Lifting Discrete Diffusion to Continuous Spaces

This document is intended to outline the way I think our work should be framed for the upcoming workshop paper.

Motivation

Discrete denoising diffusion models (D3M) scales poorly with the number of categories. D3M works by representing categories as one hot vectors, that are noised in discrete time steps by some transition matrix Q_t s.t. $q(x_t|x_{t-1}) = Q_t x_{t-1}$ where x is one-hot (see here for more details).

There are two problems with this method:

1. The predicted denoising $p_\theta(x_{t-1}|x_t)$ is in $\mathbb{R}^k$ where k is the number of categories
2. Storing the transition matricies Q_t scales as $\mathcal{O}(Tk^2)$ where T is the number of time steps
    - There are some ways to reduce this

As an example, LLM's use on the order of ten thousand tokens. In this case $p_\theta(x_{t-1}|x_t)$ would need to be modelled by a neural network that has input and output of $k = 10000$ which is inpractical to impliment.

Our Proposal

By "lifting" the discrete diffusion process to a continuous space, we can reduce the model output to $\log_2(k)$ and remove the need to store large transition matricies Q_T

We also show that a by-product of our methodology allows us to naturally extend to simplex diffusion in a simple and easy to implement manner.

Method

Categorical data can be represented as binary strings, which are geometrically the corners of a hyper-cube. Our approach is to lift the discrete problem to the continuous hyper-cube where we can apply the score matching methodology. With this approach, we have representations of dimension $\log_2(k)$ (rather than k) and don't require the use of large transition matricies Q_t.

For a review of relevent material see the sections and the bottom of the doc for score-matching and the OU-process

Diffusion on the Hyper-Cube

Taking the element-wise sigmoid of an OU-process X_t is a convenient way to do diffusion in the hyper-cube. The process we takes the form of

$$C_t = \sigma(X_t)$$

In our case we are able to get an exact solution for C_t by pushing forward the solution of the OU process, meaning that $C_t \sim \sigma\left[\mathcal{N}\left(X_0 e^{-\theta t}, \frac{1}{2\theta}\left(1 - e^{-2\theta t}\right)\right)\right]$. At each point t the resulting distribution is a product of independent logit-normal distributions. We have derived $p_{0t}(\mathbf{x}_t|\mathbf{x}_0)$ and $\nabla_x \log[p_{0t}(\mathbf{x}_t|\mathbf{x}_0)]$ already, as this is just the 1-dimensional case of the simplex math we have previously worked out.

Some Practical Notes

The forward process would take a distribution beginning at the corners of the hyper-cube at $t = 0$ to the center of the cube at $t - 1$. In practice we would represent categorical variables as relaxed binary strings where $0 \to 1 - \alpha$ and $1 \to 1 - \alpha$ for numerical stability (the logit-normal is a dirac delta at the corners of the hyper-cube).

We are interested in implimenting some form of error correcting codes s.t. when we are sampling from a trained model we can correct / reject bad samples. Another potential idea is to enforce structure into the map from categories to cube corners s.t. the hamming distance is similar to some notation of difference is the categories we want to model.

A Concern

This idea could slip into an interpretation where it is similar to using random / learned embeddings (the way some text-based diffusion model work).

Appendix

Score-matching Recap

Score matching considers a continuous time diffusion process. In the forward process, data is perturbed using a stochastic differential equation (SDE) of the following form:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{G}(\mathbf{x}, t)d\mathbf{w}.$$

where $\mathbf{w}$ is the standard Weiner process (also know as Brownian motion), $\mathbf{f}(\mathbf{x}, t)$ is called the drift term and $g(t)$ is called the diffusion coefficient. This process maps our data distribution, $p_{t=0}(\mathbf{x})$ into some limiting distribution $p_{t=1}(\mathbf{x})$. The reverse solution to this SDE is a generative model and is given by

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \mathbf{G}(\mathbf{x}, t)\mathbf{G}(\mathbf{x}, t)^\top \nabla_x \log\, p_t(\mathbf{x})\right] dt + \mathbf{G}(\mathbf{x}, t)d\mathbf{w}$$

where time now flow backwards from $t = 1$ to $t = 0$. Given this equation, we then seek to approximate the score $\nabla_x \log p_t(\mathbf{x})$ to create a generative model. The score can be approximated by a score based model $\mathbf{s}_\theta(\mathbf{x}, t)$ which can be optimized via

$$\theta^* = \text{argmin}_\theta \mathbb{E}_{t\sim U[0,1]} \mathbb{E}_{x_0 \sim p_0(x)} \mathbb{E}_{x_t \sim p_{0t}(x_t|x_0)} \lambda(t) \left[\|\mathbf{s}_\theta(\mathbf{x}, t) - \nabla_{x_t} \log p_{0t}(\mathbf{x}_t|\mathbf{x}_0)\|_2^2 \right]$$

Thus, when making a type of score matching procedure, we need:
1. To sample from $p_{0t}(\mathbf{x}_t|\mathbf{x}_0)$ (which comes from the forward process) 2. The grad log of 1.

The Ornstein-Unlenback Process

The Ornstein-Unlenbeck (OU) process is real-valued stochastic process used in financial mathematics and physical sciences. Originally, it was developed to model the velocity of a Brownian particle under the force of friction. The process can be described by the following stochastic differential equation:

$$dx_t = -\theta x_t dt + \sigma dW_t$$

where $\theta > 0$ and $\sigma > 0$ are parameters and W_t is the Weiner process. The process had the exact solution

$$O_t := \mathcal{N}\left(O_0 e^{-\theta t}, \frac{1}{2\theta}\left(1 - e^{-2\theta t}\right) \right).$$

In the limit as $t \to \infty$ the process has a distribution of $\mathcal{N}\left(0, \frac{1}{2\theta}\right)$, meaning that θ uniquely determines the limiting distribution.