Analysis and Application of Markov Chains in Diffusion models

Garvit Chittora
2021B4A71876G

Jinam Keniya
2021A7PS1622G

November 26, 2024

Abstract

Diffusion models [2] represent a breakthrough in generative modeling, achieving remarkable results across various domains, particularly in image generation and inpainting. These models leverage the principles of Markov chains to construct a probabilistic framework that is both flexible and mathematically elegant. The forward diffusion process systematically corrupts data by progressively adding noise over multiple steps, transforming the original data distribution into a simple, tractable prior distribution, often Gaussian. The reverse diffusion process, learned through a parameterized neural network, iteratively denoises samples, reconstructing data from noise with high fidelity. This report delves into the theoretical underpinnings of diffusion models, detailing the stochastic processes and variational objectives that govern their behavior. Furthermore, we explore practical applications such as inpainting, where missing or corrupted regions in images are reconstructed seamlessly, and extend the discussion to D3PM (Discrete Denoising Diffusion Probabilistic Models), which generalize the framework to discrete data domains.

1 Introduction

The forward process in diffusion models gradually corrupts data x_0 into pure noise x_T by adding small amounts of Gaussian noise over time. This process can be represented by the Itô stochastic differential equation:

$$dx = f(x,t)dt + g(t)dW,$$

where:

- x: State variable (e.g., an image or other data) at time t.

- $f(x,t)$: Drift term (controls deterministic evolution, often set to zero in simple models).
- $g(t)$: Diffusion coefficient (scales the noise added at each step).
- dW: Wiener process (Brownian motion), representing stochastic noise.

For common implementations like DDPMs (Denoising Diffusion Probabilistic Models), the drift term $f(x,t)$ is often zero, simplifying the forward SDE to:

$$dx = g(t)dW.$$

This means the forward process only involves adding Gaussian noise with time-dependent variance.

To generate data, the **reverse diffusion process** undoes the noise, recovering the original data structure. The reverse SDE is given by:

$$dx = \left[f(x,t) - g(t)^2 \nabla_x \log p_t(x)\right] dt + g(t)d\widetilde{W},$$

where $p_t(x)$ is the probability density of x at time t, and $d\widetilde{W}$ represents reverse-time noise. Learning this reverse process involves estimating the gradient $\nabla_x \log p_t(x)$, often referred to as the **score function**.

2 Simple Diffusion

2.1 Forward Process ($\mathbf{L}_T$)

The forward diffusion process begins with a clean data sample $\mathbf{x}_0$ and gradually adds Gaussian noise over T timesteps. The transition from one timestep to the next can be expressed as:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t, \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

$$= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon$$

$$= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon$$

$$= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon$$

$$= \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}}x_{t-3} + \sqrt{1-\alpha_t\alpha_{t-1}\alpha_{t-2}}\epsilon$$

$$= \sqrt{\alpha_t \alpha_{t-1} \dots \alpha_0} x_0 + \sqrt{1 - \alpha_t \alpha_{t-1} \dots \alpha_0} \epsilon$$

$$= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I)$$

where β_t is the noise adding schedule and α_t = 1 - β_t.

The variances β_t (noise levels) are fixed and predefined, simplifying the forward process. This choice ensures L_T, the KL divergence term between $q(x_T)$ and the Gaussian prior, becomes constant and does not need optimization during training.

2.2 Reverse Process ($\mathbf{L}_{1:T-1}$)

The reverse diffusion process aims to recover the original data from noise, modeled as:

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$$

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t),$$

Here, μ_θ and Σ_θ are learned parameters from a neural network that approximate the mean and variance needed to reverse the noise addition.

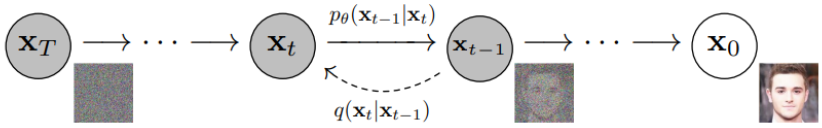

Figure 1: The directed graphical model

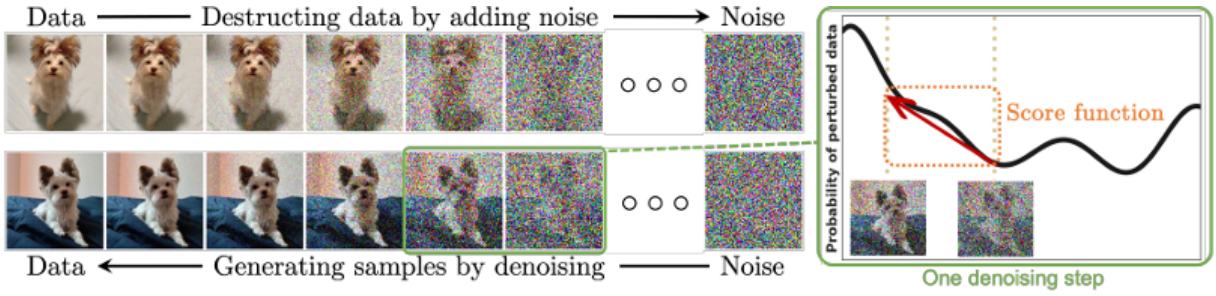

Figure 2: Adding and generating noise to create a new image through diffusion

2.3 β_t noise scheduler

The beta scheduler controls the amount of noise added at each timestep during the forward diffusion process. The process involves gradually adding noise to the data, and the beta schedule defines how this noise evolves over time. The value of β_t at each timestep controls the variance of the noise added in that step. A larger β_t means more noise is added at that timestep.

2.3.1 Linear

The noise increases linearly over time, meaning β_t grows at a constant rate.

$$\beta_t = \beta_0 + (\beta_\mathrm{T} - \beta_0) \cdot \left(\frac{t}{T}\right)$$

Figure 3: Linear Scheduler

2.3.2 Cosine

This schedule is based on the cosine function and is known to work well for generating high-quality images. It starts with a small amount of noise and increases more quickly towards the end.

$$\bar{\alpha}_t = \frac{f(t)}{f(0)} \quad \text{where} \quad f(t) = \cos\left(\frac{\frac{t}{T} + s}{1+s} \cdot \frac{\pi}{2}\right)^2$$

Variance values can then be computed using α_t as follows:

$$\beta_t = 1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_t - 1}$$

The result of the cosine-beta schedule is that it avoids adding too little noise in the beginning and too much noise at the end, while still allowing a considerable increase in the noise in the middle. This ensures that samples at any timestep are equally valuable to the training process.

2.3.3 Quadratic

The noise increases quadratically over time, meaning it accelerates as t increases.

$$\beta_t = (\sqrt{\beta_0} + (\sqrt{\beta_\mathrm{T}} - \sqrt{\beta_0}) \cdot \left(\frac{t}{T}\right))^2$$

2.3.4 Sigmoid

The noise follows a sigmoid function, starting with very little noise, ramping up rapidly in the middle, and tapering off towards the end.

$$\beta_t = \frac{1}{1+e^{-b_t}} \cdot (\beta_{\text{end}} - \beta_{\text{start}}) + \beta_{\text{start}}$$
$$\text{where } b_t = \text{linspace}(-6, 6, T)[t].$$

3 Inpainting with Stable Diffusion

Inpainting replaces or edits specific areas of an image. This makes it a useful tool for image restoration like removing defects and artifacts, or even replacing an image area with something entirely new. Inpainting relies on a mask to determine which regions of an image to fill in; the area to inpaint is represented by white pixels and the area to keep is represented by black pixels. The white pixels are filled in by the prompt. The prompt given here and onwards will be - *"concept art digital painting of an elven castle, inspired by lord of the rings, highly detailed, 8k"*

Figure 4: Image Inpainting

3.1 Varying Strengths

Strength is a measure of how much noise is added to the base image, which influences how similar the output is to the base image.

- A high strength value means more noise is added to an image and the denoising process takes longer, but you will get higher quality images that are more different from the base image
- A low strength value means less noise is added to an image and the denoising process is faster, but the image quality may not be as great and the generated image resembles the base image more

Figure 5: Variation in Strength.

3.2 Varying Guidance Scale

Guidance Scale affects how aligned the text prompt and generated image are.

- A high guidance scale value means the prompt and generated image are closely aligned, so the output is a stricter interpretation of the prompt.
- A low guidance scale value means the prompt and generated image are more loosely aligned, so the output may be more varied from the prompt

Figure 6: Variation in Guidance Scale.

4 D3PM

D3PM - Discrete Denoising Diffusion Probabilistic Models [1] extend diffusion models to discrete state spaces by modeling the forward process using transition matrices Q_t, and

categorical distributions. It provides a framework for applying diffusion processes to categorical variables (e.g., text or discrete images). An advantage of the D3PM framework described above is the ability to control the data corruption and denoising process by choosing Q_t, in notable contrast to continuous diffusion, for which only additive Gaussian noise has received significant attention.

$$q(x_t \mid x_{t-1}) = \mathrm{Cat}(x_t; \mathbf{p} = x_{t-1}\mathbf{Q}_t),$$

where $\mathrm{Cat}(\boldsymbol{x};\, \boldsymbol{p})$ is a categorical distribution over the one-hot row vector $\boldsymbol{x}$ with probabilities given by the row vector $\boldsymbol{p}$, and $\boldsymbol{x}_{t-1}\boldsymbol{Q}_t$ is to be understood as a row vector-matrix product

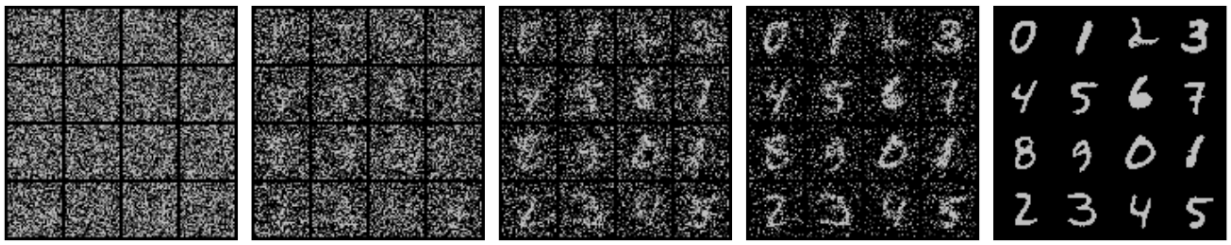

Figure 7: Denoising to generate images

5 Conclusion

In this report, we have explored the theoretical foundation behind diffusion models, which have emerged as a powerful class of generative models. By systematically introducing noise in a controlled manner over multiple timesteps, diffusion models learn to reverse this process and generate data from random noise. This approach contrasts with traditional generative models by focusing on the gradual transformation of data rather than learning a direct mapping from noise to data.

Through the example of Inpainting, we demonstrated how diffusion models can be applied to practical tasks, where the model effectively learns to generate missing parts of an image by iteratively denoising a partially corrupted input. The use of various beta schedulers in the diffusion process allows for flexible control over the rate at which noise is added and removed, influencing the model's ability to generate high-quality inpainted results.

The effectiveness of diffusion models in image generation and manipulation tasks suggest their potential for broader applications in areas such as video generation, super-resolution, and style transfer.

References

[1] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.