Basics of Score Based Models

July 30, 2022

Useful papers - [SE19],

1 Introduction

As always in generative modelling, we aim to recover distribution of data $p_{data}(x)$ given samples from it: $X = \{x_1, ..., x_N\}$. We build some parameterized model $p_\theta(x)$ and want to find θ^* such that $p_{\theta^*} = p_{data}, \forall x$. Here we will consider an approach to solving this problem which is based on Score Estimation. The main idea consists in approximation the gradient of data distribution, not the distribution proper:

$$\begin{aligned} \mathbb{F}(p_\theta||p_{data}) = \frac{1}{2}\mathbb{E}_{p_{data}(x)}\,||\nabla_x \log p_\theta(x) - \nabla_x \log p_{data}(x)||^2 \\ \theta^* = \arg\min_\theta \mathbb{F}(p_\theta||p_{data}) - ? \end{aligned} \tag{1}$$

Here $\mathbb{F}$ - Fisher divergence ($\mathbb{F} \geq 0$), $\nabla_x \log p_{data}(x), \nabla_x \log p_\theta(x)$ - score of data and model distributions. In Score Based models we define score of model distribution as neural network, i.e. $\nabla_x \log p_\theta(x) = \mathbf{nn}_\theta(x), \mathbf{nn}_\theta : \mathbb{R}^d \to \mathbb{R}^d$. So, here we do Score Estimation (approximation of $\nabla_x \log p_{data}(x)$) by Score Matching (minimization of the F-divergence).

A fair question may arise, lets say we found score of the data distribution, how it help us to solve our main problem (recover p_{data})? Lets discuss this question, so we found θ^* such that $\mathbb{F}(p_{\theta^*}||p_{data}) = 0$, that is

$$\begin{aligned} 0 = \mathbb{E}_{p_{data}(x)}\,||\mathbf{nn}_\theta(x) - \nabla_x \log p_{data}(x)||^2 \\ = \int dx p_{data}(x)||\mathbf{nn}_\theta(x) - \nabla_x \log p_{data}(x)||^2 \end{aligned} \tag{2}$$

Let $p_{data}(x) > 0, \forall x$, then $\mathbf{nn}_\theta = \nabla_x \log p_{data}$ and $p_\theta = p_{data}$. We also might have situations where for some x $p_{data}(x) = 0$. In these cases we cannot guarantee that $\mathbf{nn}_\theta = \nabla_x \log p_{data}, \forall x$ and so $p_\theta \neq p_{data}$. So, lets remember that we require that $p_{data}(x) > 0, \forall x$. More formally, we need that **support of the data distribution should equal the entire space**.

It is possible to ask one more question: is the gradient of the data distribution, $\nabla_x \log p_{data}(x)$, exists? There are two reasons that the answer is no: 1) in case of images the data is discrete, so there is no gradient; 2) because of the manifold hypothesis (data in the real world tend to concentrate on low dimensional manifolds embedded

in a high dimensional space) there is no gradients. I think that the second reason should be discussed in more detail. Lets look at an example of function embedded in two dimensional space but lies in one dimensional manifold (Figure 1). Here we have

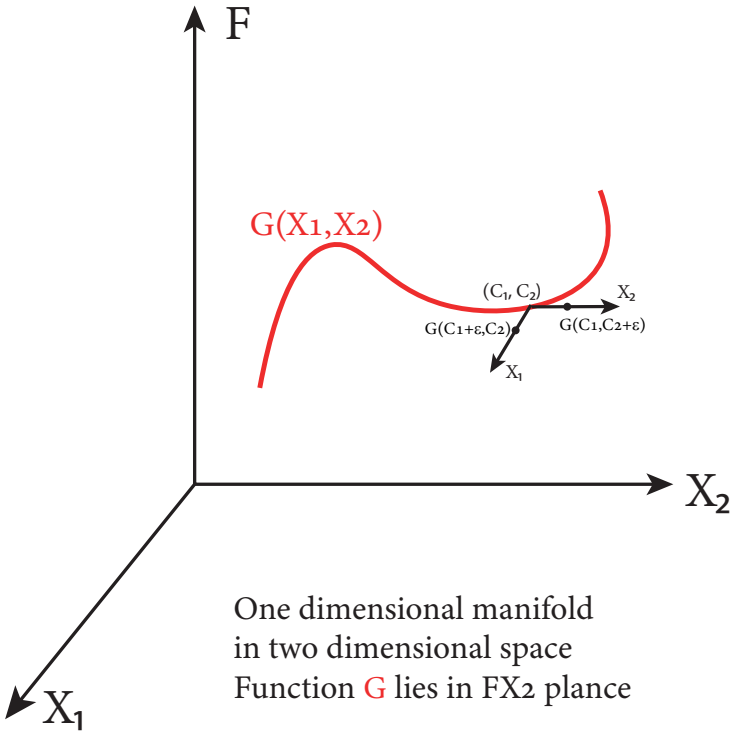

Figure 1: Derivative of function embedded in two dimensional space but lies in one dimensional manifold

function $G(X_1, X_2)$, however it does not depend from X_1. Lets take gradient of the $G(X_1, X_2)$ in point C_1, C_2.

$$\nabla_X G(X_1, X_2) = \begin{pmatrix} \frac{\partial G}{\partial X_1}(C_1, C_2) \\ \frac{\partial G}{\partial X_2}(C_1, C_2) \end{pmatrix} = \begin{pmatrix} \lim_{\epsilon \to 0} \frac{G(C_1+\epsilon, C_2) - G(C_1, C_2)}{\epsilon} \\ \lim_{\epsilon \to 0} \frac{G(C_1, C_2+\epsilon) - G(C_1, C_2)}{\epsilon} \end{pmatrix} \tag{3}$$

We have a problem when taking gradient. Because our function G lies in plane FX_2 (one dimensional manifold), it is not defined in point $C_1 + \epsilon$ for arbitrary small ϵ. So, we cannot take the gradient. I think this is absolutely shitty proof. Anyway, we have the second requirement: **the gradient of the data distribution should exist**.

It is possible to show (see notes about EBM) that

$$\begin{aligned} &\mathbb{E}_{p_{data}(x)} ||\nabla_x \log p_\theta(x) - \nabla_x \log p_{data}(x)||^2 \\ &= \mathbb{E}_{p_{data}(x)} \left[\mathbf{tr}(\nabla_x \mathbf{nn}_\theta(x)) + \frac{1}{2} ||\mathbf{nn}_\theta(x)||^2 \right] + C \end{aligned} \tag{4}$$

where $\nabla_x \mathbf{nn}_\theta(x)$ - Jacobian of $\mathbf{nn}_\theta(x)$, i.e. $\nabla_x \mathbf{nn}_\theta(x) \in \mathbb{R}^{d \times d}$. It is extremely time-consuming for calculation (see details in notes about EBM). It is possible to solve this problem and also satisfy two mentioned requirements: lets add small noise to data :). The first requirement is satisfied, since the entire space is the support for the noise

distribution. The second requirement is satisfied because noised distribution has the same dimension as space in which it is located, so the gradient exist. Finally, lets show how time-consuming problem is solved. It is possible to show that we need to optimize following objective (**Denoising Score Matching**)

$$\frac{1}{2}\,\mathbb{E}_{q_\sigma(\hat{x}|x)p_{data}(x)}\left[||\mathbf{nn}_\theta(\hat{x}) - \nabla_{\hat{x}}\log q_\sigma(\hat{x}|x)||^2\right] \tag{5}$$

where $q_\sigma(\hat{x}|x)-$ transition distribution between real sample x and noised $\hat{x}$, $\sigma-$ magnitude of noise. Usually, it takes as normal $q_\sigma(\hat{x}|x) = \frac{1}{(2\pi)^{\frac{d}{2}}\sigma}\exp\left(-\frac{1}{2}\frac{||\hat{x}-x||^2}{\sigma^2}\right)$. Here we approximate $q_\sigma(x)$ (distribution of noised samples), thus we need to take small magnitude σ such that $q_\sigma(x) \approx p_{data}(x)$. But we solved our problems. For practise realization we will rewrite formula as follows:

$$\begin{gathered}
\nabla_{\hat{x}}\log q_\sigma(\hat{x}|x) = \frac{1}{\sigma^2}(x-\hat{x}), \hat{x} = x + \sigma z, z \sim \mathbb{N}(0, I) \\
\frac{1}{2}\,\mathbb{E}_{q_\sigma(\hat{x}|x)p_{data}(x)}\left[||\mathbf{nn}_\theta(\hat{x}) - \nabla_{\hat{x}}\log q_\sigma(\hat{x}|x)||^2\right] \\
= \frac{1}{2}\,\mathbb{E}_{q_\sigma(\hat{x}|x)p_{data}(x)}\left[||\mathbf{nn}_\theta(\hat{x}) + \frac{1}{\sigma^2}(\hat{x}-x)||^2\right] \\
= \frac{1}{2}\,\mathbb{E}_{z\sim\mathbb{N}(0,I)p_{data}(x)}\left[||\mathbf{nn}_\theta(x+\sigma z) + \frac{1}{\sigma}z||^2\right] \\
\approx \frac{1}{2N}\sum_{j=1}^{N}||\mathbf{nn}_\theta(x_j + \sigma z_j) + \frac{1}{\sigma}z_j||^2, x_j \sim p_{data}(x), z_j \sim \mathbb{N}(z|0, I)
\end{gathered} \tag{6}$$

It is possible to consider another objective called **Sliced Score Matching**:

$$\begin{gathered}
\mathbb{E}_{p(v)}\,\mathbb{E}_{p_{data}(\hat{x})}\left[v^T\cdot\nabla_{\hat{x}}\left(\mathbf{nn}_\theta(\hat{x})\cdot v\right) + \frac{1}{2}||\mathbf{nn}_\theta(\hat{x})||^2\right] \\
\approx \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[v_{ij}^T\cdot\nabla_{\hat{x}_i}\left(\mathbf{nn}_\theta(\hat{x}_i)\cdot v_{ij}\right) + \frac{1}{2}||\mathbf{nn}_\theta(\hat{x}_i)||^2\right] \\
v_{ij} \sim \mathbb{N}(v|0, I), \hat{x}_i = x_i + \sigma z_i, x_i \sim p_{data}(x), z_i \sim \mathbb{N}(z|0, I)
\end{gathered} \tag{7}$$

where $v-$ random projection, $p(v) = \mathbb{N}(v|0, I)$, $\hat{x}-$ noised sample, sign $\cdot$ means scalar product. For each x_i we sample M random projections. This approach more time-consuming compared to DSM.

So, we have achieved two objectives for estimation of the score. In other words, we can obtain $\mathbf{nn}_\theta$. To obtain samples from the distribution we can use Langevin dynamics.

$$\begin{gathered}
\hat{x}_t = \hat{x}_{t-1} + \frac{\epsilon}{2}\mathbf{nn}_\theta(\hat{x}_{t-1}) + \sqrt{\epsilon}z_t \\
\hat{x}_0 \sim \mathbb{U}(0, 1)
\end{gathered} \tag{8}$$

ϵ is small, $z_t \sim \mathbb{N}(z|0, I)$. But we can meet two problems with this approach.

1. In Langevin dynamics we always start with noise sample ($\hat{x}_0 \sim \mathbb{U}(0, 1)$). It is clear that in such points estimated score $\mathbf{nn}_\theta(\hat{x}_0)$ will be inaccurate, that is, very

different from the real one, $\nabla_{\hat{x}} \log p(\hat{x}_0)$. This follows from the fact that in training stage we have never seen such samples, because our training examples is far from complete noise. They are noised but magnitude of noise is small. So, because of the inaccurate score in noise examples, Langevin dynamics will have problems with convergence. Lets summarize.
Problem: inaccurate estimation of the score for samples that are far from real.
Consequence: wrong convergence of the Langevin dynamics.
Solution: add noised samples (with high and medium magnitude of the noise) on training stage.

2. Langevin dynamics do not take into account weights of disjoint modes. If $p(x) = w_1 p_1(x) + w_2 p_2(x)$. Then $\nabla_x \log p(x) = \nabla_x \log p_1(x), x \in \mathbf{supp}(p_1)$. It can be clearly seen from Figure 2. Lets summarize.

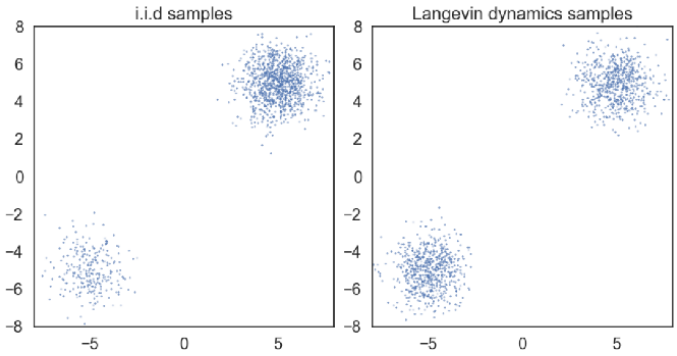

Figure 2: Problem of weight of modes in Langevin dynamics.

Problem: Langevin dynamics do not take into account weights of disjoint modes.
Consequence: wrong convergence of the Langevin dynamics.
Solution: make annealed version of the Langevin dynamics with different levels of noise. I cannot understand why it helps :(

Okay, what we have now. Firstly, we need to add small noise to our data to satisfy two mentioned requirements (support of the data distribution should equal the entire space and the gradient of the data distribution should exis) and solve problem of expensive computation of trace in Score Matching. Secondly, we also need to add different levels of noise to our data to solve two problems of the Langevin dynamics.

So, we add magnitude of noise as condition in our score, i.e. $\mathbf{nn}_\theta(x) \to \mathbf{nn}_\theta(x, \sigma)$. Thus, we modify our objectives as follows.
Denoising Score Matching:

$$\begin{aligned} \mathbf{l}_{dsm}(\theta, \sigma) &= \frac{1}{2} \mathbb{E}_{z \sim \mathbb{N}(0,I) p_{data}(x)} \left[||\mathbf{nn}_\theta(x + \sigma z, \sigma) + \frac{1}{\sigma} z||^2 \right] \\ &\approx \frac{1}{2N} \sum_{j=1}^{N} ||\mathbf{nn}_\theta(x_j + \sigma z_j, \sigma) + \frac{1}{\sigma} z_j||^2, x_j \sim p_{data}(x), z_j \sim \mathbb{N}(z|0, I) \end{aligned} \tag{9}$$

Sliced Score Matching:

$$\begin{aligned}\mathbf{l}_{ssm}(\theta,\sigma) &= \mathbb{E}_{p(v)}\,\mathbb{E}_{p_{data}(\hat{x})}\left[v^T\cdot\nabla_{\hat{x}}\left(\mathbf{nn}_\theta(\hat{x},\sigma)\cdot v\right)+\frac{1}{2}||\mathbf{nn}_\theta(\hat{x},\sigma)||^2\right]\\ &\approx \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[v_{ij}^T\cdot\nabla_{\hat{x}_i,\sigma}\left(\mathbf{nn}_\theta(\hat{x}_i,\sigma)\cdot v_{ij}\right)+\frac{1}{2}||\mathbf{nn}_\theta(\hat{x}_i,\sigma)||^2\right]\\ v_{ij}&\sim\mathbb{N}(v|0,I),\hat{x}_i=x_i+\sigma z_i, x_i\sim p_{data}(x), z_i\sim\mathbb{N}(z|0,I)\end{aligned}\tag{10}$$

Final objective has the following form:

$$\mathbb{L}(\theta,\{\sigma_i\}_{i=1}^L)=\frac{1}{L}\sum_{i=1}^{L}\lambda(\sigma_i)\mathbf{l}_{dsm}(\theta,\sigma_i)\tag{11}$$

It is called Noise Conditional Score Networks. Note that $\frac{\sigma_1}{\sigma_2}=...=\frac{\sigma_{L-1}}{\sigma_L}>1$, $\lambda(\sigma_i)=\sigma_i^2$. As sampling procedure authors of the original paper proposed 3:

Algorithm 1 Annealed Langevin dynamics.

Require: $\{\sigma_i\}_{i=1}^L, \epsilon, T$.
1: Initialize $\tilde{\mathbf{x}}_0$
2: **for** $i \leftarrow 1$ to L **do**
3: $\quad\alpha_i \leftarrow \epsilon\cdot\sigma_i^2/\sigma_L^2$ ▷ α_i is the step size.
4: $\quad$**for** $t \leftarrow 1$ to T **do**
5: $\quad\quad$Draw $\mathbf{z}_t\sim\mathcal{N}(0,I)$
6: $\quad\quad\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1}+\frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1},\sigma_i)+\sqrt{\alpha_i}\,\mathbf{z}_t$
7: $\quad$**end for**
8: $\quad\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
9: **end for**
return $\tilde{\mathbf{x}}_T$

Figure 3: Annealed Langevin dynamics

Where $L=10, T=100, \sigma_1=10, \sigma_{10}=0.01, \epsilon=2\times10^{-5}$.

References

[SE19] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.