Alon Rosen

Hardness of Score Estimation

He's started five of his eleven novels to date with the definite article. We've had two of them with "it", there's been one "but", two "ats", one "and" and a "Dolores".[*]

We now shift from prediction to *generation*. Rather than assign labels to new inputs, the goal is to produce fresh samples from an unknown distribution given empirical samples from it.

Diffusion is a method for generation. Forward diffusion gradually adds Gaussian noise, while reverse diffusion tries to undo this process. The central learning problem is *score estimation*: at each noise level, given a noisy point, estimate the correction that guides the reverse process back toward fresh samples from the data distribution.

We will ask whether these score functions can be learned efficiently from samples. The hCLWE assumption gives a negative answer: sufficiently accurate score estimation would allow one to extract the hidden hCLWE direction from empirical samples, and hence cannot be done efficiently.

The main points of this chapter are:

- Generative models learn to produce fresh samples from empirical data.
- Reverse diffusion generates samples using learned score estimates.
- Cryptography can make score estimation for reverse diffusion computationally hard.

[*]Monty Python, *Monty Python's Flying Circus*.

1 Diffusion

Watching Thomas Hardy write a novel, Dennis the live broadcaster tries to predict the first word. Five out of Hardy's prior novels start with the definite article. This suggests a pattern, but exceptions to the rule make it difficult for Dennis to guess how the novel will begin.

Generative models are tasked with producing fresh samples given empirical samples drawn from an unknown distribution. The task appears throughout learning, where a sample might be a sentence, an image, or a video clip. The challenge is that only a limited number of examples are observed and the underlying distribution might not admit a simple description.

Suppose that we are given training data S whose datapoints are drawn from an unknown distribution D. The generated samples should behave like new independent draws rather than copies of the samples already seen. Let T be a training procedure. The generator $G = T(S)$ is trained on S and uses fresh randomness each time.

Definition 1 (Learning to sample)**.** *We say that T ε-learns to sample from D if, for training data S with datapoints sampled from D, $G = T(S)$, $Y \leftarrow G$, and an independent $X \leftarrow D$, we have*

$$d_{\mathrm{TV}}\big((S, Y), (S, X)\big) \le \varepsilon.$$

The joint distribution rules out *memorization*: a distinguisher given S can test whether $Y \in S$. This always accepts when Y is sampled from S, but more rarely accepts a fresh $X \sim D$. The implicit assumption is that D places noticeable mass outside S; otherwise generation is uninteresting.

Learning to generate does not require an explicit description of D, only a procedure for sampling from it. *Diffusion* learns one by moving through progressively less noisy versions of D.

Other generative models approach the task differently. *Autoregressive models* sample one token at a time. *Latent-variable models* such as variational autoencoders sample a simple hidden variable and decode it into data, while *adversarial models* train a generator to fool a discriminator. Diffusion instead adds noise gradually and learns to reverse it.

1.1 Forward diffusion

Instead of learning how to sample from D in one step, diffusion builds a path from D to a distribution that we already know how to sample from: a standard Gaussian.

Start with $X_0 \sim D$ over $\mathbb{R}^n$. For a small step size η, one forward step shrinks $X_k \in \mathbb{R}^n$ by $\sqrt{1-\eta}$ and adds independent Gaussian noise of scale $\sqrt{\eta}$:

$$X_{k+1} = \sqrt{1-\eta}\, X_k + \sqrt{\eta}\, Z_k, \qquad Z_k \sim \mathcal{N}(0, I_n). \tag{1}$$

Unrolling the recurrence shows that the coefficient of X_0 in X_k is $(1-\eta)^{k/2}$, so the process gradually forgets where it started. The Gaussian part of X_k is the sum of all perturbations added so far, each shrunk by the later steps. Since these perturbations are independent, this accumulated noise has variance $\eta \sum_{r=0}^{k-1} (1-\eta)^r = 1 - (1-\eta)^k$ in each coordinate, which tends to one.

At equilibrium, $X_k \sim \mathcal{N}(0, I_n)$, and $X_{k+1} \mid X_k$ and $X_k \mid X_{k+1}$ have the same distribution. Away from equilibrium, the forward transition is known, but the reverse transition from X_{k+1} to X_k also depends on the distribution D_k of X_k. This is the information *reverse diffusion* must recover.

1.2 Reverse diffusion

For large T, D_T is close to $\mathcal{N}(0, I_n)$, so we may start from a fresh Gaussian sample and run the diffusion process in reverse to obtain

$$\mathcal{N}(0, I_n) \approx D_T \longrightarrow D_{T-1} \longrightarrow \cdots \longrightarrow D_0 = D.$$

As k decreases, D_k moves away from Gaussian equilibrium and increasingly reflects the structure of D, so the reverse transitions must somehow approximate the distribution D_k.

Let p_k denote the density of D_k on $\mathbb{R}^n$. To understand one reverse step, suppose $X_{k+1} = \mathbf{y}$ and consider a possible predecessor $\mathbf{x}$. Its weight in the reverse distribution $X_k \mid X_{k+1}$ is determined by $p_k(\mathbf{x})\, p_{k+1|k}(\mathbf{y} \mid \mathbf{x})$, where $p_k(\mathbf{x})$ is the density of D_k at $\mathbf{x}$, and $p_{k+1|k}(\mathbf{y} \mid \mathbf{x})$ is the *forward* transition density from $\mathbf{x}$ to $\mathbf{y}$. The forward transition is known, so the unknown factor in the reverse weight is $p_k(\mathbf{x})$.

Definition 2 (Score)**.** *The score of D_k with density p_k is $s_k(\mathbf{x}) = \nabla \log p_k(\mathbf{x})$.*

To demystify this definition, let us look at a small reverse step in three stages:

- *Localize.* The forward transition has scale about $\sqrt{\eta}$, so for small η mostly predecessors $\mathbf{x}$ near $\mathbf{y}$ matter. Thus we only need to understand p_k locally.
- *Linearize.* For a small displacement $\mathbf{v}$, first-order Taylor expansion gives

  $$\log p_k(\mathbf{x} + \mathbf{v}) - \log p_k(\mathbf{x}) \approx \langle s_k(\mathbf{x}), \mathbf{v} \rangle.$$

  Only the relative weights of nearby predecessors matter, and $p_k(\mathbf{x} + \mathbf{v})/p_k(\mathbf{x}) \approx e^{\langle s_k(\mathbf{x}), \mathbf{v} \rangle}$. Thus the score determines these weights locally.
- *Reverse.* The Gaussian forward weight has a quadratic log-density. Adding the linear approximation leaves the quadratic term unchanged and only shifts its center. Thus the reverse step is heuristically Gaussian with the same scale, centered by the score.

Thus, knowing the score at each noise level should suffice to run reverse diffusion.

1.3 Sampling reduces to score estimation

Assume that the data have bounded second moment, that the score at each positive noise level is Lipschitz, and that the forward process becomes close to a standard Gaussian in polynomial time.

Theorem 1 (Sampling reduces to score estimation)**.** *Under these conditions, sufficiently accurate scores at the required noise levels yield an efficient sampler ε-close to D in total variation.*

Proof. With exact scores and the exact endpoint D_T, reversing the forward transitions would return D. The key is therefore to understand how the score determines each small reverse step, and then to control the errors introduced by approximation.

There are three approximation errors: starting from a standard Gaussian rather than D_T, discretizing the reverse path, and replacing each score by an estimate. Longer forward time, smaller steps, and lower mean squared score error control them, respectively.

For the discretized reverse path, approximate each small step by a Gaussian of scale about $\sqrt{\eta}$ centered at $\mathbb{E}[X_k \mid X_{k+1}]$. The score determines this center exactly.

Proposition 1. *The score $s_{k+1}(\mathbf{y})$ determines the conditional mean of $X_k \mid X_{k+1}$ as*

$$\mathbb{E}[X_k \mid X_{k+1} = \mathbf{y}] = \frac{\mathbf{y} + \eta s_{k+1}(\mathbf{y})}{\sqrt{1-\eta}}.$$

Proof. Conditional on $X_k = \mathbf{x}$, the transition is $\mathcal{N}(\sqrt{1-\eta}\,\mathbf{x}, \eta I_n)$, so

$$\nabla_{\mathbf{y}} \log p_{k+1|k}(\mathbf{y} \mid \mathbf{x}) = \frac{\sqrt{1-\eta}\,\mathbf{x} - \mathbf{y}}{\eta}.$$

The marginal $p_{k+1}(\mathbf{y})$ averages these transition densities over $\mathbf{x}$. Differentiating this average and dividing by $p_{k+1}(\mathbf{y})$ computes the score; by Bayes' rule, the normalized weights are exactly those of $X_k \mid X_{k+1} = \mathbf{y}$. Hence

$$s_{k+1}(\mathbf{y}) = \frac{1}{\eta}\,\mathbb{E}\Big[\sqrt{1-\eta}\,X_k - \mathbf{y} \,\Big|\, X_{k+1} = \mathbf{y}\Big],$$

which rearranges to the stated formula. □

Repeating these approximate reverse steps and applying the three error bounds above yields the claimed sampler. □

Accurate score estimates suffice for sampling, but as we will soon see not all distributions admit efficient score estimation.

2 Score estimation

Reverse diffusion sampling reduces to estimating the scores s_k of the noised distributions D_k. We now ask how to learn these scores from samples, and when this can be done efficiently.

2.1 Learning the score

The forward process lets us train a model using examples we can generate ourselves. Starting from a data sample $X_0 \sim D$, run the known forward process. Each example consists of the input X_{k+1}; because we generated the last step, we know its noise, with $-\sqrt{\eta}\,Z_k = \sqrt{1-\eta}\,X_k - X_{k+1}$. We train a function f, for example a deep neural network, to predict $-Z_k/\sqrt{\eta}$ from X_{k+1}.

By the forward update and Proposition 1,

$$\mathbb{E}[-Z_k/\sqrt{\eta}\,|\,X_{k+1} = \mathbf{y}] = \big(\sqrt{1-\eta}\,\mathbb{E}[X_k \mid X_{k+1} = \mathbf{y}] - \mathbf{y}\big)/\eta = s_{k+1}(\mathbf{y}).$$

Thus the recorded output is noisy, but its conditional mean is exactly the score. Squared loss recovers this conditional mean, so the best possible choice is $f = s_{k+1}$. We therefore minimize

$$\mathbb{E}\left[\left\| f(X_{k+1}) + \frac{Z_k}{\sqrt{\eta}} \right\|_2^2\right].$$

Repeat this training procedure at each noise level needed for reverse diffusion, or let a single model take the noise level as an additional input. Theorem 1 only requires small mean squared error on samples from D_k, so accuracy is needed only where D_k places mass; the density p_k itself need not be learned.

2.2 Hardness of score estimation

In hCLWE, given samples, distinguish $\mathcal{N}(0, I_n)$ from a Gaussian pancake distribution with hidden unit vector $\mathbf{w}$. Orthogonal to $\mathbf{w}$ it is standard Gaussian, while along $\mathbf{w}$ its mass lies near periodic layers. Finding $\mathbf{w}$ is assumed to be computationally hard.

Let $T_{\mathbf{w}}$ accept $\mathbf{x}$ when $\langle \mathbf{x}, \mathbf{w} \rangle$ is periodic. It accepts a fresh pancake sample with probability close to 1 and a standard Gaussian sample with probability close to 0, so an ε-close sampler must make $T_{\mathbf{w}}$ accept with almost the same probability as on fresh pancake samples.

Theorem 2 (Hardness of score estimation)**.** *Accurate polynomial-time score estimation for Gaussian pancakes would give a polynomial-time algorithm for hCLWE.*

Proof. The form of the score follows directly from the geometry of the pancakes. Orthogonal to $\mathbf{w}$, the planted distribution is still standard Gaussian, so the score agrees there with the Gaussian score $-\mathbf{x}$. Only the one-dimensional density along $\mathbf{w}$ is non-Gaussian. Therefore, for some one-dimensional function ψ_k,

$$s_k^{\text{plant}}(\mathbf{x}) = -\mathbf{x} + \psi_k(\langle \mathbf{w}, \mathbf{x} \rangle)\mathbf{w}.$$

Thus all information distinguishing the pancakes from a standard Gaussian is contained in the correction along $\mathbf{w}$.

Given hCLWE samples, train the assumed score estimator and evaluate it on fresh Gaussian points, comparing its output with the known Gaussian score $-\mathbf{x}$. Under the Gaussian case, accuracy makes these deviations small.

Suppose they were also small in the planted case at every required noise level. The same estimated scores would then be accurate both along the pancake path and along the Gaussian path, which remains Gaussian under forward diffusion. By Theorem 1, the same reverse procedure would sample close to both distributions, contradicting the separation witnessed by $T_{\mathbf{w}}$. Hence some noise level has a detectable deviation, giving a polynomial-time distinguisher for hCLWE. □

Polynomially many samples suffice statistically: an exhaustive search over directions can recover $\mathbf{w}$, though it takes exponential time. Once $\mathbf{w}$ is known, the score reduces to estimating the one-dimensional correction ψ_k along $\mathbf{w}$. Thus the samples contain enough information to estimate the score; the obstruction is computational efficiency.

3 Historical notes

Subsection 1.1 is a discrete-time version of the Ornstein–Uhlenbeck process [1]. Diffusion models were introduced by Sohl-Dickstein et al. [2]; DDPMs gave the now-standard discrete formulation [3], while score-based models expressed generation through the reverse-time dynamics and the score [4]. The learning objective in Subsection 2.1 is closely related to denoising score matching, whose connection to score estimation goes back to Vincent [5].

The quantitative reduction from accurate score estimation to sampling in Theorem 1 follows Chen et al. [6]. The hardness result in Theorem 2 is due to Song [7] and builds on the hCLWE hardness of Bruna et al. [8]. The same hidden-direction geometry appeared in Chapter 2: the distribution is Gaussian orthogonal to $\mathbf{w}$ and periodic only in the scalar coordinate along $\mathbf{w}$.

References

[1] G. E. Uhlenbeck and L. S. Ornstein. On the Theory of the Brownian Motion. *Physical Review*, 36(5):823–841, 1930.

[2] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *ICML*, pages 2256–2265, 2015.

[3] J. Ho, A. Jain, and P. Abbeel. Denoising Diffusion Probabilistic Models. In *NeurIPS*, 2020.

[4] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *ICLR*, 2021.

[5] P. Vincent. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[6] S. Chen, S. Chewi, J. Li, Y. Li, A. Salim, and A. R. Zhang. Sampling Is as Easy as Learning the Score: Theory for Diffusion Models with Minimal Data Assumptions. In *ICLR*, 2023.

[7] M. J. Song. Cryptographic Hardness of Score Estimation. In *NeurIPS*, 2024.

[8] J. Bruna, O. Regev, M. J. Song, and Y. Tang. Continuous LWE. In *STOC*, pages 694–707, 2021.