Laboratorio 4

Modelos basados en *score*

Profesor: Felipe Tobar
Auxiliares: Cristóbal Alcázar, Camilo Carvajal Reyes **Ayudante**: Joaquín Barceló

Fecha de entrega: 6 de noviembre 2023

Instrucciones: El presente laboratorio tiene como objetivo la implementación de algunos elementos que conforman la familia de modelos basados en *score*. El formato de entrega consistirá en un archivo de extensión .ipynb donde contenga el código con sus respuestas tanto de código como teóricas (que deberán presentarse en celdas de *Markdown*). Considere que:

- El código debe estar ordenado de modo que quien lo lea entienda su contenido.
- Las respuestas deben ser precisas y concisas.
- El *notebook* debe ser ejecutable sin errores. Se recomienda verificar esto último reiniciando la *kernel* antes de entregar.

(P1) *Muestreo de Langevin*

El objetivo de esta pregunta será explorar visualmente la naturaleza del muestreo con el método *Langevin dynamics*, que es aquel usado para generar puntos de datos con modelos de *score*. Para esto usaremos mixturas de gaussianas, i.e., modelos que cumplen que la densidad de un punto $x \in \mathbb{R}^d$ está dada por

$$p(x) = \sum_{k=1}^{K} \alpha_k \frac{1}{\sqrt{(2\pi)^d \det(\Sigma_k)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)},$$

donde Σ_k es definida positiva para todo $k = 1, \ldots, K$ y $\sum_{k=1}^{K} \alpha_k = 1$.

a) (1.0 ptos) Defina una clase *GaussianMixture*. Considere que esta debe instanciarse tomando los parámetros de las distribuciones gaussianas que la componen, incluyendo el peso de cada componente. Incluya los métodos que le parezcan adecuados.

b) (1.5 ptos.) Exprese la función de *score* para una mixtura de gaussianas en función de un punto $x \in \mathbb{R}^d$. Implemente un nuevo método para la clase *GaussianMixture* que retorne el vector de *score* en cuestión dado x.

c) (0.5 ptos.) Usando su clase, defina una mixtura de tres Gaussianas en $\mathbb{R}^2$ y visualice tanto su densidad como su función de *score*. Las gaussianas de base deberán cumplir que al menos esté "suficientemente alejada" de las otras y que los pesos de las componentes sean "suficientemente variados".

d) (3 ptos.) Implemente el método *Langevin dynamics* usando algún paso ϵ fijo, un número de pasos T y una distribución prior $\pi(x)$ adecuada. Genere y grafique muestras usando el método y las gaussianas de la parte anterior. Incluya tanto un análisis de sus visualizaciones como una explicación de la intuición detrás de la actualización del método. Incluya una visualización de las trayectorias que siguen los puntos.

(P2) *Denoising Score matching* y *Annealed Langevin*

Considere la divergencia de Fisher, dada por

$$F(p,q) = \mathbb{E}_{p(x)}\left[\|\nabla_x \log p(x) - \nabla_x \log q(x)\|_2^2\right]$$

A continuación denotaremos por $s_\theta(x)$ a nuestra aproximación de $\nabla_x \log p(x)$.

a) (0.5 pto) Explique como nos acercamos a la distribución de los datos a través de la función de *score* usando la divergencia de Fisher. ¿Qué problema tiene usarla directamente como función de costo?

Nos referiremos a *denoising score matching* cuando minimizamos

$$l(\theta;\sigma) = \frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{x}|x)p(x)}\left[\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)\|_2^2\right]$$

para un σ dado. Usaremos esto para minimizar un objetivo dependiente de una secuencia de ruido fija $\{\sigma_t\}_{t=1}^T$:

$$\mathcal{L}(\theta;\{\sigma_t\}_{t=1}^T) = \frac{1}{T}\sum^T \lambda(\sigma_t) l(\theta;\sigma_t)\,. \tag{1}$$

Una red neuronal $s_\theta(x,\sigma)$ que dependa también del nivel de ruido y minimizando el objetivo anterior se denotará *noise conditional score network*.

b) (2.5 ptos.) Reescriba la función de pérdida de la ecuación 1 usando un proceso de ruido gaussiano $q_\sigma(\tilde{x}|x) = \frac{1}{\sigma\sqrt{(2\pi)^d}} e^{-\frac{1}{2\sigma}(\tilde{x}-x)^T(\tilde{x}-x)}$. Explique como aproximar las esperanzas en la práctica e implemente con esto una función de pérdida que tome un punto de dato $x \in \mathbb{R}$, un modelo de *pytorch* (una clase heredada de *nn.Module*) y otros parámetros que estime conveniente.

c) (0.75 pto.) Para alguna secuencia de ruido apropiada (justifique en base a la literatura de la unidad), seleccione algunos niveles y grafique su mixtura de gaussiana inyectada con ruido. Comente y justifique la idea de usar ruido al aprender la función de *score*.

d) (1.5 ptos.) Implemente el algoritmo *Annealed Langevin dynamics* de modo que use su función de *score* dependiente del ruido.

e) (0.75 ptos.) Pruebe su algoritmo muestreando de su mixtura gaussiana de la parte anterior y usando algún modelo simple. Usando la función de pérdida defina un bucle de entrenamiento y entrene el modelo. Detalle sus observaciones.

(P3) Generalización con ecuaciones diferenciales estocásticas

Los conceptos de las preguntas anteriores pueden generalizar a una inyección continua de tiempo. Haremos esto a través de ecuaciones diferenciales estocásticas (SDEs por sus siglas en inglés). La configuración discreta del algoritmo *annealed Langevin dynamics* corresponde a la discretización de la SDE: $dx = \sqrt{\frac{d}{dt}\sigma(t)^2}dw_t$, donde w_t denota el proceso de Wiener o movimiento browniano.

En general, consideraremos *SDEs* de la forma

$$dx = f(x,t)dt + g(t)dw\,, \tag{2}$$

donde $f(\cdot,t) : \mathbb{R}^d \mapsto \mathbb{R}^d$ es una función usualmente llamada como coeficiente de *drift* y donde $g(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ es una función escalar denotada coeficiente de difusión de $x(t)$. Se demostró que un proceso como el de la ecuación 2 cumple que su proceso reverso (i.e., de adelante hacia atrás en el tiempo) corresponde a la siguiente SDE:

$$dx = \Big[f(x,t) - g(t)^2\nabla_x \log p(x)\Big]\,dt + g(t)d\bar{w}\,,$$

donde $\bar{w}$ corresponde al proceso de Wiener pero con el tiempo yendo de T a 0.

a) (2 ptos) Implemente una clase *SDE* con los atributos que usted considere necesarios. Escriba un método que retorne el coeficiente de *drift* de la SDE reversa en el tiempo.

Nos interesará, en esta parte, usar el proceso de Ornstein–Uhlenbeck, que está dado por

$$dx = -\frac{1}{2}x_t dt + dw_t\,.$$

b) (Bonus) Describa propiedades del proceso de Ornstein–Uhlenbeck.

c) (1 pto.) Implemente una clase *SDE* específica para esta SDE.

Para poder resolver numéricamente ecuaciones diferenciales estocásticas existen algunos métodos numéricos. Uno de los más comunes es el de Euler-Maruyama, que consiste en aproximar la SDE de la ecuación 2 por trayectorias definidas recursivamente por:

$$x_{t+1} = x_t + f(x,t)\Delta t + g(t)z_t, \;\; \text{con } z_t \sim \mathcal{N}(0, \delta t I)\,.$$

En lo anterior comenzamos con un punto $x_0 \sim \pi(x)$ y consideramos la partición del espacio temporal $0 = t_1, t_2, \ldots, t_{n-1}, t_N = T$ con paso uniforme $\Delta t = \frac{T}{N}$.

d) (1 pto.) Implemente la discretización de Euler-Maruyama para la clase *SDE*.

e) (2 ptos.) Use el modelo de *score* de la pregunta anterior para generar muestras con el modelo en base a la discretización de su dinámica reversa. Grafique y comente.

f) (Bonus) Utilice sus partes anteriores pero con un dataset de imagenes (por ejemplo *MNIST*).