The Ill-defined Problem of Maximum Likelihood Estimation

Yibo Yang
yibo.yang@uci.edu

February 28, 2022

1 Overview

A central goal of likelihood-based generative modeling is to estimate a density model of some data-generating distribution from data samples. Indeed, much of unsupervised learning has been historically referred to as *density estimation*. The premise, of course, is that the data distribution actually *has* a probability density function, a common assumption underlying the derivation of the most popular deep generative models such as GANs [Goodfellow et al., 2014], VAEs [Kingma and Welling, 2014, Rezende et al., 2014], and normalizing flows [Rezende and Mohamed, 2016]. This assumption, while convenient, may not hold in practice. Indeed, much of real world data such as natural images is known to concentrate on a low-dimensional manifold of the ambient space [Pope et al., 2021], and behaves as if it does not have a (Lebesgue) density. In this case, standard maximum-likelihood training can become problematic, yielding arbitrarily high density and ill-conditioned models. As researchers push for increasingly expressive and flexible density models, this issue is becoming increasingly visible, especially in normalizing flow models [Behrmann et al., 2021, Koehler et al., 2021].

In this note, I review the statistical basis of maximum likelihood estimation from first principles, and examine how it can break down, depending on whether or not the data distribution has a density. In both cases, the maximum likelihood estimator can become ill-defined because of over-fitting on finite training data. Collecting more training data may fix this problem if the data distribution has a density. However, this does not work if the data distribution has no density, where the breakdown is more fundamental, caused by the underlying KL divergence between the data and the model distribution "maxing out". Finally, I discuss how a useful density model might still be obtained in the latter case, e.g., through appropriate regularization, noise injection, or manifold learning.

2 Background, and Why Maximum Likelihood

The standard recipe for estimating a model by maximum likelihood estimation is as follows:

1. Decide on a model family for the data, parameterized by a parameter vector θ. The model $p_\theta(x)$ is either a probability mass function (PMF) when the data is discrete, or a density function when the data is continuous.

2. Obtain i.i.d. data samples $\{x_1, x_2, ..., x_n\}$, and define the (sample) log-likelihood function (the scaling by $\frac{1}{n}$ is optional and for later convenience),

$$L_n(\theta) := \frac{1}{n} \log \prod_{i=1}^{n} p_\theta(x_i) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i)$$

3. Apply your favorite optimization algorithm to maximize the above log-likelihood, resulting in the maximum-likelihood estimate $\hat{\theta}_n$:

$$\hat{\theta}_n := \arg\max_\theta L_n(\theta)$$

$p_{\hat{\theta}_n}(x)$ is then the resulting model estimated from data.

In a standard statistics textbook, the starting point is usually that there exists a *true* parameter vector θ_0, such that our data is generated under the *true model* $p_{\theta_0}(x)$. Then under appropriate assumptions, the maximum-likelihood *estimator* $\hat{\theta}_n$ (viewed as a random variable) has nice statistical

properties, such as consistency and asymptotic normality, essentially ensuring that $\hat{\theta}_n$ converges to θ_0 in a nice way as we increase the sample size n.

In machine learning, of course, it is almost never the case that the data is really generated by some model p_{θ_0} within our parametric family of models; i.e., our model is *mis-specified*. Nor can we hope to find the global maximum as required in the definition of $\hat{\theta}_n$. But this does not prevent us from parameterizing p_θ by some fancy neural network architecture, and training it by maximizing the log-likelihood (or a lower bound of it) with some form of SGD. What's the theoretical justification of MLE in practice then?

The standard argument is that we are approximately computing an M-projection of the data distribution onto our model family. Since we have no knowledge of the true data distribution, we can only be very general and represent the true (unknown) data distribution by a probability measure P, defined over the data space $\mathcal{X}$. Given n data samples $x_1 ..., x_n$, we can define an empirical measure, $P_n := \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$, a mixture of Dirac delta measures centered on the data samples. The log-likelihood can be then written as an expectation resembling a negative cross entropy:

$$L_n(\theta) = \mathbb{E}_{X\sim P_n}[\log p_\theta(X)]. \tag{1}$$

From here on out things start to get tricky, and most explanations I've seen focus on the case of discrete data. Let's also only consider the discrete setting for now, so the model p_θ is a PMF with support over a countable $\mathcal{X}$, and let the PMF of the true data distribution P be p (which always exists). Similarly, using lower-case p_n to denote the PMF of the empirical measure (just need to replace Dirac delta with Kronecker delta in the definition of P_n), Eq. 1 can then be rewritten as

$$L_n(\theta) = \mathbb{E}_{X\sim P_n}[\log p_\theta(X)] = -KL(p_n \| p_\theta) - \mathbb{H}[p_n].$$

Since the discrete entropy $\mathbb{H}[p_n]$ is constant w.r.t. θ, we see that maximizing the log-likelihood (LHS) is equivalent to minimizing the KL divergence between the empirical and the model distributions. Furthermore, by the law of large numbers, as the number of samples increases, the (sample) log-likelihood function $L_n(\theta)$ (the negative "training loss") converges to the population log-likelihood (the negative "test loss", or "generalization error"), denoted by $L_\infty(\theta)$; i.e.,

$$\mathbb{E}_{X\sim P_n}[\log p_\theta(X)] \to \mathbb{E}_{X\sim P}[\log p_\theta(X)] := L_\infty(\theta) = -KL(p\|p_\theta) - \mathbb{H}[p], \tag{2}$$

so under the right regularity conditions, the argmax w.r.t. θ would also converge, i.e.,

$$\arg\max_\theta \mathbb{E}_{X\sim P_n}[\log p_\theta(X)] \to \arg\max_\theta \mathbb{E}_{X\sim P}[\log p_\theta(X)] = \arg\min_\theta KL(p\|p_\theta).$$

The limit $\theta^* = \arg\min_\theta KL(p\|p_\theta)$ corresponds to the M-projection of the true distribution p onto the model family, and would simply be θ_0 if there is no model mis-specification (indeed this sketches out the usual consistency proof of the MLE).

3 A Quick Look at A Data Distribution Without A Density

The above is fine and dandy, except when we want to extend it to the case of continuous data. The complicating factor is that unlike in the discrete case where the data distribution P always admits a PMF, in the continuous case (e.g., $\mathcal{X} \subset \mathbb{R}^D$), P may not have a probability density function (i.e., P may not be *absolutely continuous w.r.t. the Lebesgue measure* in measure theory jargon).

For example, the data might be generated by first drawing from a univariate normal distribution, then multiplying it by a constant vector $(1, 2)$, so the resulting data lies on a straight line in $\mathcal{X} = \mathbb{R}^2$. Figure 1 illustrates the situation [1]. Then there does not exist *any* function $p : \mathcal{X} \to \mathbb{R}^+$, such that integrating it over a set A gives us the probability $P(A)$, for every (measurable) set $A \subset \mathcal{X}$. For instance, if A is the line segment $A = \{(t, 2t)| - 1 \le t \le 1\}$, then no matter what function p we use, we'll always end up with $\int_A p(x)dx = 0$, when in fact $P(A) > 0$, i.e., there is non-zero probability that the data lands on the line segment A. The issue is that the line segment is "infinitely thin" according to the reference measure "dx" (the Lebesgue measure in $\mathbb{R}^2$), with respect to which our integral and probability density are defined.

[1] This generative process is basically probabilistic PCA, but without adding observation noise at the end.

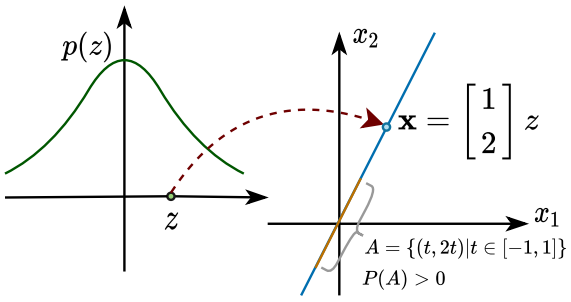

Figure 1: Illustration of a data distribution that does not admit a probability density function.

4 The Problem

Regardless of whether the data distribution admits a density function, the MLE can be ill-defined, i.e., there is no setting of θ that maximizes the log-likelihood $L_n(\theta)$, especially when $\max_\theta L_n(\theta) = \infty$. I'll discuss this issue for the two cases where the data distribution does and does not admit a density, explain why the latter case is more problematic, and give evidence in machine learning research that suggests the prevalence of the latter case in practice.

If P has a density. The textbook discussion usually just assumes the data has a density p; then the M-projection argument from the discrete case goes through largely unchanged (Eq 2 continues to hold, with $\mathbb{H}$ now denoting differential entropy instead of Shannon entropy); see, e.g., Theorem 16.3 of this lecture note. So, if a helpful oracle can guarantee us that the data distribution admits a density, then we can rest assured that maximum-likelihood training is still asymptotically minimizing the KL divergence between the (unknown) data density p and the model family.

Nonetheless, there is a potential problem of the maximum-likelihood estimator being ill-defined. Most relevant to our discussions is the problem of "extreme overfitting", caused by having too few data samples relative to the model complexity, even when there is no model mis-specification and the data has a density (this also happens more generally when the data has no density). Take the simple example of estimating a univariate Gaussian distribution, without model mis-specification. If only $n = 1$ sample is given, the log-likelihood can be made arbitrarily large by setting the mean to the sample location, and shrinking the variance parameter to arbitrarily small values. This is analogous to the classic discrete example of estimating the probability of heads v.s. tails from a single coin toss. In this example, the pathology disappears as soon as we have two (different) data samples. However, increasing the number of data samples is not a panacea. For some flexible models (such as a Gaussian mixture with two or more components), the sample MLE objective L_n can always be made arbitrarily large, even when the population objective L_∞ has a finite maximum. And if the data has no density, then even an infinite number of samples does not help.

If P has no density. When the data distribution does not have a density (which as we'll see is likely the case for much of natural data), the whole M-projection argument breaks down, and no amount of training data can fix the ill-defined MLE problem.

The basic issue is that the KL divergence $KL(P\|P_\theta)$, which we've come to regard as the principled minimization objective underlying MLE, is simply undefined (or infinite) in this case. Note I've switched the notation from p_θ to capital P_θ to emphasize we are now considering the probability measure defined by our model density p_θ, i.e., $P_\theta(A) := \int_A p_\theta(x)dx$, for every measurable event $A \subset \mathcal{X}$. The KL divergence between two measures P_1 and P_2 is defined as $KL(P_1\|P_2) := \int \log \frac{dP_1}{dP_2} dP_1$ when P_1 is absolutely continuous w.r.t. P_2, and undefined or taken to be ∞ otherwise. Without getting into too much detail, the absolute continuity requirement just means that wherever P_2 assigns zero probability, P_1 must also assign zero probability; when this is satisfied, the integrand $\frac{dP_1}{dP_2}$ is well-defined and reduces to the familiar PDF (or PMF) ratio $\frac{p_1(x)}{p_2(x)}$ in the continuous (or discrete) case. This requirement is violated when the data measure P no longer admits a density (e.g., is concentrated on a lower dimensional manifold, like a line in $\mathbb{R}^2$), yet we still model it by a density p_θ. This happens, in our 2D example, if I model the data by a 2D Gaussian density (or any density on $\mathbb{R}^2$); then the Gaussian measure P_θ cannot assign positive probability to any line segment A, yet $P(A) > 0$ under the data.

If we look past the fact that we are no longer (approximately) computing the M-projection of the data distribution, does it still make sense to train a density model by maximum likelihood? It depends on the goal of generative modeling. As we train such a model, it will assign increasingly high density (∞ if allowed) to the training data, and ideally the entire data manifold. This is OK if we only intend to use the model to draw samples from the data manifold. Of course, a model that only memorizes and returns training data may not be useful, so care still needs to be taken to prevent extreme overfitting. However, it won't make much sense to use the resulting p_θ as a density model of the data, considering the fact that 1. both the sample ("train") log-likelihood and more importantly the population ("test") log-likelihood can always be made arbitrarily large by tweaking the model family (e.g., using a more flexible normalizing flow architecture); and 2. even within the same model family, a model with a high likelihood may poorly capture the shape of the data *within* the data manifold. Note that point 1, essentially $\max_q \mathbb{E}_{X\sim P}[\log q(X)] = \infty$, cannot happen when P admits a density and has a finite differential entropy $\mathbb{H}[p]$, because the objective would be bounded above by $-\mathbb{H}[p]$.

Natural data behaves like it does not have a density. By natural data, I mean raw signals from the natural world like images and sound waves. It is widely conjectured that such real-world data lies on a low-dimensional manifold of the ambient space and thus has a low intrinsic dimension, often much lower than the nominal dimension of the data (e.g., the number of pixels in an image). There's been a long line of research lending experimental support to this *manifold hypothesis* [Pope et al., 2021]. Recent work by Pope et al. [2021] looked into actually establishing the intrinsic dimensionality of popular image datasets such as CIFAR and ImageNet; e.g., the intrinsic dimensionality of ImageNet is estimated to be between 26 and 43.

Empirical experience training deep generative models also point to the low intrinsic dimension of images. e.g., when fitting a VAE with a Gaussian observation model on MNIST images, the ELBO, which lower bounds the log-likelihood $L_n(\theta)$, can be made seemingly arbitrarily large, by simply shrinking the variance of the observation model [Lucas et al., 2019]. As another example, current deep normalizing flow models trained on images tend to be poorly conditioned and close to non-invertible [Behrmann et al., 2021, Koehler et al., 2021]. This is despite the use of dequantization noise during training (potential solution 3, to be discussed below), suggesting the flow transform has trouble mapping between the (likely still very "thin") noisy data manifold to the high dimensional ambient space of pixels, while staying invertible. Regardless whether or not the true distribution of natural images has a density (we may never know for sure), it *behaves as if it does not.*

5 Potential Solutions

The problem of arbitrarily high population log-likelihood, essentially due to an infinite $KL(P\|P_\theta)$ (also known as KL "maxing out", or the related JS divergence "maxing out" in GANs), has been recognized for a while, and spurred the development of likelihood-free approaches; the prime example is the Wasserstein-GAN [Arjovsky et al., 2017]. Such a likelihood-free approach avoids the potentially ill-defined problem of density estimation altogether. If, still, we want to learn a "useful" density model of the data, what can we do? Below I summarize a few potential solutions.

1. **Use a discrete model for discrete data.**

If the data observations are actually discrete, e.g., digital representations of natural images taking values in $\{0, 1, ..., 255\}^D$, then they may be better off modeled directly by a PMF, such as a PixelCNN [van den Oord et al., 2016] (or other autoregressive models), a discrete normalizing flow [Tran et al., 2019, Hoogeboom et al., 2019], or a VAE with a discrete observation model.

The discrete setting is better behaved, in that there is a unique PMF which maximizes the sample log-likelihood (the empirical PMF p_n), with the maximum achievable log-likelihood equal to $-\mathbb{H}[p_n]$. The trivial solution p_n is arguably also the result of overfitting (especially when n is small), but does not occur in practice often because the model p_θ is not expressive enough to reach this solution. Occasionally, there is still a silly case of MLE being undefined caused by the solution $\hat{\theta}_n$ existing on the boundary of the parameter set (e.g., logistic regression on perfectly separated data, with θ growing indefinitely), but this is typically harmless. Of course, a PMF only makes sense for discrete data. When we do use a continuous density model [2], it is paramount to:

[2] Incidentally, many discrete generative models are parameterized with an underlying density model for efficiency, such as PixelCNN++, or the base distribution of a discrete flow [Hoogeboom et al., 2019].

2. **Constrain or regularize the model**, to prevent infinite maximum likelihood.

This underlies many classical methods in statistics, such as Lasso and ridge regression, as well as MAP estimation with a Bayesian prior on θ. Besides adding a regularizer to the MLE objective (or going full Bayesian), we can also directly constrain the parameterization of the model family. The first example that comes to my mind is the common practice of fixing the variance of a Gaussian observation model $p_\theta(x|z)$ in a VAE. The amount and the nature of constraint imposed to the model is usually determined by a validation set, but sometimes also dictated by ulterior goals such as representation learning (e.g., for VAEs, we know that getting good log-likelihood is insufficient for learning good latent representations [Alemi et al., 2018]), or compression (e.g., restricting a VAE's likelihood model in an appropriate way turns it into a data compressor [Theis et al., 2017, Ballé et al., 2017, Yang et al., 2020] or an estimator of the rate-distortion function [Yang and Mandt, 2022]).

3. **Add noise to the data.** For D-dimensional data, convolving it with even a small amount of continuous noise in $\mathbb{R}^D$ ensures the resulting noisy data distribution has a density. Then there's no more issue of "KL maxing out", and we're back to doing (approximate) M-projection. This idea has been used in the analysis of GANs by Arjovsky and Bottou [2017], where the GAN and the data distributions (both with low-dimensional support) are "smoothed out" by a small Gaussian noise, in order to work with a notion of Jensen-Shannon divergence between them.

If the data is discrete, particularly if it's the quantized representation of a continuous signal (such as digital representations of natural images), then injecting *dequantization noise* during training is a particularly elegant approach. The resulting modified MLE objective is bounded from above by the (discrete) empirical data entropy, and maximum likelihood has the interpretation of minimizing the cost of lossless data compression under the discretized density model; see Theis et al. [2016] for details. This is the predominant approach for training continuous density models (especially normalizing flows) on digital images, and compression cost serves as a principled metric for model evaluation/comparison. Furthermore, the distribution of the noise itself can be optimized [Ho et al., 2019], if we view noise injection as inferring a posterior distribution over the unobserved value of the raw signal before it underwent quantization.

4. **Manifold learning**. The high-level idea here is to separate the task of estimating the support of the data distribution (i.e., the "data manifold") from estimating a probability distribution over the support. The first task is more of a geometry problem, and affects the *realism* of the samples generated by the model, whereas the latter is more of a statistics problem affecting the "spread" or "diversity" of generated samples [Lui et al., 2017]. The two-stage approach is taken by Dai and Wipf [2019]: they first train a VAE to recover the data manifold (where the model is free to assign ∞ density), and then train a separate VAE to estimate the aggregate posterior of the first VAE. The data distribution can then be accurately sampled by first drawing a sample from the second-stage VAE, and then passing it through the generative process of the first-stage VAE. This approach is reminiscent of a VQ-VAE and results in much better sample quality than vanilla Gaussian VAEs, although it does not yield a density model of the data over the estimated manifold. The flow review article by Papamakarios et al. [2021] touches on the topic of parameterizing a normalzing flow on low-dimension Riemannian manifolds, and increasing research on machine learning in non-Euclidean spaces may also offer useful tools here.

6 Closing Thoughts

Technically, the case of data being discrete is really the same as the data admitting a density, if we regard the PMF as the density w.r.t. an appropriate reference measure, i.e., the counting measure of $\mathcal{X}$. The broader distinction of whether or not the data distribution has a density just boils down to whether we know the true support of the data (this is the set of all the outcomes $\mathcal{X}$ in the discrete case, or $\mathbb{R}^D$ or a sub-manifold in the continuous case). If we do, we can declare that "the true data distribution has a density" (w.r.t. a reference measure over the support), and then we can (in principle) proceed to estimate a model of the data density over the true support. But even this broader distinction is in some sense irrelevant to the model estimation problem. As long as we live in the real world and can only learn from a finite amount of data, we may never know the true support of the data distribution and risk overfitting, assigning zero probability to events that might actually occur. Imagine estimating the categorical probabilities of an alien-invented dice with (for all we know) infinitely many faces each marked with a different symbol; even positing a Bayesian prior requires us to know all the possible outcomes! Such is the reality of statistical

inference, and fortunately the alien dice situation does not occur in practice often.

References

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, 2014.

D. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2016.

Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XJk19XzGq2J.

Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger Grosse, and Jörn-Henrik Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1792–1800. PMLR, 2021.

Frederic Koehler, Viraj Mehta, and Andrej Risteski. Representational aspects of depth and conditioning in normalizing flows. In *International Conference on Machine Learning*, pages 5628–5636. PMLR, 2021.

James Lucas, George Tucker, Roger Grosse, and Mohammad Norouzi. Understanding posterior collapse in generative latent variable models. *arXiv preprint arXiv:1903.05789*, 2019.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

A. van den Oord, N. Kalchbrenner, O. Vinyals, A. Graves L. Espeholt, and K. Kavukcuoglu. Conditional Image Generation with PixelCNN Decoders. In *Advances in Neural Information Processing Systems 29*, pages 4790–4798, 2016.

Dustin Tran, Keyon Vafa, Kumar Agrawal, Laurent Dinh, and Ben Poole. Discrete flows: Invertible generative models of discrete data. In *Advances in Neural Information Processing Systems*, pages 14719–14728, 2019.

Emiel Hoogeboom, Jorn Peters, Rianne van den Berg, and Max Welling. Integer discrete flows and lossless compression. In *Advances in Neural Information Processing Systems*, pages 12134–12144, 2019.

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken ELBO. In *International Conference on Machine Learning*, pages 159–168. PMLR, 2018.

L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy Image Compression with Compressive Autoencoders. In *International Conference on Learning Representations*, 2017.

J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end Optimized Image Compression. In *International Conference on Learning Representations*, 2017.

Yibo Yang, Robert Bamler, and Stephan Mandt. Variational Bayesian Quantization. In *International Conference on Machine Learning*, 2020.

Yibo Yang and Stephan Mandt. Towards empirical sandwich bounds on the rate-distortion function. In *International Conference on Learning Representations*, 2022.

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, Apr 2016. URL http://arxiv.org/abs/1511.01844.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pages 2722–2730. PMLR, 2019.

Kry Yik Chau Lui, Yanshuai Cao, Maxime Gazeau, and Kelvin Shuangjian Zhang. Implicit manifold learning on generative adversarial networks. *arXiv preprint arXiv:1710.11260*, 2017.

Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.