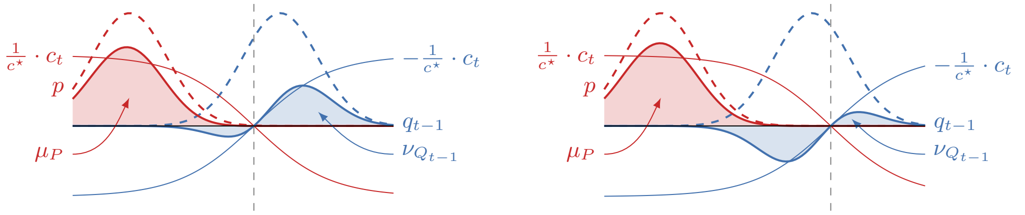
We obtain the same geometric convergence as Tolstikhin et al. (2017, Cor. 2) for a boosted distribution Q_t which is not a convex mixture, which, to our knowledge, is a new result. Corollary 6 is restricted to the KL divergence *but* we do not need the technical domination assumption that Tolstikhin et al. (2017, Cor. 2) require. From the standpoint of weak versus strong learning, Tolstikhin et al. (2017, Cor. 2) require a condition similar to (7), that is, the iterate R_t has to be close enough to P. It is the objective of the following sections to relax this requirement to something akin to the weak updates common in a boosting scheme.

3.2. Convergence under weak assumptions

In the previous section we have established two preliminary convergence results (Remark 3, Corollary 6) that equal the state of the art and/or rely on similarly, strong assumptions. We now show how to relax these in favour of placing some weak conditions on the binary classifiers learnt in Equation 2. Define the two *expected edges of* c_t (cf. Nock & Nielsen, 2008):

$$\nu_{Q_{t-1}} \overset{\text{def}}{=} \frac{1}{c^\star}\,\mathrm{E}_{Q_{t-1}}[-c_t] \quad \text{and} \quad \mu_P \overset{\text{def}}{=} \frac{1}{c^\star}\,\mathrm{E}_P[c_t], \tag{8}$$

where $c^\star \overset{\text{def}}{=} \max_t \operatorname{ess\,sup} |c_t|$ and the maximum is implicitly over all previous iterations. Classical boosting results rely on assumption on such edges for different kinds of c_t (Freund & Schapire, 1997; Schapire, 1990; Schapire & Singer, 1999) and the implicit and weak assumption, that we also make, that $0 < c^\star < \infty$, that is, the classifiers have bounded and nonzero confidence. By construction then, $\nu_{Q_{t-1}}, \mu_P \in [-1,1]$. The difference of sign of c_t is due to the decision rule for a binary classifier (Table 3), whereby $c_t(x) \ge 0$ reflects that c_t classifies $x \in \mathcal{X}$ as originating from P rather than Q_{t-1}, and vice versa for $-c_t(x)$.



(a) WLA is satisfied since μ_P and $\nu_{Q_{t-1}}$ are positive.

(b) WLA is violated since $\nu_{Q_{t-1}}$ is negative.

Figure 5: Illustration of WLA in one dimension with a classifier c_t and its decision rule (indicated by the dashed grey line). The red (resp. blue) area is the area under the $c_t/c^\star \cdot p$ (resp. $-c_t/c^\star \cdot q_{t-1}$) line (where p, q_{t-1} are corresponding density functions), that is, μ_P (resp. $\nu_{Q_{t-1}}$).

Assumption 1 (Weak learning assumption)**.**

$$\exists \gamma_P, \gamma_Q \in (0,1] : \ \mu_P \geq \gamma_P, \quad \nu_{Q_{t-1}} \geq \gamma_Q. \qquad \text{(WLA)}$$

The weak learning assumption is in effect a separation condition of P and Q_{t-1}. That is, the decision boundary associated with c_t correctly divides most of the mass of P and most of the mass of Q_{t-1}. This is illustrated in Figure 5. Note that if Q_{t-1} has converged to P, the weak learning assumption cannot hold. This is reasonable since as $Q_{t-1} \to P$ it becomes harder to build a classifier to tell them apart. We note that classical boosting would rely on a single inequality for the weak learning assumption (involving the two edges) (Schapire & Singer, 1999) instead of two as in WLA. The difference is, however, superficial as we can show that both assumptions are equivalent (Lemma 7 in §D). A boosting algorithm would ensure, for any given error $\varrho > 0$, that there exists a number of iterations T for which we do have $\mathrm{KL}(P, Q_T) \leq \varrho$, where T is required to be polynomial in all relevant parameters, in particular $1/\gamma_P, 1/\gamma_Q, c^\star, \mathrm{KL}(P, Q_0)$. Notice that we have to put $\mathrm{KL}(P, Q_0)$ in the complexity requirement since it can be arbitrarily large compared to the other parameters.

Let

$$\alpha_t \overset{\text{def}}{=} \min\left\{1, \frac{1}{2c^\star} \log\left(\frac{1 + \nu_{Q_{t-1}}}{1 - \nu_{Q_{t-1}}}\right)\right\}. \qquad (9)$$

Theorem 18. *Suppose WLA holds at each iteration. Then using Q_t as in (4) and α_t as in (9), we are guaranteed that $\mathrm{KL}(P, Q_T) \leq \varrho$ after a number of iterations T satisfying:*

$$T \geq 2 \cdot \frac{\mathrm{KL}(P, Q_0) - \varrho}{\gamma_P \gamma_Q}.$$

There is more to boosting: the question naturally arises as to whether faster convergence is possible. A simple observation allows to conclude that it should require more than just WLA. Define

$$\mu_{\varepsilon_t} \overset{\text{def}}{=} \frac{1}{c^\star} \cdot \mathrm{E}_P \log \varepsilon_t,$$

the normalised expected log-density estimation error. Then we have $\mu_P = (1/c^\star) \cdot \mathrm{KL}(P, Q_{t-1}) + \mu_{\varepsilon_t}$, so controlling μ_P does not give substantial leverage on $\mathrm{KL}(P, Q_t)$ because of the unknown μ_{ε_t}. We show that an additional weak assumption on μ_{ε_t} is all that is needed with WLA, to obtain convergence rates that compete with Tolstikhin et al. (2017, Lem. 2) but using much weaker assumptions. We call this assumption the *Weak Dominance Assumption (WDA)*.

Assumption 2 (Weak Dominance Assumption)**.**

$$\exists \varGamma_\varepsilon > 0, \forall t \geq 1 : \ \mu_{\varepsilon_t} \geq -\varGamma_\varepsilon \qquad \text{(WDA)}$$

The assumption WDA takes its name from the observation that we have

$$c_t = \log d_t = \log\left(\frac{\mathrm{d}P}{\mathrm{d}Q_{t-1}} \cdot \varepsilon_t\right) \quad \text{and} \quad |c_t| \leq c^\star,$$

so by ensuring that ε_t is going to be non-zero P-almost everywhere, WDA states that nowhere in the support do we have Q_{t-1} with respect to P. This also looks like a weak finite form of absolute continuity of P with respect to Q_{t-1}, which is not unlike the boundedness condition on the log-density ratio of Li & Barron (2000, Thm. 1).

Provided WLA and WDA hold at each iteration, Theorem 19 yields geometric boosting convergence rates.

Theorem 19. *Suppose WLA and WDA hold at each boosting iteration. Then after T boosting iterations:*

$$\mathrm{KL}(P, Q_T) \leq \left(1 - \frac{\gamma_P \min\{2, \gamma_Q/c^\star\}}{2(1 + \varGamma_\varepsilon)}\right)^T \cdot \mathrm{KL}(P, Q_0).$$

Note that the bound obtained in Theorem 19 is, in fact, logarithmic in $\mathrm{KL}(P, Q_0)$. That is have $\mathrm{KL}(P, Q_T) \leq \varrho$ when

$$T \geq \frac{2(1 + \varGamma_\varepsilon)}{\gamma_P \min\{2, \gamma_Q/c^\star\}} \log\left(\frac{\mathrm{KL}(P, Q_0)}{\varrho}\right).$$

4. Experiments

Let $t \in \{0, \dots, T\}$. Our experiments mostly take place in $\mathcal{X} \overset{\text{def}}{=} \mathbb{R}^2$, where we use a simple neural network classifier $c_t \in \mathfrak{C}(\mathcal{X})$, which we train using cross entropy error by post composing it with the logistic sigmoid: $\sigma \circ c_t$. After training c_t we transform it into to a density ratio using an exponential function: $d_t \overset{\text{def}}{=} \exp \circ c_t$ (cf. §2) which we use to update Q_{t-1}.

In most experiments we train for $T > 1$ rounds therefore we need to sample from Q_{t-1}.[4] Our setting here is simple and so this is easily accomplished using random walk Metropolis–Hastings. As noted in the introduction, in more sophisticated domains it remains an open question how to sample effectively from a density of the form (4), in particular for a support having large dimensionality.

Since our classifiers c_t are the outputs of a neural network they are unbounded, this violates the assumptions of §3, therefore in most cases we use the naive choice $\alpha_t \overset{\text{def}}{=} 1/2$.

Metrics At each $t \in \{0, \dots, T\}$ we estimate compute $\mathrm{KL}(P, Q_t)$, (normalised) Negative Log-Likelihood (NLL) $\frac{1}{\mathrm{E}_P \log p} \mathrm{E}_P \log q_t$, and accuracy $\mathrm{E}_P \left[\!\left[c_t > \frac{1}{2} \right]\!\right]$. Note that we normalise NLL by its true value to make this quantity more interperable. The KL divergence is computed using numerical integration, and as such it can be quite tricky to ensure stability when running stochastically varying experiments, and becomes very hard to compute in dimensions higher than $n = 2$. In these computationally difficult cases we use NLL, which is much more stable by comparison. We plot the mean and 95% confidence intervals for these quantities.

4.1. Results

Complete details about the experimental procedures including target data and network architectures are deferred to the supplementary material (§E).

4.1.1. Error and convergence

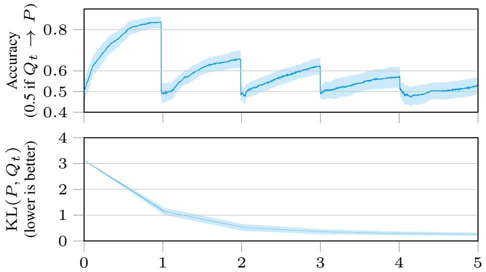


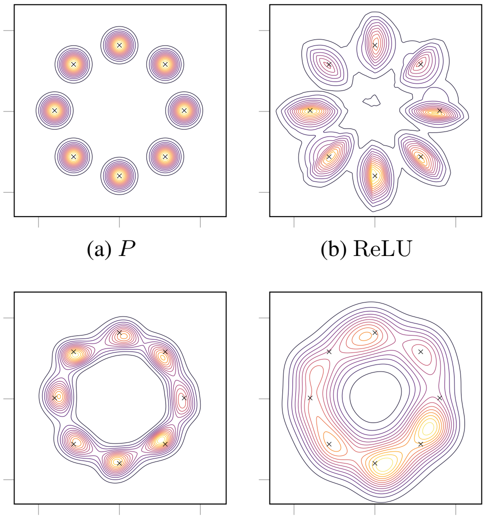
Figure 6: As $\mathrm{KL}(P, Q_t) \to 0$ it becomes harder and harder to train a good classifier c_t.

In order to minimise the divergence of Q_t from P we need to train a sufficiently good classifier c_t such that we can build a good approximation to $\mathrm{d}P/\mathrm{d}Q_{t-1}$. Naturally as $Q_t \to P$ it should become harder and harder to tell the difference between a sample from P and Q_t with high probability.

This is exactly what we observe. In Figure 6 we train a classifier with the same neural network topology as in §4.1.2. The test accuracy over the course of training before each t is plotted. As $\mathrm{KL}(P, Q_t) \to 0$ samples from P and Q_t become harder and harder to tell apart and the best accuracy we can achieve over the course of training decreases, approaching $1/2$. Dually, the higher the training accuracy achieved by c_t, the greater the reduction from $\mathrm{KL}(P, Q_{t-1})$ to $\mathrm{KL}(P, Q_t)$, thus the decreasing saw-tooth shape in Figure 6 is characteristic of convergence.

4.1.2. Activation functions

To look at the effect of the choice of activation function we train the same network topology, for a set of activation functions: Numerical results trained to fit a ring of Gaussians are plotted in Figure 8a, contour plots of some of the resulting densities are presented Figure 7. All activation functions except for $\mathrm{Softplus}$ performed about the same by the end of the sixth round, with ReLU and SELU being the marginal winners. It is also interesting to note the narrow error ribbons on $\tanh$ compared to the other functions, indicating more consistent training.



(a) P (b) ReLU

(c) $\tanh$ (d) $\mathrm{Softplus}$

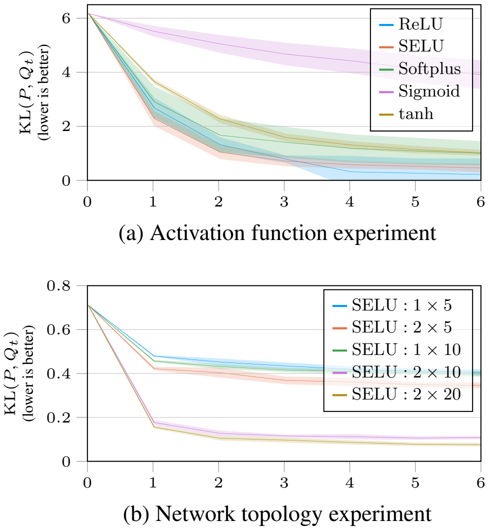
Figure 7: The effect of different activation functions, modelling a ring of Gaussians. The “petals” in the ReLU condition are likely due to the linear hyperplane sections the final network layer being shaped by the final exponential layer.

[4] It is easy to pick Q_0 to be convenient to sample from.

4.1.3. Network topology

To compare the effect of the choice of network architecture we fix activation function and try a variety of combinations of network architecture, varying both the depth and the number nodes per layer. For this experiment the target distribution P is a mixture of 8 Gaussians that are randomly positioned at the beginning of each run of training. Let $m \times n$ denote a fully connected neural network c_t with m hidden layers and n nodes per layer. After each hidden layer we apply the SELU activation function.

Numerical results are plotted in Figure 8b. Interestingly doubling the nodes per layer has little benefit, showing only moderate advantage. By comparison, increasing the network depth allows us to achieve over a 70% reduction in the minimal divergence we are able to achieve.



(a) Activation function experiment

(b) Network topology experiment

Figure 8: KL divergence for a variety of activation functions and architectures over six iterations of boosting.

4.1.4. Convergence across dimensions

For this experiment we vary the dimension $n \in \{2, 4, 6\}$ of the space $\mathcal{X} = \mathbb{R}^n$ using a neural classifier c_t that is trained without regard for overfitting and look at the convergence of NLL (Figure 9). After we achieve the optimal NLL of 1, we observe that NLL becomes quite variable as we begin to overfit. Secondly overfitting the likelihood becomes harder as we increase the dimensionality, taking roughly two times the number of iterations to pass NLL $= 1$ in the $n = 4$ condition as the $n = 2$ condition. We conjecture that not overfitting is a matter of early stopping boosting, in a similar way as it was proven for the consistency of boosting algorithms (Bartlett & Traskin, 2006).

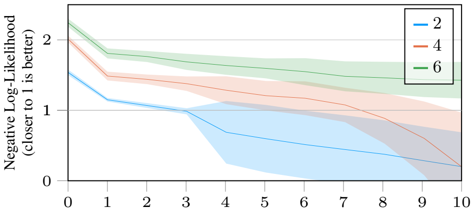


Figure 9: Convergence in more dimensions.

4.1.5. Comparison with Tolstikhin et al. (2017)

To compare the performance of our model (here called Discrim) with AdaGAN we replicate their Gaussian mixture toy experiment,[5] fitting a randomly located eight component isotropic Gaussian mixture where each component has constant variance. These are sampled using the code provided by Tolstikhin et al. (2017).

We compute the coverage metric[6] of Tolstikhin et al. (2017): $C_\kappa(P, Q) \overset{\text{def}}{=} P(\text{lev}_{>\beta}\, q)$, where $Q(\text{lev}_{>\beta}\, q) = \kappa$, and $\kappa \in [0, 1]$. That is, we first find β to determine a set where most of the mass of Q lies, $\text{lev}_{>\beta}\, q$, then look at how much of the mass of P resides there.

Results from the experiment are plotted in Figure 10. Both Discrim and AdaGAN converge closely to the true NLL, and then we observe the same characteristic overfitting in previous experiments after iteration 4 (Figure 10a). It is also interesting that this also reveals itself in a degradation of the coverage metric Figure 10b.

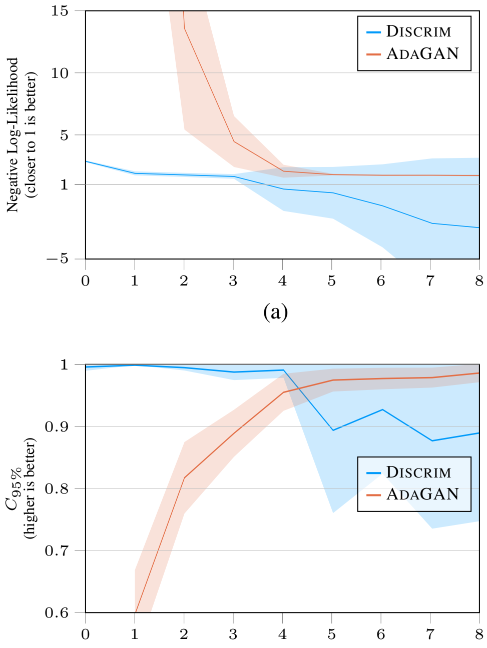
Notably AdaGAN converges tightly, with NLL centred around its mean, while Discrim begins to vary wildly. However the AdaGAN procedure includes a step size that decreases with $1/t$ — thereby preventing overfitting — whereas Discrim uses a constant step size of 1/2. Suggesting that a similarly decreasing procedure for α_t may have desirable properties.

4.1.6. Comparison with KDE

In this experiment we compare our boosted densities with Kernel Density Estimation (KDE). For this experiment we train a deep neural network with three hidden layers. The step size α is selected to minimise NLL by evaluating the training set at 10 equally spaced points over $[0, 1]$. We compare the resultant density after $T = 2$ rounds with a

[5] This is the experiment gaussian_gmm.py at github.com/tolstikhin/adagan

[6] The coverage metric C_κ can be a bit misleading since any density Q that covers P will yield high $C_\kappa(P, Q)$, no matter how spread out it is. This is the case at $t = 0$ when we initially fit Q_0. A high coverage metric, however, is sufficient to claim that a model Q has not ignored any of the mass of P when combined with another metric such as NLL. That is, a high C_κ is a necessary condition for mode-capture.



(a)

(b)

Figure 10: Comparing the performance of DISCRIM and ADAGAN.

variety of kernel density estimators, with bandwidth selected via the Scott/Silverman rule.[7]

Results from this experiment are displayed in Figure 11 (in the supplementary material). On average Q_1 fits the target distribution P better than all but the most efficient kernels, and at Q_2 we begin overfitting, which aligns with the observations made in §4.1.4. We note that his performance is with a model with around 200 parameters, while the kernel estimators each have 2000 — *i.e.* we achieve KDE's performances with models whose size is the *tenth* of KDE's. Also, in this experiment α_t is selected to minimise NLL, however it is not hard to imagine that a different selection criteria for α_t would yield better properties with respect to overfitting.

4.2. Summary

We summarise here some key experimental observations:

- Both the activation functions and network topology have a large effect on the ease of training and the quality of the learned density Q_T with deeper networks with fewer nodes per layer yielding the best results (§4.1.2, §4.1.3).
- When the networks c_t are trained long enough we observe overfitting in the resulting densities Q_T and instability in the training procedure after the point of overfitting (§4.1.4 §4.1.6, §4.1.5), indicating that a procedure to take $\alpha_t \to 0$ should be optimal.
- We were able to match the performance of kernel density estimation with a naive procedure to select α_t (§4.1.6).
- We were able to at least match the performance of ADAGAN with respect to density estimation (§4.1.5).

Finally, while we have used KDE as a point of comparison of algorithm, there is no reason why the two techniques could not be combined. Since KDE is a closed form mixture distribution that's quite easy sampled, there is no reason why one couldn't build some kind of kernel density distribution and use this for Q_0 which one could refine with a neural network.

5. Conclusion

The prospect of learning a density iteratively with a boosting-like procedure has recently been met with significant attention. However, the success these approaches hinge on the existence of oracles satisfying very strong assumptions.

By contrast, we have shown that a weak learner in the original sense of Kearns (1988) is sufficient to yield comparable or better convergence bounds than previous approaches when reinterpreted in terms of learning a density ratio. To derive this result we leverage a series of related contributions including 1) a finer characterisation of f-GAN (Nowozin et al., 2016), and 2) a full characterisation we learn, in terms of an exponential family.

Experimentally, our approach shows promising results for with respect to the capture of modes, and significantly outperforms AdaGAN during the early boosting iterations using a comparatively very small architecture. Our experiments leave open the challenge to obtain a black box sampler for domains with moderate to large dimension. We conjecture that the exponential family characterisation should be of significant help in tackling this challenge.

Acknowledgements

We wish to thank Ilya Tolstikhin, Olivier Bousquet, and the reviewers, for many helpful suggestions and remarks. The boosted density implementation is available at https://github.com/ZacCranko/BoostedDensities.jl.

[7] The Scott and Silverman rules yield identical bandwidth selection criteria in the two-dimensional case.

References

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

Amari, S.-I. and Nagaoka, H. *Methods of Information Geometry*. Oxford University Press, 2000.

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

Banerjee, A., Merugu, S., Dhillon, I., and Ghosh, J. Clustering with Bregman divergences. *JMLR*, 6:1705–1749, 2005.

Barndorff-Nielsen, O. *Information and Exponential Families in Statistical Theory*. Wiley Publishers, 1978.

Bartlett, P. and Traskin, M. Adaboost is consistent. In *NIPS*19*, 2006.

Csiszár, I. Information–type measures of difference of probability distributions. *Studia Scientiarum Mathematicarum Hungarica*, 17:123–149, 1967.

Dai, B., He, N., Dai, H., and Song, L. Provable bayesian inference via particle mirror descent. In *Artificial Intelligence and Statistics*, pp. 985–994, 2016.

Dudík, M., Phillips, S.-J., and Schapire, R.-E. Performance guarantees for regularized maximum entropy density estimation. In *COLT*, 2004.

Freund, Y. and Schapire, R. E. A Decision-Theoretic generalization of on-line learning and an application to Boosting. *JCSS*, 55:119–139, 1997.

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.

Grover, A. and Ermon, S. Boosted generative models. In *AAAI*, 2018.

Guo, F., Wang, X., Fan, K., Broderick, T., and Dunson, D.-B. Boosting variational inference. *CoRR*, abs/1611.05559, 2016.

Innes, M. Flux: Elegant machine learning with julia. *Journal of Open Source Software*, 2018. doi: 10.21105/joss.00602.

Kearns, M. Thoughts on hypothesis boosting. *Unpublished manuscript*, 45:105, 1988.

Khan, M.-E., Babanezhad, R., Lin, W., Schmidt, M.-W., and Sugiyama, M. Faster stochastic variational inference using proximal-gradient methods with general divergence functions. In *UAI*, 2016.

Li, J. Q. and Barron, A. R. Mixture density estimation. In *Advances in neural information processing systems*, pp. 279–285, 2000.

Locatello, F., Khanna, R., Ghosh, J., and Rätsch, G. Boosting variational inference: an optimization perspective. *CoRR*, abs/1708.01733, 2017.

McDiarmid, C. Concentration. In Habib, M., McDiarmid, C., Ramirez-Alfonsin, J., and Reed, B. (eds.), *Probabilistic Methods for Algorithmic Discrete Mathematics*, pp. 1–54. Springer Verlag, 1998.

Miller, A.-C., Foti, N.-J., and Adams, R.-P. Variational boosting: Iteratively refining posterior approximations. In *ICML*, pp. 2420–2429, 2017.

Naito, K. and Eguchi, S. Density estimation with minimization of U-divergence. *Machine Learning*, 90(1):29–57, 2013.

Nguyen, X., Wainwright, M. J., and Jordan, M. I. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Nock, R. and Nielsen, F. A $\mathbb{R}$eal Generalization of discrete AdaBoost. *Artificial Intelligence*, 171:25–41, 2007.

Nock, R. and Nielsen, F. On the efficient minimization of classification-calibrated surrogates. In *Advances in neural information processing systems*, pp. 1201–1208, 2008.

Nock, R., Cranko, Z., Menon, A. K., Qu, L., and Williamson, R. C. f-gans in an information geometric nutshell. In *Advances in Neural Information Processing Systems*, pp. 456–464, 2017.

Nowozin, S., Cseke, B., and Tomioka, R. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pp. 271–279, 2016.

Penot, J.-P. *Calculus without derivatives*, volume 266. Springer Science & Business Media, 2012.

Reid, M. D. and Williamson, R. C. Information, divergence and risk for binary experiments. *Journal of Machine Learning Research*, 12(Mar):731–817, 2011.

Rosset, S. and Segal, E. Boosting density estimation. In *NIPS*, pp. 641–648, 2002.

Schapire, R. E. The strength of weak learnability. *Machine Learning*, pp. 197–227, 1990.

Schapire, R. E. and Singer, Y. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.

Simić, S. On a new converse of Jensen's inequality. *Publications de l'Institut Mathématique*, 85:107–110, 2009a.

Simić, S. On an upperbound for Jensen's inequality. *Journal of Inequalities in Pure and Applied Mathematics*, 10, 2009b.

Tolstikhin, I.-O., Gelly, S., Bousquet, O., Simon-Gabriel, C., and Schölkopf, B. Adagan: Boosting generative models. In *NIPS*, pp. 5430–5439, 2017.

Zhang, T. Sequential greedy approximation for certain convex optimization problems. *IEEE Trans. IT*, 49:682–691, 2003.