Conditional sampling in diffusion generative models

Tamás Papp
STOR-i CDT, Lancaster University

Diffusion model

Diffusion model defined as joint distribution $p_{T:0}(Z_T, Z_{T-\varepsilon}, \dots, Z_\varepsilon Z_0)$, where

- **Forward** "noising" kernels $p_{t|t-\varepsilon}(Z_t \mid Z_{t-\varepsilon})$ e.g. $\mathcal{N}(Z_t \mid e^{-\varepsilon} Z_{t-\varepsilon}, 1 - e^{-2\varepsilon})$.
- **Backward** "denoising" kernels $p_{t-\varepsilon|t}(Z_{t-\varepsilon} \mid Z_t)$ e.g. $\mathcal{N}(Z_{t-\varepsilon} \mid \varepsilon(Z_t + 2\nabla \log p_t(Z_t)), 2\varepsilon)$.
- **Data** distribution $p_0(Z_0)$, of interest.
- **Noise** distribution $p_T(Z_T)$ e.g. $\approx \mathcal{N}(0, I)$.

For this talk, I will assume that the model is exact, i.e. that

1. No error in estimating the score.
2. No discretization error.
3. $p_{t,t-1}(Z_t, Z_{t-1}) = p_{t|t-1}(Z_t \mid Z_{t-1}) p_{t-1}(Z_{t-1}) = p_{t-1|t}(Z_{t-1} \mid Z_t) p_t(Z_t)$ for all $t \geq 0$.

Inpainting and key insight

Diffusion model offers a model for the marginal $p_0(Z_0)$.

Inpainting: If the state is $Z_0 = [X_0, Y_0]$ and I have observed Y_0, **can I sample** $X_0 \mid Y_0$?

Aim: Want to sample from the conditional $p_0(X_0 \mid Y_0)$ **without additional training**. Morally, if I have modelled the joint $p_0(X_0, Y_0)$, then I have also implicitly modelled the conditional $p_0(X_0 \mid Y_0)$.

Insight: to do so consistently, exploit various model factorizations.

The replacement method

Replacement method

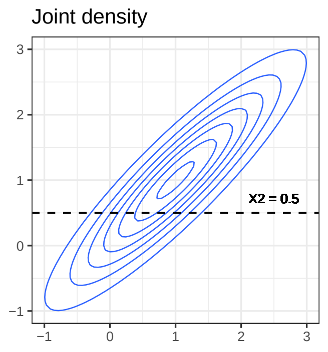
Algorithm 1: Replacement method

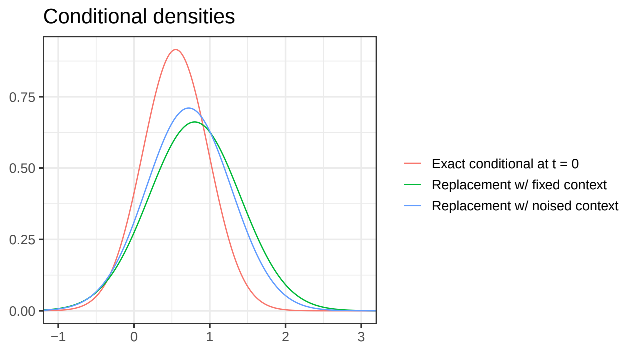
1. Draw a path $Y_\varepsilon, \ldots, Y_T$.
2. Draw $X_T \sim p_T\,(X_T \mid Y_T)$.
3. For $t = T, T-\varepsilon, \ldots, \varepsilon$:
    - Sample $X_{t-\varepsilon}, \sim p_{t-\varepsilon|t}\,(X_{t-\varepsilon} \mid X_t, Y_t)$.
4. Retain X_0.

For example, the "context" path could be chosen as:

- **Fixed**: $Y_t = Y_0$ for all t.
- A **path of the forward process**: $Y_{T:\varepsilon} \sim p_{T:1|0}\,(Y_{T:\varepsilon} \mid Y_0)$.

Inconsistency of replacement method





Replacement method is **inconsistent**:

- Conditioning information is too weak at each time-step.
- Method cannot be exact even if there is no score or discretization error.

Correcting with Langevin steps

Langevin corrector

Fix the time $t = 0$. Because

$$\nabla_{X_0} \log p_0 (X_0, Y_0) = \nabla_{X_0} \log p_0 (X_0 \mid Y_0) + \nabla_{X_0} \log p_0 (Y_0) = \nabla_{X_0} \log p_0 (X_0 \mid Y_0),$$

in principle, we could sample X_0 from the conditional $p_0(X_0 \mid Y_0)$ by iterating Langevin dynamics

$$X_0 \leftarrow X_0 + \varepsilon \nabla_{X_0} \log p_0 (X_0, Y_0) + \sqrt{2\varepsilon} Z, \quad Z \sim \mathcal{N}(0, I).$$

This **only uses the joint score!**

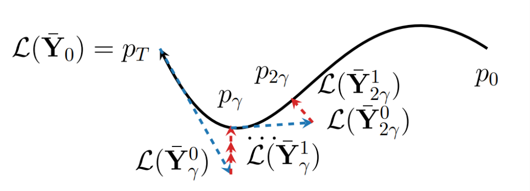
We don't want to do this: in complex problems, at $t = 0$ there is a **large score error** and the **mixing is slow**. (Especially if there are multiple modes.)

Instead, we **apply several Langevin correctors at each time-step** of the replacement method.

Langevin-corrected replacement method

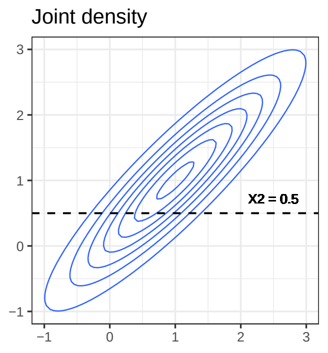
Algorithm 2: Replacement method w/ Langevin corrector

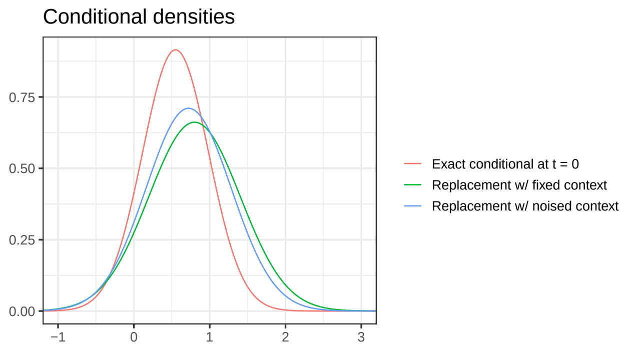
1. Draw a path $Y_\varepsilon, \ldots, Y_T$.
2. Draw $X_T \sim p_T(X_T \mid Y_T)$.
3. For $t = T, T-\varepsilon, \ldots, \varepsilon$:
   - Sample $X_{t-\varepsilon}, \sim p_{t-\varepsilon|t}(X_{t-\varepsilon} \mid X_t, Y_t)$.
   - Update $X_{t-\varepsilon}$ using L steps of Langevin with score $\nabla_{X_{t-\varepsilon}} \log p_{t-\varepsilon}(X_{t-\varepsilon}, Y_{t-\varepsilon})$.
4. Retain X_0.



Consistent if no discretization error and as **number of Langevin steps** $L \to \infty$. Works irrespective of context path.

Consistency of Langevin-corrected replacement method





In practice:

- Always have discretization error.
- With estimated score, the method can diverge when $L \to \infty$.
- Computational cost increases by a factor of L.

Particle filtering

i.e. consistency by importance weighting

Consistency by weighting

The "vanilla" replacement method is inconsistent because it **does not put enough weight** on the conditioning information.

- Suppose that we drew a path $Y_{T:\varepsilon} \sim p_{T:\varepsilon}(Y_{T:\varepsilon} \mid Y_0)$ from the noising process.
- When moving $t \to (t-\varepsilon)$ conditional on this path, we know that we should land the context near $Y_{t-\varepsilon}$.
- Replacement method does not use this information.

Idea: use **multiple particles**, first **weight them according to where they should land**, then **propagate them forward** as in the replacement method.

As it turns out, the right weight is $p_{t-\varepsilon|t}(Y_{t-\varepsilon} \mid X_t, Y_t)$ and we get a **bootstrap particle filter**.

Bootstrap particle filter

Algorithm 3: Bootstrap particle filter (a.k.a. "SMCDiff")

1. Draw a path $Y_{T:\varepsilon} \sim p_{T:\varepsilon}\left(Y_{T:\varepsilon} \mid Y_0\right)$ from the noising process.
2. Draw N samples $X_T^{(1:N)} \sim p_T\left(X_T \mid Y_T\right) \approx N(0, I)$.
3. For $t = T, T-\varepsilon, \ldots, \varepsilon$:
   - **Weight** $w^{(k)} = p\left(Y_{t-\varepsilon} \mid X_t^{(k)}, Y_t\right), \forall k$.
   - **Normalize** weights such that $\sum_k w^{(k)} = 1$.
   - **Resample** particles $X_t^{(1:N)} \leftarrow \text{Resample}\left(X_t^{(1:N)}, w^{(1:N)}\right)$.
   - **Propagate** $X_{t-\varepsilon}^{(k)} \sim p_{t-\varepsilon|t}\left(X_{t-\varepsilon} \mid X_t^{(k)}, Y_t\right), \forall k$.
4. Retain one of the $X_0^{(k)}$.

Consistent if no discretization error and as number of particles $N \to \infty$.

Must run the entire procedure multiple times to obtain i.i.d. samples.

Correctness (i)

Factorization of the model ensures that procedure is correct.

For ease of notation, set $\varepsilon = 1$. Consider the factorization:

$$
\begin{aligned}
p(X_{T:t}, Y_{T:t}) &= p(X_{T:(t+1)}, Y_{T:(t+1)})p(X_t, Y_t \mid X_{T:(t+1)}, Y_{T:(t+1)}) \\
&= p(X_{T:(t+1)}, Y_{T:(t+1)})p(X_t, Y_t \mid X_{t+1}, Y_{t+1}) && \text{(joint is Markov)} \\
&= p(X_{T:(t+1)}, Y_{T:(t+1)}){\color{red}p(Y_t \mid X_{t+1}, Y_{t+1})}{\color{blue}p(X_t \mid X_{t+1}, Y_{t+1})}. && \text{(separable dynamics)}
\end{aligned}
$$

By Bayes' rule,

$$
p(X_{T:t} \mid Y_{T:t}) \propto p(X_{T:(t+1)} \mid Y_{T:(t+1)}){\color{red}p(Y_t \mid X_{t+1}, Y_{t+1})}{\color{blue}p(X_t \mid X_{t+1}, Y_{t+1})}.
$$

Insight: if we sampled from this and only kept the marginal X_t, we would have a sample from $X_t \mid Y_{T:t}$.

Integrating,

$$
p(X_t \mid Y_{T:t}) \propto \int p(X_{t+1}, Y_{T:(t+1)}){\color{red}p(Y_t \mid X_{t+1}, Y_{t+1})}{\color{blue}p(X_t \mid X_{t+1}, Y_{t+1})}\mathrm{d}X_{t+1}.
$$

(Continues on next slide.)

Correctness (ii)

Recall:

$$p(X_t \mid Y_{T:t}) \propto \int p(X_{t+1}, Y_{T:(t+1)}) p(Y_t \mid X_{t+1}, Y_{t+1}) p(X_t \mid X_{t+1}, Y_{t+1}) \mathrm{d}X_{t+1}.$$

So, if we have an approximation

$$p(X_{t+1}, Y_{T:(t+1)}) \approx \sum_{k=1}^{N} \delta_{X_{t+1}^{(k)}},$$

then our approximation to $p(X_t \mid Y_{T:t})$ is

$$p(X_t \mid Y_{T:t}) \approx \sum_{k=1}^{N} w^{(k)} p(X_t \mid X_{t+1}^{(k)}, Y_{t+1}),$$

where $w^{(k)} \propto p(Y_t \mid X_{t+1}^{(k)}, Y_{t+1})$ then normalized.

We sample N particles with equal weight from this by (i) deciding on the mixture component k using $w^{(k)}$, then (ii) sampling from the mixture component.

A plethora of other methods

Inpainting:

- Particle MCMC: use the fact that the particle filter gives an unbiased approximation to the marginal likelihood.

More general conditioning:

- Train the score model to model $\nabla_Z \log p_t(Z_t \mid y)$ directly.
- Train a separate classifier model on $\nabla_Z \log p_t(y \mid Z_t)$ and use that

$$\nabla_Z \log p_t(Z_t \mid Y) = \nabla_Z \log p_t(Z_t) + \nabla_Z \log p_t(y \mid Z_t).$$

- The "guidance" heuristic.