HW7

Problem 1 Consider 1 billion people distributed across three states (1, 2, 3). Let $p(x)$ be the number (in billion) of people in state x, $x \in (1, 2, 3)$. Let $p(y|x)$ be the fraction of those in state x who will move to state y, $y \in (1, 2, 3)$. If we random sample a person, $p(x)$ is the probability that the person is in x, and $p(y|x)$ is the conditional probability that this person will move to y if this person is in x. Let $\tilde{p}(y)$ be the number (in billion) of people who will end up in y, which can again be translated into the probability that the random person ends up in y.

(1) Explain that

$$p(x, y) = p(x)p(y|x),$$

where $p(x, y)$ is the number of people who are in x and who end up in y. This is **chain rule**.

Explain that

$$\tilde{p}(y) = \sum_x p(x, y) = \sum_x p(x)p(y|x).$$

The first equation is the **rule of marginalization**. The last equation is the **rule of total probability**.

(2) Let $p(x|y)$ be the faction of those people in y who come from x. Explain that

$$p(x|y) = \frac{p(x, y)}{\tilde{p}(y)} = \frac{p(x)p(y|x)}{\tilde{p}(y)} = \frac{p(x)p(y|x)}{\sum_x p(x)p(y|x)}.$$

The first equation is the **rule of conditioning**. The final equation is called the **Bayes rule** if we interpret x as cause and y as effect.

As to conditionals, we may call $p(y|x)$ the **forward conditional**, and $p(x|y)$ the **backward conditional**.

The above three rules (chain rule, marginalization, conditioning) underlie all the probability calculations in machine learning.

(3) Suppose for those $\tilde{p}(y)$ billion people in y, we send $p(x|y)$ of them back to x. Explain that the distribution of the 1 billion people will be back to $p(x)$, even though the people in x now may not be the same people who were in x before.

(4) Please illustrate the above using concrete numbers:

State (x)	$p(x)$	Interpretation
1		... billion people
2		
3		

Table 1: Initial distribution

The transition probabilities $p(y|x)$ are:

The inverse transition probabilities $p(x|y)$ are:

Problem 2 For continuous random variable, we have probability = density × size, or density = probability/size. Here the probability can be conditional probability. We can assume the continuous random variables are one-dimensional scalars.

(1) For continuous x and y, we can interpret $p(x)\Delta x$ to be the number of people in $(x, x + \Delta x)$. $p(y|x)\Delta y$ be the proportion of those in $(x, x + \Delta x)$ who will move to $(y, y + \Delta y)$. Then among

$p(y\|x)$	$y=1$	$y=2$	$y=3$
$x=1$			
$x=2$			
$x=3$			

Table 2: Transition probabilities

$p(x\|y)$	$x=1$	$x=2$	$x=3$
$y=1$			
$y=2$			
$y=3$			

Table 3: Inverse transition probabilities

those who end up in $(y, y+\Delta y)$, the proportion of those who come from $(x, x+\Delta x)$ is $p(x|y)\Delta x$. Please show

$$p(x,y) = p(x)p(y|x).$$

$$\tilde{p}(y) = \int p(x,y)dx = \int p(x)p(y|x)dx.$$

$$p(x,y) = p(x)p(y|x) = \tilde{p}(y)p(x|y).$$

$$p(x|y) = \frac{p(x,y)}{\tilde{p}(y)} \propto p(x)p(y|x),$$

where the above proportionality is in terms of functions of x, where y is fixed, and x is the variable of the functions.

(2) Let $p(x)$ be the probability density function of random variable x. Let $y = x + e$, where $e \sim \mathrm{N}(0, \sigma^2)$ for a small σ^2, and e is independent of x, i.e., $p(y|x) \sim \mathrm{N}(x, \sigma^2)$, with

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(y-x)^2\right] \propto \exp\left[-\frac{1}{2\sigma^2}(y-x)^2\right]$$

Please show that for small σ^2, approximately,

$$p(x|y) \sim \mathrm{N}(y + \sigma^2 \nabla \log p(y), \sigma^2),$$

where $\nabla f(x)$ is the derivative (slope, gradient) of the function f at x. You can use the equation

$$p(x|y) \propto p(x)p(y|x),$$

and the first order Taylor expansion of $\log p(x)$ around y.

Problem 3

(1) Let $x_0 \sim p_0(x)$. Let $x_t = x_{t-1} + e_t$, where $e_t \sim \mathrm{N}(0, \sigma^2)$ for a small σ^2, and e_t are independent for different t, $t = 1, ..., T$, so that for large T, approximately $x_T \sim \mathrm{N}(0, T\sigma^2)$. Let p_t be the marginal density of x_t. Based on the previous problem, explain that for small σ^2, approximately,

$$p(x_{t-1} \mid x_t) \sim \mathrm{N}(x_t + \sigma^2 \nabla \log p_{t-1}(x_t), \sigma^2).$$

Explain that we can estimate the score function $\nabla \log p_{t-1}(x_t)$ using a neural network $s_\theta(x_t, t)$ by minimizing the least squares loss

$$L(\theta) = \mathrm{E}_{t,x_0,x_{t-1},x_t}[(x_{t-1} - (x_t + \sigma^2 s_\theta(x_t, t)))^2],$$

where $\mathrm{E}_{t,x_0,x_{t-1},x_t}$ can be approximated by averaging over $t = 1, ..., T$ and (x_0, x_{t-1}, x_t).

(3) After learning $s_\theta(x, t)$, explain that we can generate a new x_0 by sampling $x_T \sim \mathrm{N}(0, T\sigma^2)$, and iterating

$$x_{t-1} = x_t + \sigma^2 s_\theta(x_t, t) + \tilde{e}_t,$$

where $\tilde{e}_t \sim \mathrm{N}(0, \sigma^2)$ independently, for $t = T, ..., 1$. This is the reverse denoising process.

(4) Explain that we can also iterate

$$x_{t-2} = x_t + \sigma^2 s_\theta(x_t, t),$$

which is a deterministic denoising process.

(5) Explain the above in terms of the movements of 1 billion particles on the real line. In particular, in (4), why deterministic movements reverse random noising perturbations n as far as the overall marginal distribution is concerned?

Problem 4 Instead of predicting x_{t-1} from x_t, we can also predict x_0 directly from x_t to learn $s_\theta(x_t, t)$, by minimizing

$$L(\theta) = \mathrm{E}_{x_0,t,x_t}[(x_0 - (x_t + t\sigma^2 s_\theta(x_t, t)))^2],$$

where $x_t = x_0 + \epsilon_t$, $\epsilon_t \sim \mathrm{N}(0, t\sigma^2)$. x_0 is a better target than x_{t-1} because x_0 is the clean version without noise.

Let

$$\epsilon_\theta(x, t) = -t\sigma^2 s_\theta(x, t),$$

we can write

$$L(\theta) = \mathrm{E}_{x_0,t,\epsilon_t}[(\epsilon_t - \epsilon_\theta(x_0 + \epsilon_t, t))^2].$$

That is, we learn ϵ_θ network to estimate the noise ϵ_t from the noisy observation $x_0 + \epsilon_t$.

After estimating $\epsilon_\theta(x, t)$, please derive the backward denoising process, both stochastic version and deterministic version.

Note: In real implementation, people also introduce some scaling coefficients such as $x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{\beta_t} e_t$. But this is less essential.

Problem 5 Play with the PyTorch code provided by the following webpage:

`https://github.com/albarji/toy-diffusion`

Write a brief explanation of the code and show your results.