Likelihood Estimation of Deep Generative Models with Annealed Importance Sampling

Xuan Wang
Department of Computer Science
Courant Institute of Mathematics
New York University
xwang@nyu.edu

Abstract

Recent work from Theis et. al.[9] shows the difficulties in evaluating generative models. In particular, the quality of generated samples and the likelihood of a model can be totally independent. However, likelihood estimation remains a important aspect of automatic evaluation, rather than relying on human eyeballing on generated samples to asses the quality of model. In this project, I explored the effectiveness of using AIS to evaluate generative models. The advantage of AIS over importance sampling has been shown, especially in the situation of high dimensional spaces. Results in VAE and GAN suggest that VAE is superior than GAN in terms of consistently better likelihood. The sensitiveness of GAN to the choice of observation model has been shown, suggesting particular caution has to be taken when evaluating with AIS, or we need to look for better likelihood evaluation theories and practices for GAN.

1 Introduction

Deep generative models have shown remarkable potential and progress in generating convincing samples of images. A few well-known examples include variational autoencoders(VAEs)[3], generative adversarial networks(GANs)[2], generative moment matching networks(GMMNs)[5] and autoregressive models(PixelRNNs)[7].

While many of these models are capable of synthesizing appealing images, decent evaluation remains a challenge. And it has been pointed out by Theis et. al.[9] that just an overfitted model can produce quite high quality images, and we should resort to carefully designed evaluation methods on generative models. Although they also pointed out that good likelihood and good quality of samples can be quite independent of each other, measurements through likelihood estimation are still considered important: 1) automated measurements is necessary when doing large scale or real life application of generative models; 2) likelihood still provides an idea of how well the model fits data and whether overfits or underfits, which is important for further improvement of models.

However, the probability distributions we are interested in are of high dimensionality, which results in a huge support to be inspect in order to evaluate the likelihood. More severe situations such as unmatched supports of proposal distribution and target distribution can happen quite often in high dimensional spaces, which makes simple importance sampling perform poorly. A recent approach from [10] proposed using Annealed Importance Sampling to bridge the gap between the proposal distribution and target distribution by inserting intermediate distributions using MCMC. Their results showed an advantage over other widely used methods such as Parzen windows[8] and Importance Weighted Autoencoder[1].

This project aims to experiment on Annealed Importance Sampling method to explore its properties and to do likelihood estimation on generative models.

2 Background

2.1 Generative Adversarial Network

A GAN[2] is a generative model trained by a game between a generator network and a discriminator network. Discriminator network is designed to distinguish samples from the generator distribution from real data. The generator network, on the other hand, strives to produce samples to best fool the discriminator. The game between the two networks is characterized in the following minimax problem:

$$\min_G \max_D \mathbb{E}_{x\sim p_data}[\log D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))] \tag{1}$$

It is important to notice that GAN, given a latent code, does not produce for its generated sample a distribution but deterministically. This fact results in the practice that we associate with a sample an *observation model*

$$p(x|z) \sim \mathcal{N}(G(z), \sigma) \tag{2}$$

in order to use sampling techniques to evaluate the likelihood.

2.2 Variational Auto-Encoder

VAE[3] is a probabilistic directed graphical model. It factorizes a joint distribution over a set of latent random variables z and the observed variables x by $p(x, z) = p(x|z)p(z)$. In order to generate a sample, one follows the two step approach: 1) sample $z \sim p(z)$; 2) sample $x \sim p(x|z)$. The prior $p(z)$, is usually chosen to be simple to sample from, e.g. a standard Gaussian or uniform. The data likelihood p(x|z) is usually a Gaussian distribution whose center μ and variance σ depend on z through a deep neural network, known as the decoder network. An approximate inference model $q(z|x)$ is also jointly learned called an encoder or recognition network, to approximate the posterior $p(z|x)$. The decoder network and the encoder networks are jointly trained to maximize the evidence lower bound (ELBO):

$$\mathbb{E}_{q(z|x)}[\log p(x|z) - \mathrm{KL}(q(z|x)||p(z))] \leq \log p(x) \tag{3}$$

2.3 Annealed Importance Sampling

Annealed Importance Sampling [6] produces a sequence of points, $x^{(1)}$,...,$x^{(N)}$, and corresponding weights, $w^{(1)}$,...,$w^{(N)}$. To estimate an expectation of a function (in our case the likelihood of some model of interest), $e(x)$, we approximate it by $\overline{e} = \sum_{i=1}^N w^{(i)} e(x^{(i)}) / \sum_{i=1}^N w^{(i)}$. To generate one such a point of $x^{(i)}$ and $w^{(i)}$, we generate a sequence of points x_{n-1},...,x_0 as follows:

Generate x_{n-1} from p_n.
Generate x_{n-2} from x_{n-1} using $\mathcal{T}_{n-1}$.
...
Generate x_1 from x_2 using $\mathcal{T}_2$.
Generate x_0 from x_1 using $\mathcal{T}_1$.

where each $\mathcal{T}_i$ is an MCMC transition operator.
Then $x^{(i)} = x_0$ and $w^{(i)}$ is set to be

$$\frac{f_{n-1}(x_{n-1})}{f_n(x_{n-1})} \frac{f_{n-2}(x_{n-2})}{f_{n-1}(x_{n-2})} \cdots \frac{f_1(x_1)}{f_2(x_1)} \frac{f_0(x_0)}{f_1(x_0)} \tag{4}$$

In our setting, the target distribution is the joint probability distribution $p(x, z)$, the initial distribution is the prior $p(z)$, and we are using intermediate distributions of the form

$$f_j = p(z)^{1-\beta_j} p(x, z)^{\beta_j} = p(z)p(x|z)^{\beta_j} \tag{5}$$

Also in VAE since we have the approximation inference distribution $p(z|x)$ available, we may also use it as the initial distribution.

Table 1: Average variance ratios of AIS over importance sampling varying the number of samples

#samples	50	100	200	500	1000
avg variance ratio (%)	54.89	53.91	54.80	56.67	67.48

3 Methodology

This work primarily explores the effectiveness of AIS. There are two lines of work: 1) use a simple generator with only one dimension hidden code to experiment the properties of AIS; 2) use AIS to evaluate VAEs and GANs.

3.1 Explore AIS effectiveness

Consider a deterministic generator, we can view it as a perfect GAN. In the practical evaluation, since GAN does not decode the latent code to predict probability distribution for the observation, the actual computation of $p(x|z)$, see eq.5, would be a delta function $\delta(x)$. In order for the probabilities to be well-defined, we need to associate with the GAN an *observation model*. Here I am using a Gaussian kernel density model, following the line of Wu et al[10].

On the other hand, a deterministic generator associated with a Gaussian kernel density observation model can also be viewed as a VAE which gives identical observation variance for all the latent codes.

3.2 Evaluate VAE and GAN with AIS

Both VAE and GAN models trained on MNIST[4] are evaluated using AIS. For VAE, experiments have been done both using the approximation inference network $q(z|x)$ as initial AIS sampling distribution (and also in the intermediate factorization of $p(x, z)$ in eq.5) and using a manual set Gaussian prior $p(z)$. Using $q(z|x)$ results a rewriting of eq.5 as

$$f_j = q(z|x)p(x|z)^{\beta_j} \tag{6}$$

For GAN, as mentioned in 3.1, a Gaussian observation model is associated in order to compute $p(x|z)$, see eq.2.

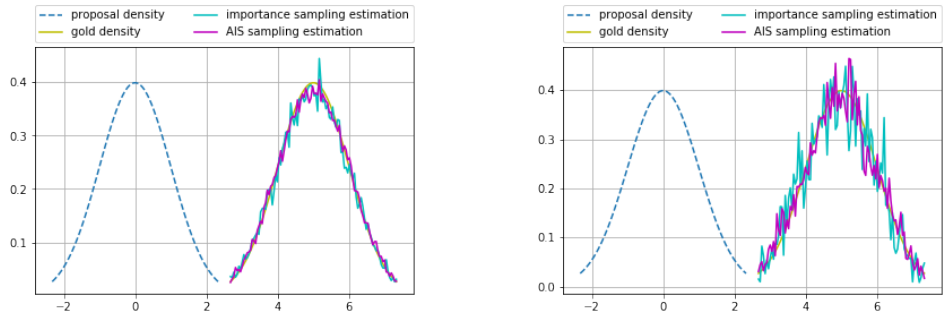
4 Experiments

4.1 AIS vs Importance Sampling

Unmatched supports AIS was proposed as a way of alleviating the problem of importance sampling when the supports of proposal distribution and target distribution are far away from each other. By designing a target distribution of Gaussian whose mean is varying in distance with the proposal distribution, we see that AIS does considerably better than importance sampling by reducing the average variance by half, see fig.1a.

Varying sample sizes By reducing the number of samples used to evaluate the likelihood of a point, I found that AIS is consistently better than importance sampling, see fig.1b and table.1.

Sharper observation model When evaluating GAN, the choice of observation model (eq.2) will have an impact on the accuracy of likelihood estimation. However, the specific choice is up to tuning: we typically will not assume a kernel variance too high because GAN itself defines a delta function, and on the other hand, if kernel variance too high then $p(x|z)$ can be flattened to semantically unrelated dimensions, which also causes variance in the likelihood estimation. Experiments show that AIS does considerably better than importance sampling when the kernel variance is small, and it shows the robustness over different choices of the kernel variances, see fig.2 and table.2.



(a) Unmatched supports (b) #samples=100

Figure 1: (a) AIS vs importance sampling when the proposal distribution has support away from the target distribution. In this case the ratio of average variance of AIS over importance sampling is only 43.92%; (b) when the number of samples becomes small, AIS still performs better than importance sampling. The ratio of average variance of AIS over importance sampling is 46.50%

Table 2: Average variance ratios of AIS over importance sampling with different kernel variances

kernel variance	0.01	0.05	0.1	0.25	0.4
avg variance ratio (%)	9.54	16.17	15.80	34.93	46.87

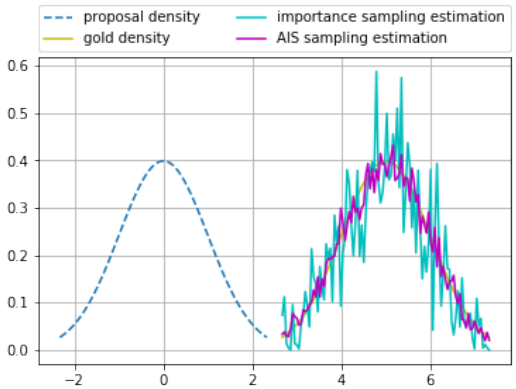


Figure 2: When the kernel variance down to 0.01, AIS does significantly better than importance sampling. The average variance ratio of AIS over importance sampling is 10.02%.

4.2 AIS on deep generative models

Experiments have been done on VAE and GAN. With VAE, we can have a typical Gaussian prior as proposal distribution, and, different from GAN, we can also use the approximation inference model $q(z|x)$ as proposal (eq.6).

Table 3: Log likelihood estimation of VAE and GAN. The number of AIS intermediate distributions (eq.5) are set to 200, with a sigmoid schedule to ensure that when at the beginning and the end there are the densest distributions inserted, since these are the places where the distributions changes the fastest.

Model	VAE-10	GAN-10	VAE-50	GAN-50
train	551.81	266.45	613.23	417.09
test	535.35	256.12	572.14	430.87

Table 4: Log likelihood estimation of VAE and GAN. The number of AIS intermediate distributions (eq.5) are set to 200, with a sigmoid schedule to ensure that when at the beginning and the end there are the densest distributions inserted, since these are the places where the distributions changes the fastest.

observation model variance	0.001	0.005	0.01	0.025	0.05	0.1	0.5
log-likelihood	-9894.83	-1014.48	-60.98	243.64	188.98	39.06	-489.40

VAE vs GAN As shown in table.3, VAE performs significantly better than GAN regardless of the architecture size. (Here VAE-10 refers to a smaller model and VAE-50 a larger one, same goes with GAN.) For both models, larger architectures yield better likelihoods.

Due to the effect that estimation of GAN is very sensitive to the choice of observation model (see the second paragraph of 4.2), I chose the observation model variance (0.025, see table.4) which gives the best log-likelihood score throughout the experiment.

Problem with AIS when evaluating GAN As discussed in section3.1 and section3.2, when evaluating GAN, the choice of observation model is up to tuning, different kernel variance may produce quite different results. A kernel variance too high will result in $p(x|z)$ being flattened to semantically unrelated dimensions, while a kernel variance too low will place too much penalty on a final z which is far from the ideal value which generates the observation x.

Results can be found in table.4. It shows that the evaluation of GAN is very sensitive to the specific choice of observation model variance: a variance from 0.001 to 0.025 will result in the log-likelihood change from -9894.83 to 243.64! It suggests that evaluating GAN with AIS or general sampling method is greatly subjective to the choice of observation model. Particular caution needs to be taken when we deal with GAN.

5 Conclusion

In this project, I explored the effectiveness of using AIS to evaluate generative models. The advantage of AIS over importance sampling has been shown, especially in the situation of high dimensional spaces. Results in VAE and GAN suggest that VAE is superior than GAN in terms of consistently better likelihood. The sensitiveness of GAN to the choice of observation model has been shown, suggesting particular caution has to be taken when evaluating with AIS, or we need to look for better likelihood evaluation theories and practices for GAN.

Acknowledgments

I sincerely thank Professor Rajesh Ranganath who provided thought-provoking and eye-opening lectures. Also thank my lovely classmates for the enjoyable talk after class around theories and life in general. I am graduating with gratefulness and in tears of joy.

References

[1] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[4] Yann LeCun. Gradient-based learning applied to document recognition. 1998.

[5] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727, 2015.

[6] Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

[7] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[8] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[9] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[10] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016.