

Modelos Generativos Profundos para Imágenes

Introducción al modelado probabilista

Pablo Musé

pmuse@fing.edu.uy

Instituto de Ingeniería Eléctrica
Facultad de Ingeniería



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Agenda

① Motivación

② Preliminares

 Repasso de Probabilidad

 Repasso de Teoría de la Información

 Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

 Función objetivo: estimador de máxima verosimilitud

 Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

 Modelos gaussianos

 Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

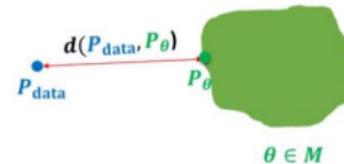
Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

Modelado generativo

Disponemos de un conjunto de muestras de entrenamiento, e.g., fotos de perros, sorteadas de una distribución subyacente p_{data}



$$\mathbf{x}_n \sim p_{data}(\mathbf{x}), n = 1, \dots, N \quad \text{Familia de modelos}$$

Objetivo: aprender una densidad de probabilidad $p(\mathbf{x})$ sobre imágenes \mathbf{x} con capacidad de:

- **Generación (muestreo):** Si muestreamos $\mathbf{x}_{nuevo} \sim p(\mathbf{x})$, debería parecerse a un perro.
- **Estimación de densidad (detección de anomalías):** $p(\mathbf{x})$ debería ser alta si \mathbf{x} se parece a un perro, y baja en caso contrario
- **Aprendizaje no supervisado de representaciones (características):** extraer *features* que codifiquen lo que estas imágenes tienen en común, e.g., orejas, cola, etc.

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

Preliminares: repaso de Probabilidad

- Modelamos los datos como **variables aleatorias** $\mathbf{X} \in \mathbb{R}^d$, distribuidos según $\mathbf{X} \sim p(\mathbf{x})$
 - Si \mathbf{X} es discreta, $p(\mathbf{x})$ es una función de masa de probabilidad, $p(\mathbf{x}) \geq 0$, $\sum_{\mathbf{x}} p(\mathbf{x}) = 1$.
 - Si \mathbf{X} es continua, $p(\mathbf{x})$ es una densidad de probabilidad, $p(\mathbf{x}) \geq 0$, $\int_{\mathbf{x}} p(\mathbf{x}) d\mathbf{x} = 1$.
- **Independencia:** \mathbf{X} et \mathbf{Y} son independientes si y solo si $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$.
- **Densidades marginales:**

$$\text{Continuo: } p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y} \quad \text{Discreto: } p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$$

- **Densidad condicional:**
- **Regla del producto:**
- **Regla de Bayes:**

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

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}) &= p(\mathbf{x} | \mathbf{y})p(\mathbf{y}) \\ &= p(\mathbf{y} | \mathbf{x})p(\mathbf{x}) \end{aligned}$$

$$p(\mathbf{y} | \mathbf{x}) = \frac{p(\mathbf{x} | \mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}$$

- **Esperanza de una función de V.A.:**

$$\text{Continuo: } \mathbb{E}_{\mathbf{X}} [f(\mathbf{x})] = \int_{\mathbf{x}} f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

$$\text{Discreto: } \mathbb{E}_{\mathbf{X}} [f(\mathbf{x})] = \sum_{\mathbf{x}} f(\mathbf{x}) p(\mathbf{x})$$

Preliminares: repaso de Probabilidad

- Distribución de Bernoulli: $X \in \{0, 1\}$, $\theta \in [0, 1]$

$$\text{Ber}(x; \theta) = \theta^x (1 - \theta)^{1-x}, \quad \mathbb{E}[X] = \theta, \quad \text{Var}[X] = \theta(1 - \theta).$$

- Distribución categórica: $\mathbf{X} = (X_1, X_2, \dots, X_K)$ vector *one-hot* (todas las coordenadas 0 salvo una que vale 1), $\theta_k \in [0, 1]$, $\sum_{k=1}^K \theta_k = 1$

$$\text{Cat}(\mathbf{x}; \boldsymbol{\theta}) = \prod_{k=1}^K \theta_k^{x_k}, \quad \mathbb{E}[X_k] = \theta_k, \quad \text{Var}[X_k] = \theta_k (1 - \theta_k).$$

- Distribución gaussiana: $\mathbf{X} \in \mathbb{R}^d$, $\boldsymbol{\mu} \in \mathbb{R}^d$, $\boldsymbol{\Sigma}$ matriz de covarianza $d \times d$

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}$$
$$\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu}, \quad \text{Cov}[\mathbf{X}] = \boldsymbol{\Sigma}.$$

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

Preliminares: repaso de Teoría de la Información

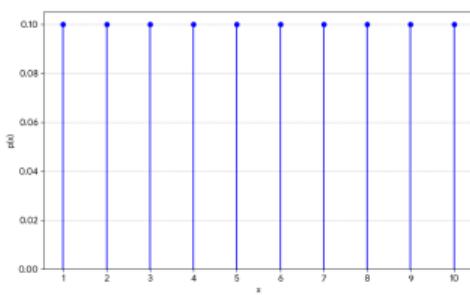
- Entropía de una V.A. \mathbf{X}

- Representa el grado de "incertidumbre" en el valor que tome \mathbf{X}

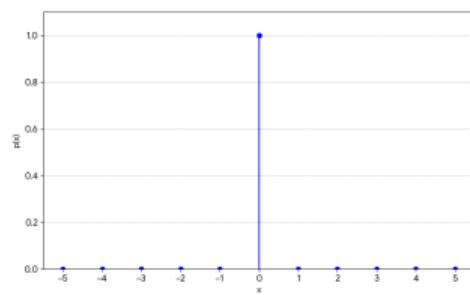
$$H(\mathbf{X}) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [-\log p(\mathbf{x})] \quad \text{Continuo: } H(\mathbf{X}) = - \int_{\mathbf{x}} \log(p(\mathbf{x})) p(\mathbf{x}) d\mathbf{x}$$

$$\text{Discreto: } H(\mathbf{X}) = - \sum_k \log(p_k) p_k, \quad p_k = P(\mathbf{X} = k)$$

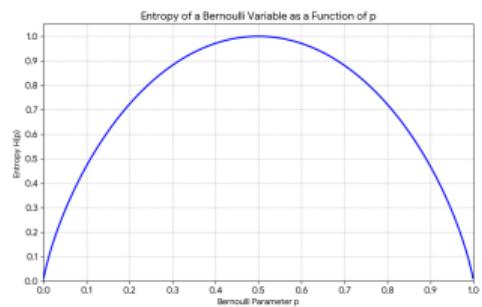
Uniforme (entropía máxima entre las distrib. de soporte acotado)



Delta de Kronecker (entropía nula)



$H(p)$ para $\text{Ber}(p)$ (máxima en $p = 1/2$)



Preliminares: repaso de Teoría de la Información

- **Entropía condicional:** incertidumbre sobre \mathbf{X} habiendo observado \mathbf{Y} :

$$H(\mathbf{X} \mid \mathbf{Y}) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim p(\mathbf{x}, \mathbf{y})} [-\log p(\mathbf{x} \mid \mathbf{y})]$$

- **Información mutua:**

$$I(\mathbf{X}; \mathbf{Y}) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim p(\mathbf{x}, \mathbf{y})} \left[\log \left(\frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} \right) \right]$$

- Medida de dependencia entre \mathbf{X} e \mathbf{Y}
- Reducción de incertidumbre en \mathbf{X} al observar \mathbf{Y}

$$I(\mathbf{X}; \mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X} \mid \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y} \mid \mathbf{X})$$

- Si \mathbf{X} , \mathbf{Y} son independientes, $I(\mathbf{X}; \mathbf{Y}) = 0$.

Preliminares: repaso de Teoría de la Información

- Divergencia de Kullback-Liebler entre dos distribuciones p y q :

$$KL[p(\mathbf{x})\|q(\mathbf{x})] = \mathbb{E}_{\mathbf{x} \sim p} \left[\log \frac{p(\mathbf{x})}{q(\mathbf{x})} \right]$$

- Cantidad esperada de **bits extra** que se necesitan para describir muestras de p usando un código de compresión basado en q en lugar de p
- Mide cuán disímiles son dos distibuciones
- $KL[p(x)\|q(x)] \geq 0$. La igualdad se da si y solo si $p = q$
- No es una distancia (en general no es simétrica ni verifica la desigualdad triangular).
- Entropía cruzada

$$\begin{aligned} KL[p(\mathbf{x})\|q(\mathbf{x})] &= \mathbb{E}_{\mathbf{x} \sim p} [\log p(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p} [\log q(\mathbf{x})] \\ &= -H(p(\mathbf{x})) + H(p(\mathbf{x}), q(\mathbf{x})) \end{aligned}$$

$$H(p(\mathbf{x}), q(\mathbf{x})) = H(p(\mathbf{x})) + KL[p(\mathbf{x})\|q(\mathbf{x})]$$

- Cantidad esperada **total de bits** que se necesitan para describir muestras de p usando un código de compresión basado en q .

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

Repaso de estimación Monte Carlo

- ① Expresar la cantidad de interés como el valor esperado de una V.A.:

$$\mathbb{E}_{X \sim P(x)} [g(X)] = \sum_x g(x)P(x).$$

- ② Consideramos N V.A. i.i.d. $X_1, \dots, X_N \sim P(x)$.
- ③ Definimos el estimador Monte Carlo \hat{g} de $\mathbb{E}_{X \sim P(x)} [g(X)]$ como:

$$\hat{g}(X_1, \dots, X_N) = \frac{1}{N} \sum_{n=1}^N g(X_n).$$

Repaso de estimación Monte Carlo

Propiedades del estimador Monte Carlo:

- \hat{g} es insesgado:

$$\mathbb{E}_P [\hat{g}] = \frac{1}{N} \sum_{n=1}^N \mathbb{E}_P [g(X_n)] = \mathbb{E}_P [g(X)].$$

- \hat{g} es consistente: por la ley de los grandes números,

$$\hat{g} = \frac{1}{N} \sum_{n=1}^N g(X_n) \xrightarrow{N \rightarrow +\infty} \mathbb{E}_P [g(X)].$$

- La varianza de \hat{g} decrece linealmente con la cantidad de muestras:

$$\text{Var}_P [\hat{g}] = \text{Var}_P \left[\frac{1}{N} \sum_{n=1}^N g(X_n) \right] = \frac{1}{N} \text{Var}_P [g(X)].$$

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

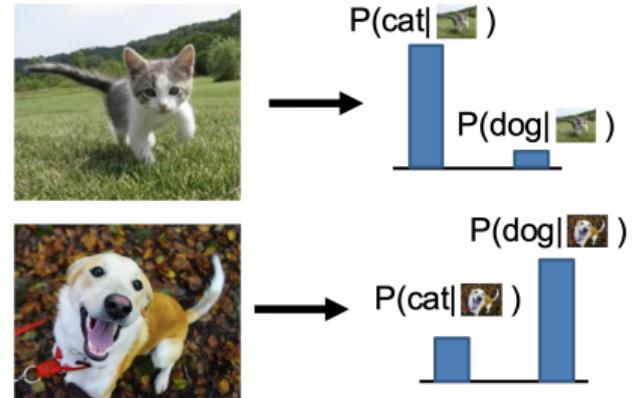
Modelos generativos vs. modelos discriminativos

- En machine learning, los modelos se pueden categorizar según su enfoque para aprender de los datos.
- Dos categorías fundamentales son los **Modelos Generativos** y los **Modelos Discriminativos**.
- Difieren significativamente en lo que aprenden y cómo se utilizan.

Modelos discriminativos: enfoque en las fronteras de decisión

- Aprender un mapeo directo de los datos de entrada x a las etiquetas de salida y .
- Modelan la probabilidad condicional $p(y|x)$.
- Esencialmente, aprenden la **frontera de decisión** entre diferentes clases.

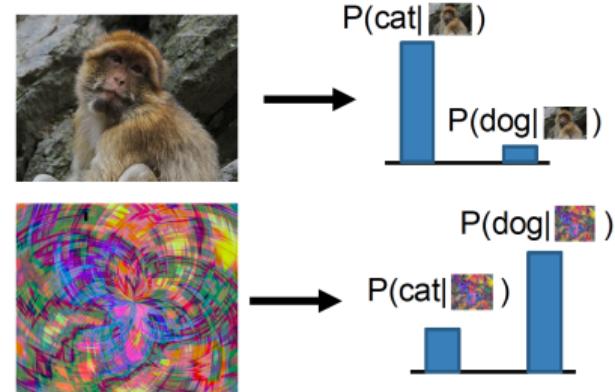
Posibles etiquetas de cada imagen
compiten por la probabilidad



Modelos discriminativos: enfoque en las fronteras de decisión

- Aprender un mapeo directo de los datos de entrada x a las etiquetas de salida y .
- Modelan la probabilidad condicional $p(y|x)$.
- Esencialmente, aprenden la **frontera de decisión** entre diferentes clases.

No hay forma de manejar entradas fuera de clase: asignan una distribución de etiquetas para cada imagen



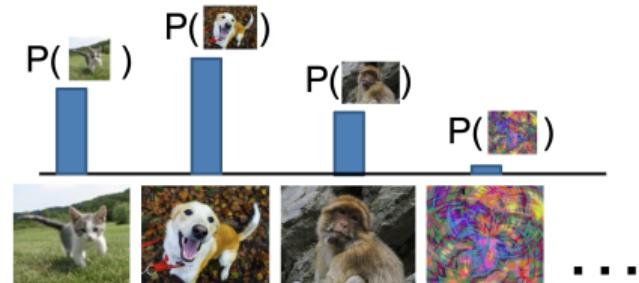
- **Ejemplos:** regresión logística, SVM, árboles de decisión, redes neuronales para clasificación o regresión, ...
- **Caso de Uso:** Principalmente para tareas de clasificación y regresión.

Modelos generativos: comprender la distribución de los datos

- Aprender la distribución subyacente de los datos, $p(\mathbf{x}, \mathbf{y})$ o $p(\mathbf{x})$.
- Modelan cómo se generan los datos.
- Pueden usarse para generar nuevas muestras de datos que se parezcan a los datos de entrenamiento.
- También pueden usarse para clasificación aplicando Bayes: $p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}$.

Modelo generativo $p(\mathbf{x})$:

- Todas las posibles imágenes compiten por la densidad de probabilidad
- El modelo puede detectar outliers asignándoles baja probabilidad.

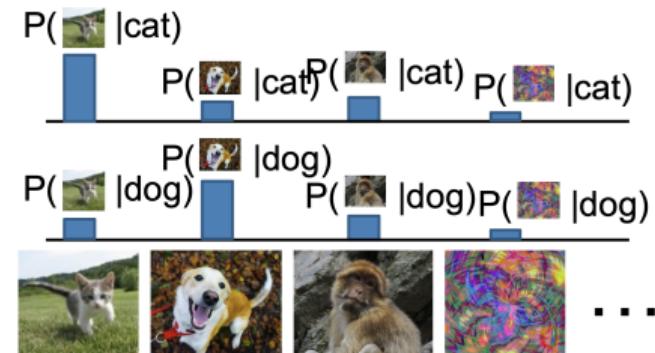


Modelos generativos: comprender la distribución de los datos

- Aprender la distribución subyacente de los datos, $p(\mathbf{x}, y)$ o $p(\mathbf{x})$.
- Modelan cómo se generan los datos.
- Pueden usarse para generar nuevas muestras de datos que se parezcan a los datos de entrenamiento.
- También pueden usarse para clasificación aplicando Bayes: $p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$.

Modelo generativo condicional $p(\mathbf{x} | y)$:

- Cada posible etiqueta induce una competencia entre todas las imágenes posibles



Por qué modelos generativos (más allá de la generación de datos)

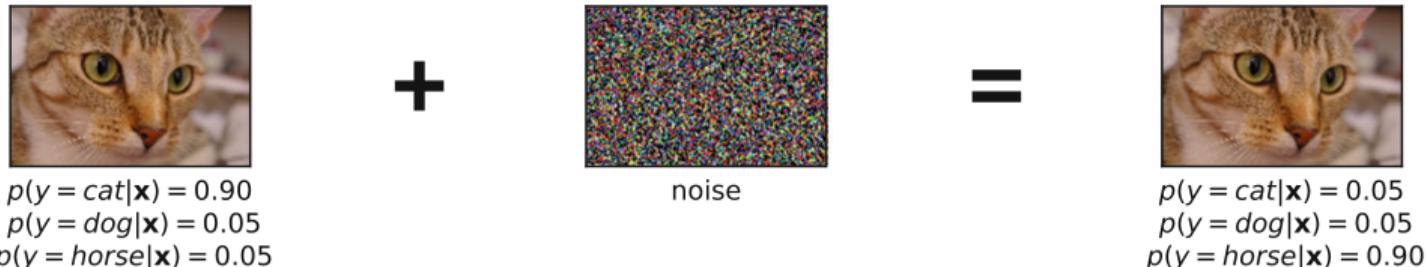


Fig. 1.1 An example of adding noise to an almost perfectly classified image that results in a shift of predicted label.

Importancia del modelado
de los datos y de la
incertidumbre en los
procesos de decisión

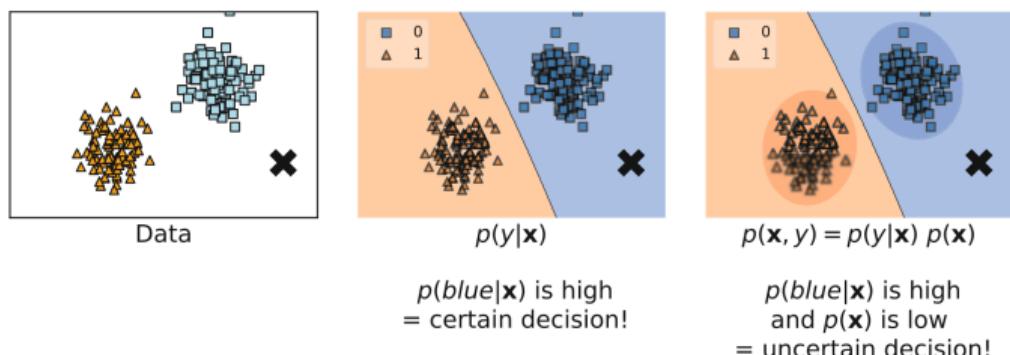


Fig. 1.2 And example of data (left) and two approaches to decision-making: (middle) a discriminative approach, (right) a generative approach.

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

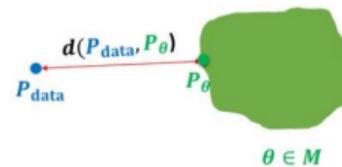
Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

Modelado generativo

Recordemos el problema que queremos resolver:

- Disponemos de muestras de entrenamiento sorteadas de una distribución subyacente p_{data}



$$\mathbf{x}_n \sim p_{data}(\mathbf{x}), \quad n = 1, \dots, N \quad \text{Familia de modelos}$$

- **Objetivo:** aprender una densidad de probabilidad $p(\mathbf{x})$ que nos permita:
 - Generar muestras realistas
 - Tener una estimación fiable de la densidad (e.g. para detección de anomalías)
 - Aprender buenas representaciones (vectores de features) de los datos

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

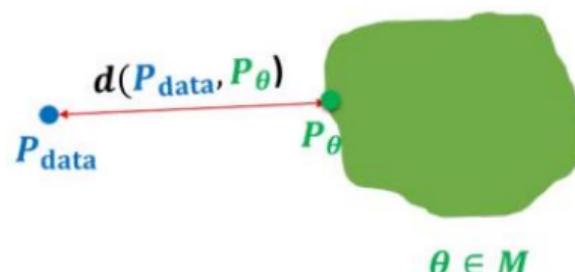
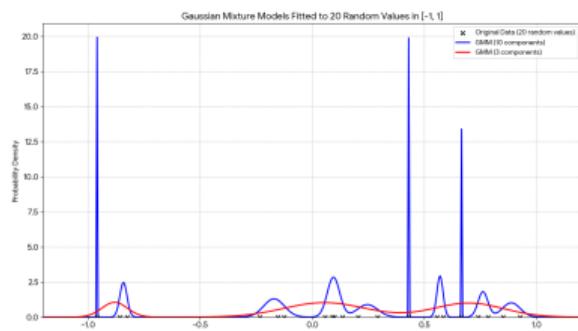
⑥ Modelos generativos profundos: introducción y taxonomía

Aprendizaje como un problema de estimación de densidades

- **Estimación de densidades:** problema difícil cuando los datos \mathbf{x} son de alta dimensión (maldición de la dimensionalidad). Dos aspectos del problema:
 - ① Cómo representar $p(\mathbf{x})$;
 - ② Cómo aprender o estimar $p(\mathbf{x})$ a partir de los datos.
- Los **algoritmos de modelado generativo** que veremos tienen la siguiente estructura:
 - ① Definir una familia de modelos $p_\theta(\mathbf{x})$, paramétrica en θ ;
 - ② Definir una función objetivo en la variable θ (e.g., maximizar $\frac{1}{N} \sum_{n=1}^N \log p_\theta(\mathbf{x}_i)$);
 - ③ Encontrar el mejor θ posible mediante algún algoritmo de optimización.
- Una vez estimado p_θ podemos **generar muestras** de esta distribución.

Aprendizaje como estimación de densidades

- Los datos \mathbf{x} (e.g., imágenes naturales) se distribuyen de acuerdo a una densidad $p_{data}(\mathbf{x})$.
- No conocemos $p_{data}(\mathbf{x})$: contamos con N muestras $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$ **i.i.d.** para estimarla.
- Nos damos una familia paramétrica de densidades $p_\theta(\mathbf{x})$ como modelo:
 - ① Qué familia de modelos consideramos?
 - ② Cómo ajustamos $p_\theta(\mathbf{x})$ a $p_{data}(\mathbf{x})$?



⇒ Buscamos que $p_\theta(\mathbf{x})$ esté cerca de $p_{data}(\mathbf{x})$ minimizando la divergencia KL en θ

Aprendizaje como estimación de densidades

$$\begin{aligned} KL [p_{data}(\mathbf{x}) \| p_{\theta}(\mathbf{x})] &= \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} \left[\log \frac{p_{data}(\mathbf{x})}{p_{\theta}(\mathbf{x})} \right] \\ &= \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [\log p_{data}(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [\log p_{\theta}(\mathbf{x})] \\ &= -H(p_{data}) \underbrace{- \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [\log p_{\theta}(\mathbf{x})]}_{\text{Entropía cruzada (CE), } H(p_{data}, p_{\theta})} \end{aligned}$$

⇒ Minimizar KL en θ es equivalente a minimizar CE

No conocemos p_{data} ⇒ Estimamos la esperanza por Monte Carlo a partir las muestras \mathcal{D} :

$$H(p_{data}, p_{\theta}) = -\frac{1}{N} \sum_{n=1}^N \log p_{\theta}(\mathbf{x}_n)$$

⇒ Es la $-\log$ -verosimilitud de una muestra \mathcal{D} i.i.d. (es el estimador MLE)

$$\operatorname{argmin}_{\theta} KL [p_{data} \| p_{\theta}] = \operatorname{argmin}_{\theta} H(p_{data}, p_{\theta}) = \operatorname{argmax}_{\theta} \frac{1}{N} \sum_{n=1}^N \log p_{\theta}(\mathbf{x}_n).$$

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

Optimización de parámetros: descenso por gradiente estocástico

- Función de costo

$$L(\theta) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [f(\mathbf{x}; \theta)].$$

- Objetivo: encontrar los parámetros θ que minimizan la función de costo:

$$\min_{\theta} L(\theta).$$

- Métodos de primer orden:

- Escalan bien con la dimensión (trabajamos en espacios de muy alta dimensión).
- Método de descenso por el gradiente:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_{\theta} L(\theta^{(t)}).$$

- Existen variantes con aceleraciones (Adagrad, RMSProp, Adam, etc.) y versiones no diferenciables (operador proximal).

Optimización de parámetros: descenso por gradiente estocástico

- ¿Cómo calculamos $\nabla_{\theta} L(\theta) = \nabla_{\theta} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [f(\mathbf{x}; \theta)]$?
 - En general no conocemos $p(\mathbf{x})$ de forma explícita, y aún conociéndola la integración es costosa (alta dimensionalidad).
 - Lo usual es considerar aproximaciones Monte Carlo del gradiente:
 - $\mathbf{x}_1, \dots, \mathbf{x}_b$ batch de muestras i.i.d. $\sim p(\mathbf{x})$.
 - $\frac{1}{b} \sum_{i=1}^b \nabla_{\theta} f(\mathbf{x}_i; \theta)$ estimador insesgado de $\nabla_{\theta} L(\theta)$.
- Algoritmo de descenso por gradiente estocástico:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \frac{1}{b} \sum_{i=1}^b \nabla_{\theta} f \left(\mathbf{x}_i; \boldsymbol{\theta}^{(t)} \right).$$

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

Modelo gaussiano

- $\mathcal{D} = \{\mathbf{x}_n, n = 1, \dots, N\}$ muestras i.i.d., $\mathbf{x}_n \in \mathbb{R}^d$.
- Modelo gaussiano $p_\theta(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$:

$$p_\theta(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp \left[-(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right].$$

- **Objetivo:** estimar $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ por máxima verosimilitud.

$$p_\theta(\mathcal{D}) = \prod_{n=1}^N p_\theta(\mathbf{x}_n) \implies \ell(\boldsymbol{\theta}) := \log p_\theta(\mathcal{D}) = \sum_{n=1}^N \log p_\theta(\mathbf{x}_n)$$

- Estimador de máxima verosimilitud $\boldsymbol{\theta}_{ML} = \{\boldsymbol{\mu}_{ML}, \boldsymbol{\Sigma}_{ML}\}$:

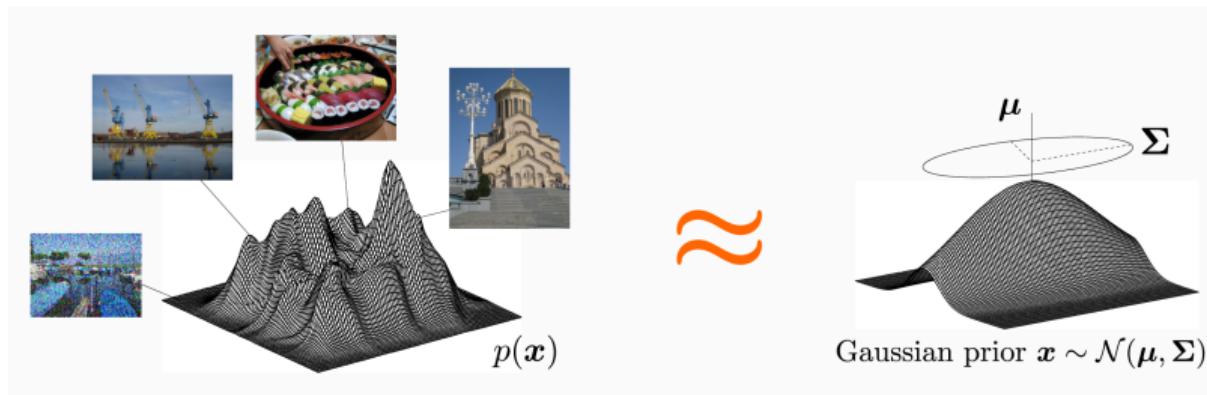
$$\begin{cases} \nabla_{\boldsymbol{\mu}} \ell(\boldsymbol{\theta}_{ML}) = 0 \\ \nabla_{\boldsymbol{\Sigma}} \ell(\boldsymbol{\theta}_{ML}) = 0 \end{cases} \xrightarrow{\text{ejercicio}} \begin{cases} \boldsymbol{\mu}_{ML} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \\ \boldsymbol{\Sigma}_{ML} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}_{ML})(\mathbf{x}_n - \boldsymbol{\mu}_{ML})^T. \end{cases}$$

Ejemplo 1: generación de imágenes con modelo gaussiano

- Consideramos un modelo gaussiano para distribución de imágenes $\mathbf{x} \in \mathbb{R}^d$:

$$\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp \left[-(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

- $\boldsymbol{\mu}$: imagen media,
- $\boldsymbol{\Sigma}$: matriz de covarianza de las imágenes.



Ejemplo 1: generación de imágenes con modelo gaussiano

- Armamos un conjunto \mathcal{D} de imágenes:

$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} = \left\{ \begin{array}{c} \text{[Image 1]}, \text{[Image 2]}, \text{[Image 3]}, \text{[Image 4]}, \text{[Image 5]}, \text{[Image 6]}, \\ \dots \end{array} \right.$$

- Estimamos la media

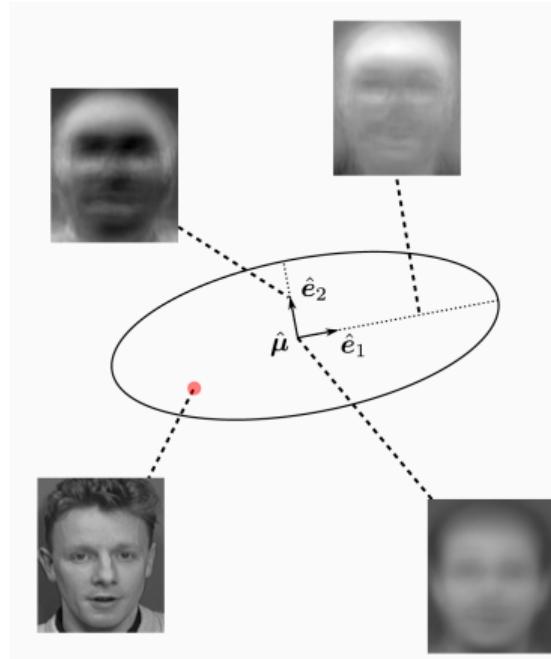
$$\hat{\mu} = \frac{1}{N} \sum_i \mathbf{x}_i = \text{[Blurry Face Image]}$$

- Estimamos la matriz de covarianza: $\hat{\Sigma} = \frac{1}{N} \sum_i (\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^T = \hat{\mathbf{E}} \hat{\Lambda} \hat{\mathbf{E}}^T$

$$\hat{\mathbf{E}} = \underbrace{\left\{ \begin{array}{c} \text{[Blurry Face Image 1]}, \text{[Blurry Face Image 2]}, \text{[Blurry Face Image 3]}, \text{[Blurry Face Image 4]}, \text{[Blurry Face Image 5]}, \text{[Blurry Face Image 6]}, \\ \dots \end{array} \right\}}_{\text{vectores propios de } \hat{\Sigma}, \text{ i.e., direcciones principales}}$$

Ejemplo 1: generación de imágenes con modelo gaussiano

Hemos entrenado un modelo generativo de caras:



Ejemplo 1: generación de imágenes con modelo gaussiano

¿Cómo generamos muestras de $\mathcal{N}(\hat{\mu}, \hat{\Sigma})$?

$$\begin{cases} z & \sim \mathcal{N}(0, I_d) \leftarrow \text{Generar una variable latente aleatoria} \\ x & = \hat{\mu} + \hat{E}\hat{\Lambda}^{1/2}z \end{cases}$$

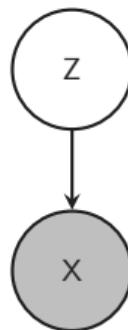


- **El modelo no genera muestras realistas** (la hipótesis de modelo gaussiano es demasiado simple).

Ejemplo 2: modelos generativos con mezcla de gaussianas

2.1. Definición del modelo paramétrico

- Queremos estimar la distribución del peso de los gatos.
- Llamamos X la variable aleatoria del peso de un gato.
- Utilizamos una distribución gaussiana para el peso de los machos, y otra para el peso de las hembras. La V.A. latente $Z \in \{0, 1\}$ codifica si el gato es hembra o macho, resp.



$$Z \sim \text{Ber}(\omega)$$

$$X \sim \mathcal{N}(\mu(z), \sigma^2(z))$$

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

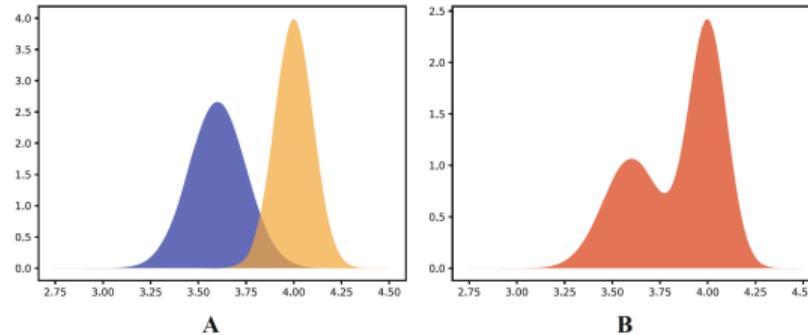
$$= \mathcal{N}(x; \mu(z), \sigma^2(z))\text{Ber}(z; \omega)$$

Ejemplo 2: modelos generativos con mezcla de gaussianas

2.1. Definición del modelo paramétrico

$$\begin{aligned} p(x) &= \sum_z p(x, z) = \sum_z \mathcal{N}(x; \mu(z), \sigma^2(z)) \text{Ber}(z; \omega) \\ &= (1 - \omega) \mathcal{N}(x; \mu_0, \sigma_0^2) + \omega \mathcal{N}(x; \mu_1, \sigma_1^2), \end{aligned}$$

donde $\mu_i = \mu(z = i)$, $\sigma_i^2 = \sigma^2(z = i)$.



Ejemplo 2: modelos generativos con mezcla de gaussianas

2.2. Definición de la función objetivo

- Mezcla de gaussianas (MoG) de K componentes
- $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ datos i.i.d. $\sim p_{\theta}(\mathbf{x}) = \sum_{k=1}^K \omega_k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, $(\sum_k \omega_k = 1)$
- Encontrar el vector de parámetros θ que maximice la verosimilitud conjunta $p_{\theta}(\mathcal{D})$:

$$\begin{aligned} p_{\theta}(\mathcal{D}) &= \prod_{n=1}^N p_{\theta}(\mathbf{x}_n) \\ &= \prod_{n=1}^N \sum_{k=1}^K \omega_k \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \end{aligned}$$

Ejemplo 2: modelos generativos con mezcla de gaussianas

2.2. Definición de la función objetivo

Para MoG es más sencillo y numéricamente mejor maximizar $\log p_{\theta}(\mathcal{D})$:

$$\log p_{\theta}(\mathcal{D}) = \sum_{n=1}^N \log p_{\theta}(\mathbf{x}_n) = \sum_{n=1}^N \log \left(\sum_{k=1}^K \omega_k \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

Usando la identidad $\exp(\log x) = x$:

$$\begin{aligned}\log p_{\theta}(\mathcal{D}) &= \sum_{n=1}^N \log \left(\sum_{k=1}^K \exp \left(\underbrace{\log \omega_k + \log \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{a_k^n} \right) \right) \\ &= \sum_{n=1}^N \text{LSE}(a_1^n, \dots, a_K^n)\end{aligned}$$

(implementaciones numéricamente estables de LogSumExp disponibles en varios frameworks de ML)

Ejemplo 2: modelos generativos con mezcla de gaussianas

2.3. Entrenamiento del modelo

Se puede optimizar la MoG de dos formas:

- ① Usando el algoritmo EM* (Expectation-Maximization): conduce a un algoritmo iterativo en los parámetros ω_k , μ_k y Σ_k que converge a un máximo local de la log-verosimilitud.
- ② Usando métodos basados en gradiente, en este caso, gradiente ascendente:
 - $\theta(t+1) = \theta(t) + \eta \nabla_{\theta} \log p_{\theta(t)}(\mathcal{D})$, $\eta > 0$.
 - Los gradientes se calculan con Autograd.

*Ver por ejemplo C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006

Ejemplo 2: modelos generativos con mezcla de gaussianas

Resultados

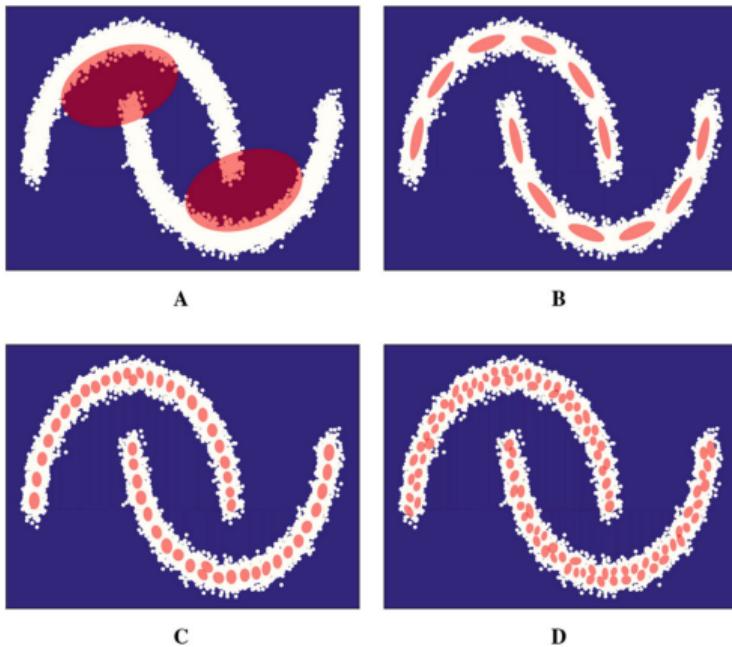


Fig. 2.5 Several examples of MoGs with varying number of components: (a) $K = 2$, (b) $K = 12$, (c) $K = 50$, (d) $K = 100$.

① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

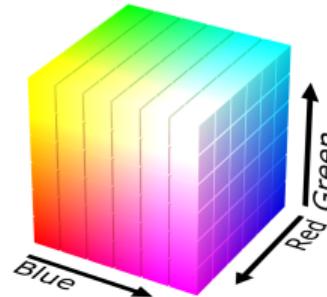
Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

Ejemplo 3: Imágenes RGB

- Para definir una distribución sobre un pixel usamos tres V.A. discretas:

- Canal rojo, $R \in \{0, \dots, 255\}$
- Canal verde, $G \in \{0, \dots, 255\}$
- Canal azul, $B \in \{0, \dots, 255\}$



- Muestrear de la distribución conjunta $(R, G, B) \sim p(r, g, b) = \Pr(R = r, G = g, B = b)$ genera un pixel de una imagen a color.
- ¿Cuántos parámetros se necesitan para especificar la distribución conjunta $p(r, g, b)$?

$$256 \times 256 \times 256 - 1 = 16.777.215$$

La maldición de la dimensionalidad en los modelos probabilísticos

- Queremos modelar imágenes B&N de dígitos, con $d = 28 \times 28$ pixels.



- **Modelo:** pixels X_1, \dots, X_d V.A.s Bernoulli i.e., $X_i \in \{0, 1\} = \{\text{Negro, Blanco}\}$.
- ¿Cuántos estados posibles?

$$\underbrace{2 \times 2 \times \cdots \times 2}_{d \text{ veces}} = 2^d$$

- Muestrear de $p(x_1, \dots, x_d)$ genera una imagen
- Cuántos parámetros para especificar la distribución conjunta $p(x_1, \dots, x_d)$?

$$2^d - 1 \approx 10^{236}$$

Modelos eficientes en parámetros mediante independencia condicional

- Si X_1, \dots, X_d son independientes, entonces

$$p(x_1, \dots, x_d) = p(x_1)p(x_2) \cdots p(x_d).$$

- Cantidad de posibles estados: 2^d
- Cantidad de parámetros para especificar la distribución conjunta $p(x_1, \dots, x_n)$: d
- Ejemplos de imágenes sorteadas según $p(x_1, \dots, x_d) = p(x_1)p(x_2) \cdots p(x_d)$:



⇒ La hipótesis de independencia es demasiado fuerte, no genera muestras útiles.

Introduciendo estructura mediante independencia condicional

- Regla de la cadena:

$$p(x_1, \dots, x_d) = p(x_1) p(x_2 | x_1) p(x_3 | x_1, x_2) \cdots p(x_d | x_1, \dots, x_{d-1})$$

- ¿Cuántos parámetros para especificar la distribución?

- $p(x_1)$ requiere 1 parámetro
 - $p(x_2 | x_1 = 0)$ requiere 1 parámetro, $p(x_2 | x_1 = 1)$ requiere 1 parámetro
⇒ Total: 2 parámetros
 - $p(x_2 | x_1, x_2)$: requiere en total 2^2 parámetros
 - ...

⇒ Total para $p(x_1, \dots, x_d)$: $1 + 2 + \cdots + 2^{d-1} = 2^d - 1$ parámetros

⇒ Sigue siendo exponencial en d (es normal, todavía no impusimos nada).

Introduciendo estructura mediante independencia condicional

- Tenemos

$$p(x_1, \dots, x_d) = p(x_1) p(x_2 | x_1) p(x_3 | x_1, x_2) \cdots p(x_d | x_1, \dots, x_{d-1})$$

- Supongamos ahora $X_{i+1} \perp \{X_1, \dots, X_{i-1}\} | X_i$ (propiedad Markoviana):

$$\begin{aligned} \implies p(x_1, \dots, x_d) &= p(x_1) p(x_2 | x_1) p(x_3 | \cancel{x_1}, x_2) \cdots p(x_d | \cancel{x_1}, \dots, \cancel{x_{i-1}}) \\ &= p(x_1) p(x_2 | x_1) p(x_3 | x_2) \cdots p(x_d | x_{d-1}) \end{aligned}$$

- ¿Cuántos parámetros para especificar la distribución?

$2d - 1 \longrightarrow$ Reducción exponencial!

Redes bayesianas: idea

Se construyen especificando parametrizaciones condicionales:

- ① Para cada V.A. X_i , se especifica $p(x_i | \mathbf{x}_{A_i})$ par el conjunto \mathbf{X}_{A_i} de V.A.s.
- ② Se obtiene la parametrización conjunta

$$p(x_1, \dots, x_d) = \prod_i p(x_i | \mathbf{x}_{A_i})$$

garantizando que sea una distribución de probabilidad (factorización mediante regla de la cadena e imposición de independencias condicionales).

Redes bayesianas: definición

- Se especifica con un grafo acíclico dirigido (DAG) $G = (V, E)$ con:
 - ① Un nodo $i \in V$ para cada V.A. X_i
 - ② Una distribución condicional (CPD) por nodo, $p(x_i | \mathbf{x}_{\text{Pa}(i)})$, especificando la probabilidad de X_i condicionada a los nodos padres.
- $G = (V, E)$ define la estructura del la red bayesiana
- Distribución conjunta:

$$p(x_1, \dots, x_d) = \prod_{i \in V} p(x_i | \mathbf{x}_{\text{Pa}(i)})$$

- Representación económica: exponencial en $|\text{Pa}(i)|$, no en $|V|$.

Redes bayesianas: ejemplo

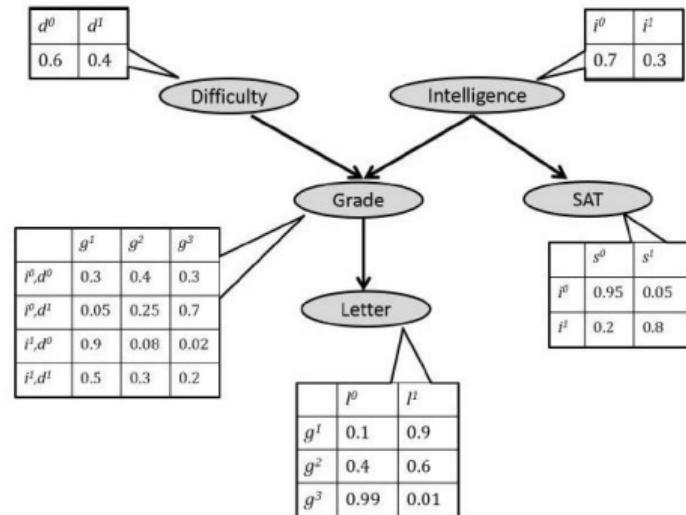
- ¿Cuál es la densidad conjunta?

$$p(d, i, g, s, l) = p(d)p(i)p(g | i, d)p(s | i)p(l | g)$$

- Supuestos de independencia:

Regla de la cadena: $p(d, i, g, s, l) = p(d)p(i | d)p(g | i, d)p(s | i, d, g)p(l | g, d, i, s)$

$$\implies D \perp I, \quad S \perp \{D, G\} | I, \quad L \perp \{I, D, S\} | G$$



① Motivación

② Preliminares

Repasso de Probabilidad

Repasso de Teoría de la Información

Repasso de estimación Monte Carlo

③ Modelos generativos vs. modelos discriminativos

④ Aprendizaje de modelos generativos

Función objetivo: estimador de máxima verosimilitud

Algoritmo de aprendizaje: descenso por gradiente estocástico

⑤ Tipos de modelos generativos

Modelos gaussianos

Modelos generativos generales y maldición de la dimensionalidad

⑥ Modelos generativos profundos: introducción y taxonomía

Redes bayesianas vs. modelos neuronales: modelos autorregresivos

- Modelo general:

$$p(x_1, \dots, x_d) = p(x_1) p(x_2 | x_1) p(x_3 | x_1, x_2) \dots p(x_d | x_1, \dots, x_{d-1}).$$

- En las redes bayesianas imponemos estructura asumiendo independencia condicional (e.g. propiedad markoviana):

$$p(x_1, \dots, x_d) \approx p(x_1) p(x_2 | x_1) p(x_3 | \cancel{x_1}, x_2) \dots p(x_d | \cancel{x_1}, \dots, \cancel{x_{d-2}}, x_{d-1}).$$

- Modelos Neuronales:

$$p(x_1, \dots, x_d) \approx p(x_1) p(x_2 | x_1) \text{pNeuralNet}(x_3 | x_1, x_2) \dots \text{pNeuralNet}(x_d | x_1, \dots, x_{d-1})$$

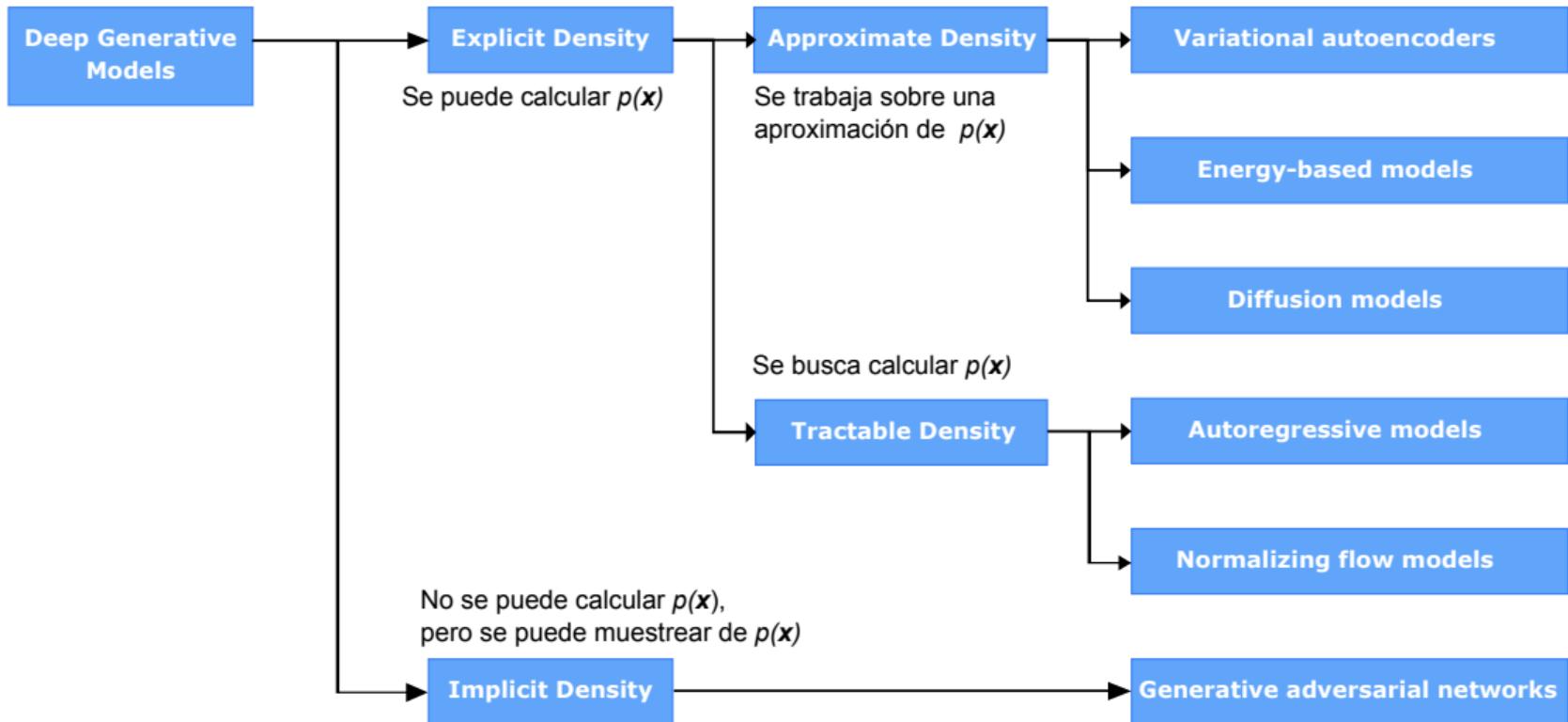
⇒ Asumimos una forma funcional específica, paramétrica, para las densidades condicionales: **redes neuronales profundas (pueden aproximar cualquier función)**.

Redes bayesianas vs. modelos neuronales: mezclas de gaussianas

Ejemplos:

- **Mezcla de K gaussianas:** red bayesiana $Z \rightarrow X$, $p_{Z,X}(z,x) = p_Z(z) p_{X|Z}(x | z)$,
 - $Z \sim \text{Cat}(p_1, p_2, \dots, p_K)$
 - $X | (Z = k) \sim \mathcal{N}(x; \mu_k, \sigma_k^2)$
 - Parámetros: p_k, μ_k, σ_k^2 , $k = 1, 2, \dots, K$.
- **Mezcla de infinitas gaussianas:** red bayesiana $Z \rightarrow X$, $p_{Z,X}(z,x) = p_Z(z) p_{X|Z}(x | z)$,
 - $Z \sim \mathcal{U}(a, b)$
 - $X | (Z = z) \sim \mathcal{N}(x; z, \sigma^2)$
 - Parámetros: a, b, σ^2
- **Mezcla neuronal de infinitas gaussianas:** red $Z \rightarrow X$, $p_{Z,X}(z,x) = p_Z(z) p_{X|Z}(x | z)$,
 - $Z \sim \mathcal{N}(0, 1)$
 - $X | (Z = z) \sim \mathcal{N}(\mu_\theta(z), e^{\sigma_\phi(z)})$
 - $\mu_\theta, \sigma_\phi : \mathbb{R} \rightarrow \mathbb{R}$ son redes neuronales con pesos θ, ϕ , resp.

Modelos generativos profundos



Referencias

-  J. M. Tomczak, *Deep Generative Modeling*.
Springer Cham, 2024.
-  C. M. Bishop, *Pattern Recognition and Machine Learning*.
Springer, 2006.
-  Stanford, “CS236 Deep Generative Models.” <https://deepgenerativemodels.github.ioLecture>, 2023.