

# Aprendizaje no Supervisado

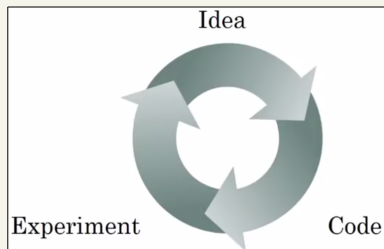
**Taller de Procesamiento de Señales**

# Agenda

- 1 Autoencoders
- 2 Principal Components Analysis (PCA)
- 3 K-Means
- 4 Algoritmo EM
- 5 Factor Analysis

# Aprendizaje Estadístico

- No se conoce la verdadera estadística.
- Se aprende por medio de datos.
- El buen desempeño no debe limitarse a los datos conocidos.

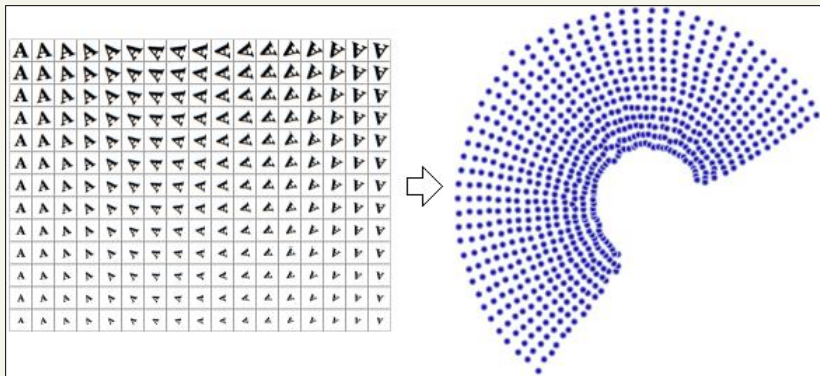


## TIPOS DE APRENDIZAJES

- Aprendizaje supervisado: Cuento con pares de datos  $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ .
- Aprendizaje no supervisado: Cuento solamente con datos  $\{\mathbf{x}^{(i)}\}_{i=1}^n$ .
- Aprendizaje semi-supervisado: Cuento con muchos datos no supervisados y unos pocos supervisados.

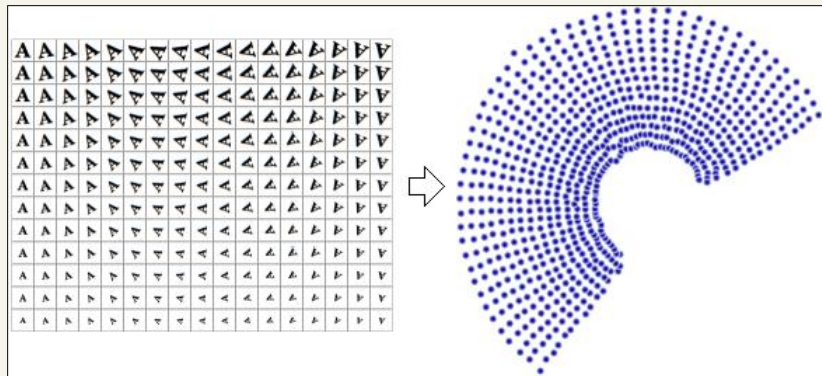
# Manifold

¿Cuál es la dimensión efectiva de los datos?

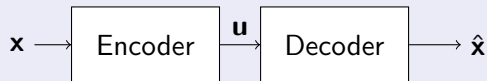


# Manifold

¿Cuál es la dimensión efectiva de los datos?



## Diagrama en bloques de un Autoencoder



# Manifold

¿Cuál es la dimensión efectiva de los datos?

## Objetivo

Hay que entender que el objetivo no es simplemente reconstruir los datos. Sino que es reconstruir los datos a partir de una representación relevante para explicar algún fenómeno o resolver otra tarea. Si no se reconocen patrones en la naturaleza de los datos no hay aprendizaje.

---

Mathematical Snippets - "An unexpected bijection between the real plane and the real line" <https://www.youtube.com/watch?v=XcMZsF4vDbo>

# Manifold

¿Cuál es la dimensión efectiva de los datos?

## Objetivo

Hay que entender que el objetivo no es simplemente reconstruir los datos. Sino que es reconstruir los datos a partir de una representación relevante para explicar algún fenómeno o resolver otra tarea. Si no se reconocen patrones en la naturaleza de los datos no hay aprendizaje.

## Cuidado!

Existen transformaciones  $\mathcal{T} : \mathbb{R}^{d_x} \rightarrow \mathbb{R}$  biyectivas (googlear por ejemplo Teorema de Cantor-Schröder-Bernstein). Pero las representaciones reducidas obtenidas de esta manera pueden no ser interesantes. Hay que tener en cuenta la precisión del computo y, sobre todo, la aplicación en la que se va a utilizar.

---

Mathematical Snippets - "An unexpected bijection between the real plane and the real line" <https://www.youtube.com/watch?v=XcMZsF4vDbo>

# Manifold

## Regularización de autoencoders

Bajo ECM para  
cualquier tipo  
de entrada



Bajo ECM para  
los sets de entre-  
namiento y testeo



Bajo ECM  
solamente  
en el set de  
entrenamiento



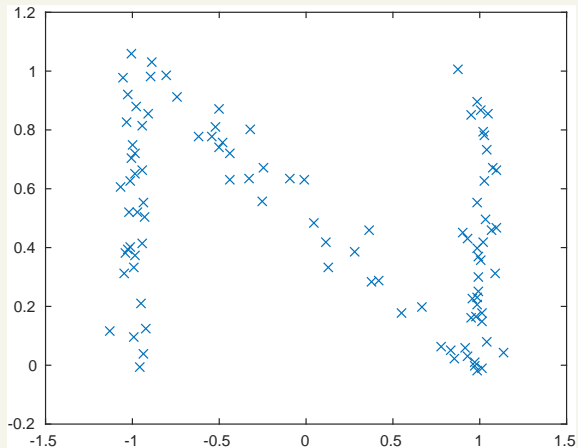
### Objetivo

No quiero memorizar el conjunto de datos ni aprender una transformación biyectiva: Busco aprender el manifold. La regularización en un autoencoder busca balancear estos conceptos.



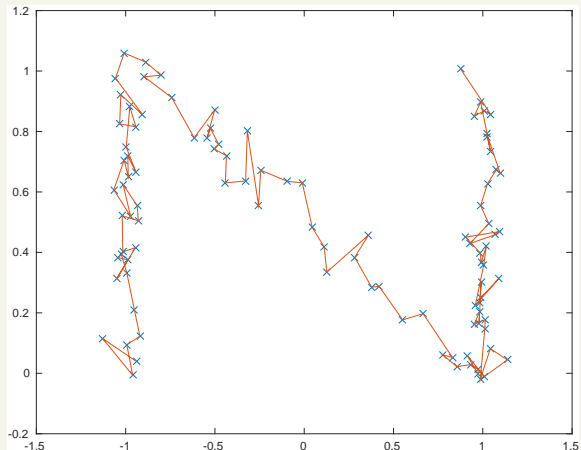
# Manifold

## Regularización de autoencoders



# Manifold

## Regularización de autoencoders



### OVERFITTING

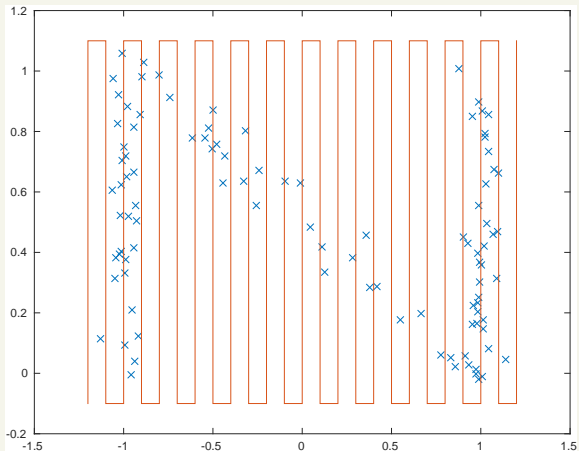
No hay aprendizaje, se están memorizando las muestras.



**Necesito regularización**

# Manifold

## Regularización de autoencoders



### IDENTIDAD

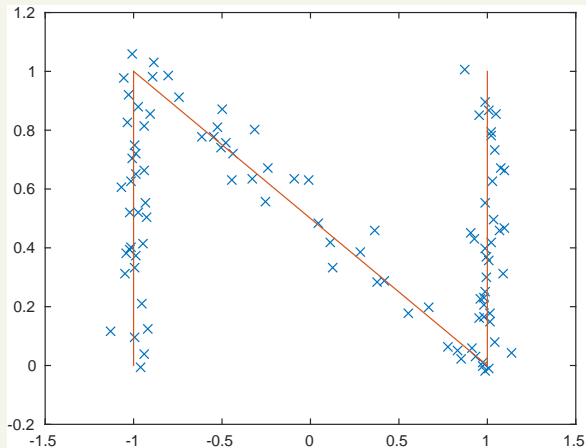
Se está aprendiendo la función identidad y no la naturaleza de los datos.



**Necesito regularización**

# Manifold

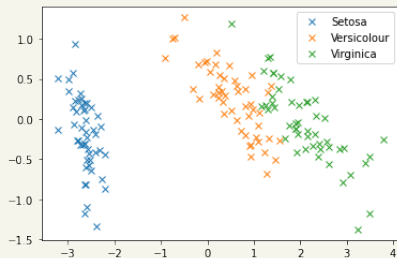
## Regularización de autoencoders



# Algunas Aplicaciones

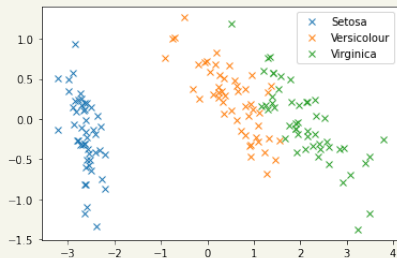
- Para efectuar una inferencia más precisa
- Para pre-procesar los datos
- Para detectar anomalías

# Inferencia

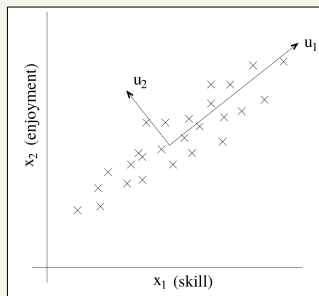


Visualizar en un gráfico 2d o 3d  
para explicar algunos fenómenos  
(iris dataset)

# Inferencia



Visualizar en un gráfico 2d o 3d para explicar algunos fenómenos (iris dataset)

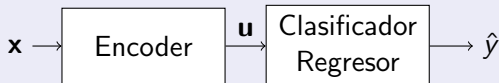


Generar alguna métrica que combine variables muy distintas entre si (radio-controlled helicopters)

# Pre-processing

## Preprocessing: Opción 1

Entrenar el autoencoder y luego usar las muestras en el espacio latente para entrenar el clasificador/regresor.

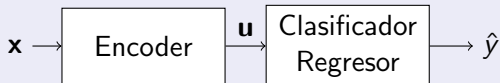




# Pre-processing

## Preprocessing: Opción 1

Entrenar el autoencoder y luego usar las muestras en el espacio latente para entrenar el clasificador/regresor.



## Preprocessing: Opción 2

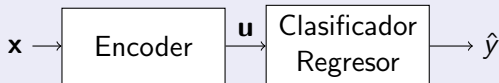
Entrenar el autoencoder y luego usar las reconstrucciones para entrenar el clasificador.



# Pre-processing

## Preprocessing: Opción 1

Entrenar el autoencoder y luego usar las muestras en el espacio latente para entrenar el clasificador/regresor.



## Preprocessing: Opción 2

Entrenar el autoencoder y luego usar las reconstrucciones para entrenar el clasificador.



## Semi-supervise learning

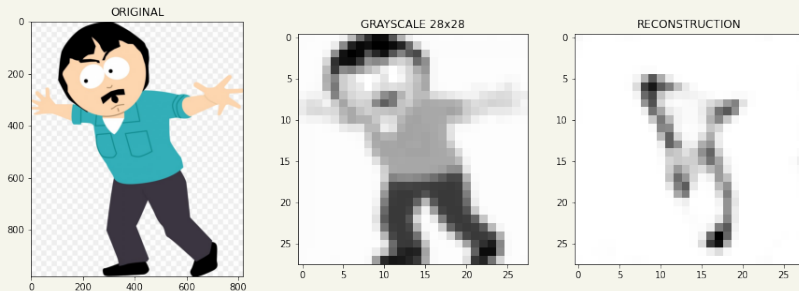
Puedo usar las muestras no supervisadas para entrenar el autoencoder y las supervisadas para el clasificador o el regresor final.

# Detección de anomalías

## Paradigma

Durante el entrenamiento un autoencoder aprende patrones en los datos para reconstruirlos con cierta facilidad. Entonces es de esperar que una muestra que no cumpla los patrones aprendidos sea más difícil de reconstruir.

## EJEMPLO AUTOENCODER ENTRENADO CON MNIST:



# ¿Cuándo usar un autoencoder?

## Clasificación de las aplicaciones

Las aplicaciones de los autoencoders se dividen en dos grupos:

- Las que son relevantes por si mismas.
- Las que son un paso intermedio hacia una tarea de clasificación o regresión. ← **¿Siempre servirá?**

# ¿Cuándo usar un autoencoder?

## Clasificación de las aplicaciones

Las aplicaciones de los autoencoders se dividen en dos grupos:

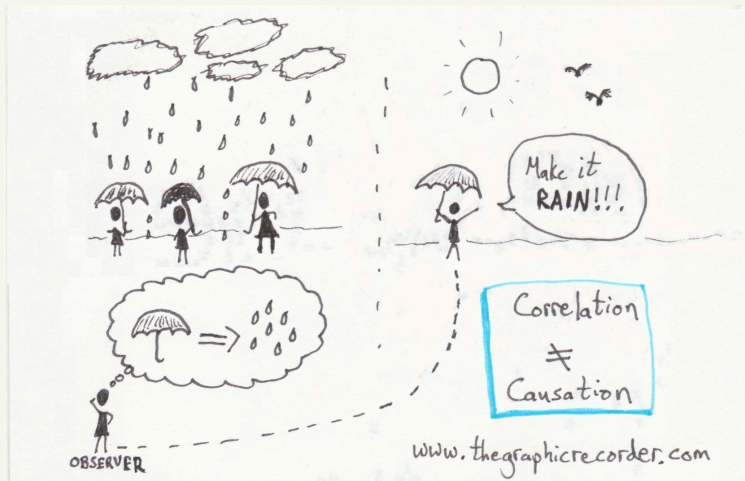
- Las que son relevantes por si mismas.
- Las que son un paso intermedio hacia una tarea de clasificación o regresión. ← **¿Siempre servirá?**

## ¿Que distribución aprende durante el entrenamiento?

Desde un punto de vista probabilístico, el entrenamiento de un algoritmo busca aprender la distribución estadística (total o parcial) de los datos:

- **Aprendizaje supervisado:** Para cada entrada  $x$ , se desea aprender parte de la información contenida en la distribución de una variable objetivo  $Y|X = x$ .
- **Aprendizaje no supervisado:** Toda la información aprendida estará contenida en distribución de los datos  $X$ .

# Hablemos de causalidad



# Causalidad: ¿Quién causa a quién?

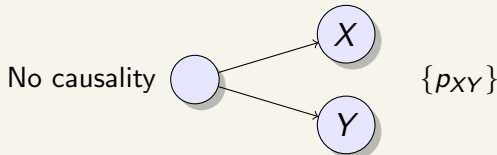
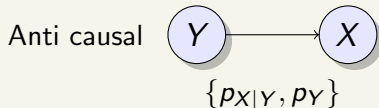
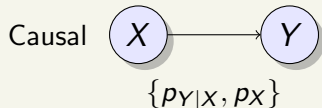
## Independent Causal Mechanisms (ICM) Principle

The causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other. In the probabilistic case, this means that the conditional distribution of each variable given its causes (i.e., its mechanism) does not inform or influence the other mechanisms.

# Causalidad: ¿Quién causa a quién?

## Independent Causal Mechanisms (ICM) Principle

The causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other. In the probabilistic case, this means that the conditional distribution of each variable given its causes (i.e., its mechanism) does not inform or influence the other mechanisms.





## Causalidad: ¿Quién causa a quién?

$$Y = g(X, U) \quad \text{con} \quad X \perp U \quad \text{o} \quad X = g(Y, U) \quad \text{con} \quad Y \perp U$$

# Causalidad: ¿Quién causa a quién?

$$Y = g(X, U) \quad \text{con} \quad X \perp U \quad \text{o} \quad X = g(Y, U) \quad \text{con} \quad Y \perp U$$

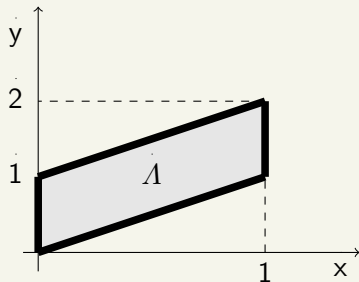
La estadística no basta!

Para toda conjunta  $p_{XY}$  siempre existe  $U \perp X$  y  $g(\cdot, \cdot)$  tal que  $Y = g(X, U)$

# Causalidad: ¿Quién causa a quién?

## La estadística no basta!

Para toda conjunta  $p_{XY}$  siempre existe  $U \perp X$  y  $g(\cdot, \cdot)$  tal que  $Y = g(X, U)$



$$(X, Y) \sim \mathcal{U}(\Lambda)$$

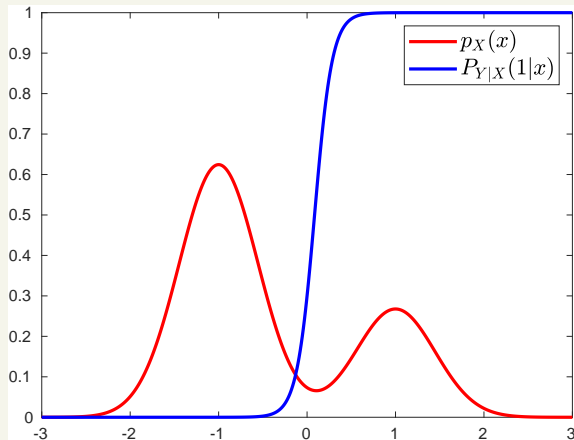
$$Y|X = x \sim \mathcal{U}(x, x + 1) \equiv x + \mathcal{U}(0, 1)$$

$$X|Y = y \sim \begin{cases} \mathcal{U}(0, y) & 0 < y < 1 \\ \mathcal{U}(y - 1, 1) & 1 < y < 2 \end{cases}$$

# Causalidad: ¿Quién causa a quién?

## La estadística no basta!

Para toda conjunta  $p_{XY}$  siempre existe  $U \perp X$  y  $g(\cdot, \cdot)$  tal que  $Y = g(X, U)$



$$Y \sim \text{Cat}\{-1, 1\}$$

$$X|Y = y \sim \mathcal{N}(y, \sigma^2)$$

$$X \sim p_X$$

$$Y|X = x \sim P_{Y|X}(y|x)$$

# Causalidad: ¿Quién causa a quién?

## La estadística no basta!

Para toda conjunta  $p_{XY}$  siempre existe  $U \perp X$  y  $g(\cdot, \cdot)$  tal que  $Y = g(X, U)$

$$\begin{aligned} p_{XY}(x, y) &= e^{-x} \mathbb{1}\{0 < y < x\} \\ &= \underbrace{xe^{-x} \mathbb{1}\{x > 0\}}_{p_X(x)} \underbrace{\frac{1}{x} \mathbb{1}\{0 < y < x\}}_{p_{Y|X}(y|x)} \\ &= \underbrace{e^{-(x-y)} \mathbb{1}\{x > y\}}_{p_{X|Y}(x|y)} \underbrace{e^{-y} \mathbb{1}\{y > 0\}}_{p_Y(y)} \end{aligned}$$

## Causalidad: ¿Quién causa a quién?

### La estadística no basta!

Para toda conjunta  $p_{XY}$  siempre existe  $U \perp X$  y  $g(\cdot, \cdot)$  tal que  $Y = g(X, U)$

$$\begin{aligned} p_{XY}(x, y) &= e^{-x} \mathbb{1}\{0 < y < x\} \\ &= \underbrace{xe^{-x} \mathbb{1}\{x > 0\}}_{p_X(x)} \underbrace{\frac{1}{x} \mathbb{1}\{0 < y < x\}}_{p_{Y|X}(y|x)} \\ &= \underbrace{e^{-(x-y)} \mathbb{1}\{x > y\}}_{p_{X|Y}(x|y)} \underbrace{e^{-y} \mathbb{1}\{y > 0\}}_{p_Y(y)} \end{aligned}$$

$$Y = X \cdot \mathcal{U}(0, 1), \quad X = Y + \mathcal{E}(1)$$

# Causal and Anticausal Learning

## Causal Learning

Desde esta perspectiva, en una configuración causal  $X \rightarrow Y$  no debería ayudarnos conocer  $p_X$  a inferir  $p_{Y|X}$ .

## Solución Óptima

Las decisiones óptimas  $\hat{P}_\theta(y|x) = P_{Y|X}(y|x)$ ,  $\varphi_\theta(x) = \mathbb{E}[Y|X = x]$  y  $\phi_\theta(x) = \arg \max_y P_{Y|X}(y|x)$  no dependen de la marginal. Es decir, la solución es la misma por más que cambie la marginal  $p_X$ .

# Causal and Anticausal Learning

## Causal Learning

Desde esta perspectiva, en una configuración causal  $X \rightarrow Y$  no debería ayudarnos conocer  $p_X$  a inferir  $p_{Y|X}$ .

## Solución Óptima

Las decisiones óptimas  $\hat{P}_\theta(y|x) = P_{Y|X}(y|x)$ ,  $\varphi_\theta(x) = \mathbb{E}[Y|X = x]$  y  $\phi_\theta(x) = \arg \max_y P_{Y|X}(y|x)$  no dependen de la marginal. Es decir, la solución es la misma por más que cambie la marginal  $p_X$ .

## Igual un poquito ayuda

$$\arg \min_{\theta \in \Theta} \mathbb{E}[-\log \hat{P}_\theta(Y|X)] = \arg \min_{\theta \in \Theta} \mathbb{E}_{p_X} [\text{KL}(P_{Y|X}(\cdot|X) \| \hat{P}_\theta(\cdot|X))]$$

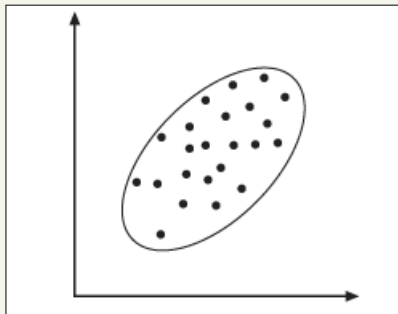
$$\arg \min_{\theta \in \Theta} \mathbb{E}[(Y - \varphi_\theta(X))^2] = \arg \min_{\theta \in \Theta} \mathbb{E}_{p_X} [(\varphi_\theta(X) - \mathbb{E}[Y|X])^2]$$

$$\arg \min_{\theta \in \Theta} \mathbb{P}(Y \neq \phi_\theta(X)) = \arg \min_{\theta \in \Theta} \mathbb{E}_{p_X} \left[ \max_y P_{Y|X}(y|X) - P_{Y|X}(\phi_\theta(X)|X) \right]$$



# Principal Components Analysis

Reducción lineal

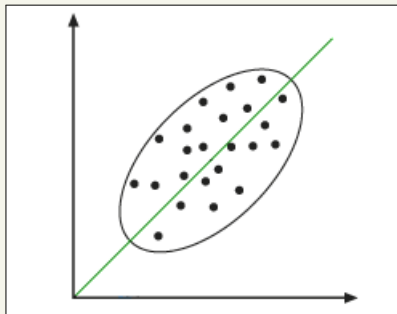


---

Lectura recomendada: *Andrew Ng* - "Lecture notes: Principal components analysis".

# Principal Components Analysis

Reducción lineal



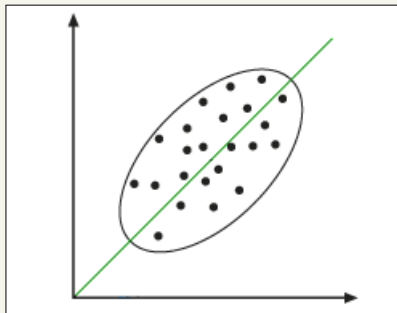
---

Lectura recomendada: *Andrew Ng* - "Lecture notes: Principal components analysis".

# Principal Components Analysis

## Reducción lineal

### PASO 1: Normalizar



$$\tilde{x}_j^{(i)} = \frac{x_j^{(i)} - \mu_j}{\sigma_j}$$

con

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_j^{(i)}$$

$$\sigma_j^2 = \frac{1}{n} \sum_{i=1}^n (x_j^{(i)} - \mu_j)^2$$

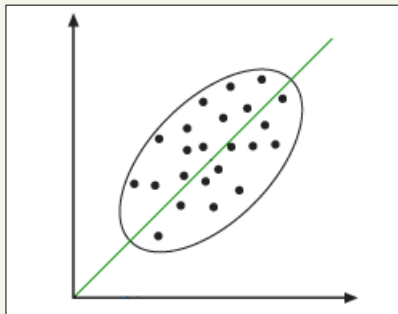
---

Lectura recomendada: *Andrew Ng* - "Lecture notes: Principal components analysis".

# Principal Components Analysis

## Reducción lineal

### PASO 1: Normalizar



$$\tilde{x}_j^{(i)} = \frac{x_j^{(i)} - \mu_j}{\sigma_j}$$

con

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_j^{(i)}$$

$$\sigma_j^2 = \frac{1}{n} \sum_{i=1}^n (x_j^{(i)} - \mu_j)^2$$

### PASO 2: Buscar el principal autovector $\mathbf{v}_1$

$$\min_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \sum_{i=1}^n \|\tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1\|^2 \quad \text{con} \quad \langle \tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1; \mathbf{v}_1 \rangle = 0$$

Lectura recomendada: *Andrew Ng* - "Lecture notes: Principal components analysis".

# Principal Components Analysis

## Algunas cuentas

Condicion de ortogonalidad:

$$\langle \tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1; \mathbf{v}_1 \rangle = 0 \quad \rightarrow \quad \langle \tilde{\mathbf{x}}^{(i)}; \mathbf{v}_1 \rangle = \alpha_i \|\mathbf{v}_1\|^2 = \alpha_i$$

# Principal Components Analysis

## Algunas cuentas

Condicion de ortogonalidad:

$$\langle \tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1; \mathbf{v}_1 \rangle = 0 \quad \rightarrow \quad \langle \tilde{\mathbf{x}}^{(i)}; \mathbf{v}_1 \rangle = \alpha_i \|\mathbf{v}_1\|^2 = \alpha_i$$

Optimización:

$$\min_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \sum_{i=1}^n \|\tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1\|^2 = \min_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \sum_{i=1}^n \|\tilde{\mathbf{x}}^{(i)}\|^2 - \alpha_i^2$$

# Principal Components Analysis

## Algunas cuentas

Condición de ortogonalidad:

$$\langle \tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1; \mathbf{v}_1 \rangle = 0 \quad \rightarrow \quad \langle \tilde{\mathbf{x}}^{(i)}; \mathbf{v}_1 \rangle = \alpha_i \|\mathbf{v}_1\|^2 = \alpha_i$$

Optimización:

$$\min_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \sum_{i=1}^n \|\tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1\|^2 = \min_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \sum_{i=1}^n \|\tilde{\mathbf{x}}^{(i)}\|^2 - \alpha_i^2$$

$$\max_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \frac{1}{n} \sum_{i=1}^n \langle \tilde{\mathbf{x}}^{(i)}; \mathbf{v}_1 \rangle^2 = \max_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \mathbf{v}_1^T \underbrace{\left( \frac{1}{n} \sum_{i=1}^n \tilde{\mathbf{x}}^{(i)} (\tilde{\mathbf{x}}^{(i)})^T \right)}_{\Sigma} \mathbf{v}_1$$

# Principal Components Analysis

Algunas cuentas

$$J(\mathbf{v}_1) = \mathbf{v}_1^T \Sigma \mathbf{v}_1 - \lambda \left( \mathbf{v}_1^T \mathbf{v}_1 - 1 \right)$$

---

Lectura recomendada: *Petersen and Pedersen* - “Matrix Cookbook”.



# Principal Components Analysis

Algunas cuentas

$$J(\mathbf{v}_1) = \mathbf{v}_1^T \Sigma \mathbf{v}_1 - \lambda \left( \mathbf{v}_1^T \mathbf{v}_1 - 1 \right)$$

$$\nabla J(\mathbf{v}_1) = 2\Sigma \mathbf{v}_1 - 2\lambda \mathbf{v}_1 = 0$$

---

Lectura recomendada: *Petersen and Pedersen* - "Matrix Cookbook".

# Principal Components Analysis

Algunas cuentas

$$J(\mathbf{v}_1) = \mathbf{v}_1^T \Sigma \mathbf{v}_1 - \lambda (\mathbf{v}_1^T \mathbf{v}_1 - 1)$$

$$\nabla J(\mathbf{v}_1) = 2\Sigma \mathbf{v}_1 - 2\lambda \mathbf{v}_1 = 0$$

$$\Sigma \mathbf{v}_1 = \lambda \mathbf{v}_1 \quad \rightarrow \quad \mathbf{v}_1 \text{ es AVE de } \Sigma \text{ y } \lambda \text{ es AVA}$$

---

Lectura recomendada: *Petersen and Pedersen* - "Matrix Cookbook".

# Principal Components Analysis

## Algunas cuentas

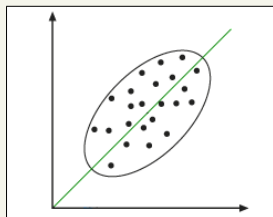
$$J(\mathbf{v}_1) = \mathbf{v}_1^T \Sigma \mathbf{v}_1 - \lambda (\mathbf{v}_1^T \mathbf{v}_1 - 1)$$

$$\nabla J(\mathbf{v}_1) = 2\Sigma \mathbf{v}_1 - 2\lambda \mathbf{v}_1 = 0$$

$$\Sigma \mathbf{v}_1 = \lambda \mathbf{v}_1 \quad \rightarrow \quad \mathbf{v}_1 \text{ es AVE de } \Sigma \text{ y } \lambda \text{ es AVA}$$

El problema de optimización pasa a ser de la forma

$$\max_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \mathbf{v}_1^T \Sigma \mathbf{v}_1 = \max_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \lambda(\mathbf{v}_1) \quad \rightarrow \quad \text{Máximo AVA}$$



Lectura recomendada: *Petersen and Pedersen* - "Matrix Cookbook".

# Principal Components Analysis

## Reducción y Reconstrucción

### Componentes principales

Este procedimiento se puede repetir para encontrar el 2do, 3er, etc. componente principal. El resultado son el 2do, 3er, etc autovalor con su autovector como dirección.

# Principal Components Analysis

## Reducción y Reconstrucción

### Componentes principales

Este procedimiento se puede repetir para encontrar el 2do, 3er, etc. componente principal. El resultado son el 2do, 3er, etc autovalor con su autovector como dirección.

### Sobre los autovalores

El porcentaje de energía perdida puede medirse por la proporción de autovalores despreciados.

- **V**: Matriz de autovectores más relevantes.
- **x**: Variable de entrada a procesar (ya normalizada).
- **u**: Variable latente.
- **$\hat{x}$** : Reconstrucción

$$\mathbf{u} = \mathbf{V} \cdot \mathbf{x}, \quad \hat{\mathbf{x}} = \mathbf{V}^T \cdot \mathbf{u}$$

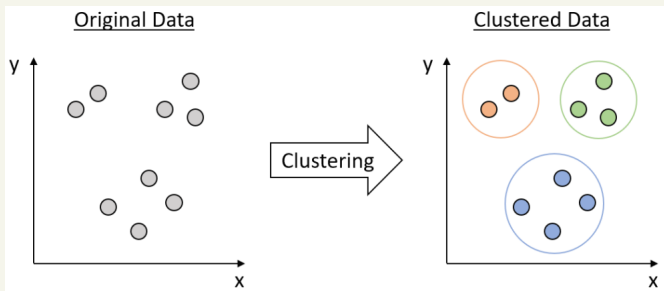
# Outline

- 1 Autoencoders
- 2 Principal Components Analysis (PCA)
- 3 K-Means**
- 4 Algoritmo EM
- 5 Factor Analysis

# Clustering

## Clustering

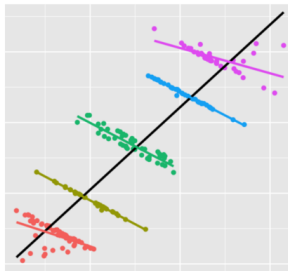
Estos algoritmos son la versión no supervisada de la clasificación. Su objetivo es agrupar muestras de manera de tener un mayor entendimiento del *manifold*.



# Motivación: Paradoja de Simpson

## Paradoja de Simpson

La paradoja de Simpson se da cuando dos (o más) variables tienen una correlación hacia un sentido pero al agrupar los datos se ve que, en cada cluster, la correlación posee en realidad el sentido opuesto.





# Paradoja de Simpson: Covid-19 Case Fatality Rates (CFR)

Edad	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70-79	≥ 80	Total
Italia	0% (0/43)	0% (0/85)	0% (0/296)	0% (0/470)	0.1% (1/891)	0.2% (3/1453)	2.5% (37/1471)	6.4% (114/1785)	13.2% (202/1532)	4.4% (357/8026)
China	0% (0/0)	0.2% (1/549)	0.2% (7/3619)	0.2% (18/7600)	0.4% (38/8571)	1.3% (130/10008)	3.6% (309/8583)	8% (312/3918)	14.8% (208/1408)	2.3% (1023/44672)

---

Julius von Kugelgen, Luigi Gresele and Bernhard Scholkopf "Simpson's paradox in Covid-19 case fatality rates: A mediation analysis of age-related causal effects" IEEE Transactions on Artificial Intelligence 2021.

# Algoritmo K-Means

## K-means

Algoritmo de clustering para agrupar los datos en  $K$  clusters (previamente definidos). Se basa en encontrar, de forma iterativa, los *centroides* de cada clase y asignar cada muestra al centroide más cercano.

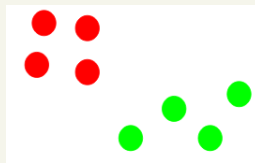
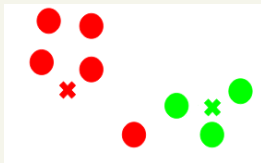
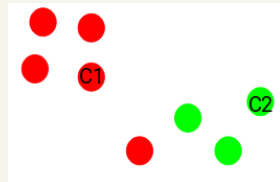
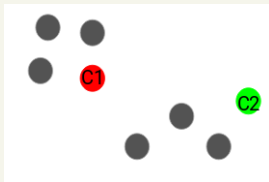
---

### Algorithm 1 K-means

---

- 1: **procedure** KMEANS( $X, K$ )  
    **Input:**  $X \in \mathbb{R}^{n \times d_x}$  matriz de datos y  $K$  número de clusters.  
    **Output:**  $\mu \in \mathbb{R}^{K \times d_x}$  centroides e  $y \in \{1, \dots, K\}^n$  etiquetas.
  - 2:     Inicializar  $\mu$  con el valor de  $K$  columnas de  $X$  elegida al azar.
  - 3:     **repeat**
  - 4:          $y[i] = \arg \min_k \|X[i, :] - \mu[k, :]\|$  ▷ Con  $i = 1, \dots, n$ .
  - 5:          $\mu[k, :] = \mathbb{E}[X[y == k, :]]$  ▷ Con  $k = 1, \dots, K$
  - 6:     **until** convergencia
  - 7:     **Return:**  $\mu$  e  $y$
  - 8: **end procedure**
-

# Algoritmo K-Means



# Outline

- 1 Autoencoders
- 2 Principal Components Analysis (PCA)
- 3 K-Means
- 4 Algoritmo EM**
- 5 Factor Analysis

# Máxima Verosimilitud

## Algoritmos de Máxima Verosimilitud

La minimización de la *cross-entropy* equivale a encontrar algoritmos de máxima verosimilitud. El problema es que estos son muchas veces analíticamente intratables y computacionalmente muy pesados de tratar (ej. mezcla de gaussianas).

$$\hat{\theta} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \log p(X_i | \theta)$$

si la densidad no es sencilla se pone pesado computacionalmente

## Variables no observable

Sea  $Z$  una variable no observable del problema con densidad condicional  $p(z|x, \theta)$ , y sea  $\mathcal{P}$  la familia de todas las posibles densidades condicionales de  $Z|X = x$ . Luego, el estimador de MV puede reescribirse como:

$$\begin{aligned} \hat{\theta} &= \arg \max_{\theta \in \Theta} \max_{q \in \mathcal{P}} \sum_{i=1}^n [\log p(X_i | \theta) - \text{KL}(q(\cdot | X_i) \| p(\cdot | X_i, \theta))] \\ &= \arg \max_{\theta \in \Theta} \max_{q \in \mathcal{P}} \text{ELBO}(\theta, q) \end{aligned}$$

REF1

primero busco el maz con respecto a q, lo dejo fijo y repito con respecto a tita, loo dejo fijo y vuelvo a q, asi sucesivamente

# Algoritmo EM

## Algoritmo Expectation - Maximization

El algoritmo EM consiste en inicializar en algún valor  $\theta_0$  e iterar entre:

- $q^{(t)} = \arg \max_{q \in \mathcal{P}} \text{ELBO}(\theta^{(t-1)}, q)$  (Expectation)
- $\theta^{(t)} = \arg \max_{\theta \in \Theta} \text{ELBO}(\theta, q^{(t)})$  (Maximization)

# Algoritmo EM

## Algoritmo Expectation - Maximization

El algoritmo EM consiste en inicializar en algún valor  $\theta_0$  e iterar entre:

- $q^{(t)} = \arg \max_{q \in \mathcal{P}} \text{ELBO}(\theta^{(t-1)}, q)$  (Expectation)
- $\theta^{(t)} = \arg \max_{\theta \in \Theta} \text{ELBO}(\theta, q^{(t)})$  (Maximization)

## Expectación

El paso *Expectation* puede simplificarse a la relación

$q^{(t)}(z|x) = p(z|x, \theta^{(t-1)})$ . Es decir:

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \left[ \log p(X_i|\theta) - \text{KL} \left( p(\cdot|X_i, \theta^{(t-1)}) \| p(\cdot|X_i, \theta) \right) \right]$$

# Algoritmo EM

## Maximización

El paso *Maximization* puede simplificarse reescribiendo cada sumando como  
es como un bayes

$$\log p(x|\theta) - \text{KL}(q(\cdot|x) || p(\cdot|x, \theta)) = H(q(\cdot|x)) + \mathbb{E}_q [\log p(x, Z|\theta) | X = x]$$

REF2

donde la entropía no depende de  $\theta$ . Es decir,

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i, Z|\theta) | X_i]$$



# Algoritmo EM

## Maximización

El paso *Maximization* puede simplificarse reescribiendo cada sumando como

$$\log p(x|\theta) - \text{KL}(q(\cdot|x) || p(\cdot|x, \theta)) = H(q(\cdot|x)) + \mathbb{E}_q [\log p(x, Z|\theta) | X = x]$$

donde la entropía no depende de  $\theta$ . Es decir,

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i, Z|\theta) | X_i]$$

## Teorema: Monotonía

En el algoritmo EM ocurre que

$$\sum_{i=1}^n \log p(X_i | \theta^{(t)}) \geq \sum_{i=1}^n \log p(X_i | \theta^{(t-1)})$$

REF3

Hint: Expectación + KL  $\geq 0$ .

# Algoritmo EM para mezcla de gaussianas

## Definición del problema

Si  $Z \sim \text{Cat}(\{c_1, \dots, c_K\})$  y  $X|Z = k \sim \mathcal{N}(\mu_k, \Sigma_k)$ , está claro que  $X$  es una mezcla de gaussianas. Sea  $\theta = \{c_k, \mu_k, \Sigma_k\}_{k=1}^K$ , se desea estimar estos parámetros (de forma no supervisada, es decir siendo  $Z$  no observable). El estimador de máxima verosimilitud es intratable y por eso recurrimos al algoritmo EM.

# Algoritmo EM para mezcla de gaussianas

## Definición del problema

Si  $Z \sim \text{Cat}(\{c_1, \dots, c_K\})$  y  $X|Z = k \sim \mathcal{N}(\mu_k, \Sigma_k)$ , está claro que  $X$  es una mezcla de gaussianas. Sea  $\theta = \{c_k, \mu_k, \Sigma_k\}_{k=1}^K$ , se desea estimar estos parámetros (de forma no supervisada, es decir siendo  $Z$  no observable). El estimador de máxima verosimilitud es intratable y por eso recurrimos al algoritmo EM.

## Expectación

El paso de expectación es simplemente elegir:

$$q(k|x) = p(k|x, \theta) = \frac{c_k \cdot |\Sigma_k|^{-1/2} \cdot e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}}{\sum_{m=1}^K c_m \cdot |\Sigma_m|^{-1/2} \cdot e^{-\frac{1}{2}(x-\mu_m)^T \Sigma_m^{-1}(x-\mu_m)}}$$

# Algoritmo EM para mezcla de gaussianas

## Maximización

Dado un  $q$ , se desea maximizar:

$$\max_{\theta} \sum_{i=1}^n \mathbb{E}_q [\log p(X_i, Z|\theta)|X_i] \quad \text{s.t.} \quad \sum_{k=1}^K c_k = 1$$

Es decir que, utilizando multiplicadores de Lagrange, la función a derivar e igualar a cero es:

$$\mathcal{L}(\theta) = \sum_{i=1}^n \sum_{k=1}^K q(k|x_i) \left[ \log c_k - \frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) \right] + \lambda \left( 1 - \sum_{k=1}^K c_k \right)$$

# Algoritmo EM para mezcla de gaussianas

## Derivada respecto a $c_k$

Igualamos a cero la derivada respecto a  $c_k$  y usamos que  $\sum_{k=1}^K c_k = 1$ .

$$\frac{\partial \mathcal{L}(\theta)}{\partial c_k} = \left( \sum_{i=1}^n \frac{q(k|x_i)}{c_k} \right) - \lambda = 0$$

$$\Rightarrow c_k = \frac{1}{\lambda} \sum_{i=1}^n q(k|x_i)$$

$$\Rightarrow c_k = \frac{1}{n} \sum_{i=1}^n q(k|x_i)$$

# Algoritmo EM para mezcla de gaussianas

## Derivada respecto a $\mu_k$

Igualamos a cero (vector) la derivada respecto a  $\mu_k$ .

$$\begin{aligned}\frac{\partial \mathcal{L}(\theta)}{\partial \mu_k} &= \sum_{i=1}^n q(k|x_i) \Sigma_k^{-1} (x_i - \mu_k) = 0 \\ \Rightarrow \mu_k &= \frac{\sum_{i=1}^n q(k|x_i) \cdot x_i}{\sum_{i=1}^n q(k|x_i)}\end{aligned}$$

# Algoritmo EM para mezcla de gaussianas

## Derivada respecto a $\mu_k$

Igualamos a cero (vector) la derivada respecto a  $\mu_k$ .

$$\begin{aligned}\frac{\partial \mathcal{L}(\theta)}{\partial \mu_k} &= \sum_{i=1}^n q(k|x_i) \Sigma_k^{-1} (x_i - \mu_k) = 0 \\ \Rightarrow \mu_k &= \frac{\sum_{i=1}^n q(k|x_i) \cdot x_i}{\sum_{i=1}^n q(k|x_i)}\end{aligned}$$

## Derivada respecto a $\Sigma_k$

el paso M del algoritmo EM asume conocida q y busca los parametros

Igualamos a cero (matriz) la derivada respecto a  $\Sigma_k$ .

$$\begin{aligned}\frac{\partial \mathcal{L}(\theta)}{\partial \Sigma_k} &= \sum_{i=1}^n q(k|x_i) \left[ -\frac{1}{2} \Sigma_k^{-1} + \frac{1}{2} \Sigma_k^{-1} (x_i - \mu_k)(x_i - \mu_k)^T \Sigma_k^{-1} \right] = 0 \\ \Rightarrow \Sigma_k &= \frac{\sum_{i=1}^n q(k|x_i) \cdot (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^n q(k|x_i)}\end{aligned}$$

Petersen and Pedersen - "Matrix Cookbook".

# Algoritmo EM para máximo a posteriori

## Estimador puntual con enfoque Bayesiano

Si modelamos  $\theta$  como variable aleatoria y suponemos alguna distribución *a priori*  $\pi(\theta)$ , definimos el estimador MAP como:

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta \in \Theta} \log p(\theta | \underline{X}) \\ &= \arg \max_{\theta \in \Theta} \log \pi(\theta) + \sum_{i=1}^n \log p(X_i | \theta) \\ &= \arg \max_{\theta \in \Theta} \log \pi(\theta) + \max_{q \in \mathcal{P}} \text{ELBO}(\theta, q)\end{aligned}$$

## M-step

La prior solo modifica la maximización:

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \log \pi(\theta) + \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i, Z | \theta) | X_i]$$

Hasta ejercicios 3.5 (teórico) y 3.6



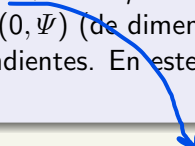
# Outline

- 1 Autoencoders
- 2 Principal Components Analysis (PCA)
- 3 K-Means
- 4 Algoritmo EM
- 5 Factor Analysis**

# Aplicación de EM: Factor Analysis

## Factor Analysis

Al igual que PCA, el algoritmo EM puede utilizarse para reducir la dimensión. El modelo consiste en suponer que los *features* se puede descomponer en factores:  $X = \mu + W \cdot Z + \epsilon$  con  $\mu \in \mathbb{R}^{d_x}$ ,  $W \in \mathbb{R}^{d_x \times d_z}$ ,  $Z \sim \mathcal{N}(0, I)$  (de dimensión  $d_z$ ) y  $\epsilon \sim \mathcal{N}(0, \Psi)$  (de dimensión  $d_x$ ) con  $\Psi$  una matriz diagonal y con  $Z$  y  $\epsilon$  independientes. En este caso,  $\theta = \{\mu, W, \Psi\}$ .



$Z \parallel \epsilon$

variable de menor dimension

# Aplicación de EM: Factor Analysis

## Factor Analysis

Al igual que PCA, el algoritmo EM puede utilizarse para reducir la dimensión. El modelo consiste en suponer que los *features* se puede descomponer en factores:  $X = \mu + W \cdot Z + \epsilon$  con  $\mu \in \mathbb{R}^{d_x}$ ,  $W \in \mathbb{R}^{d_x \times d_z}$ ,  $Z \sim \mathcal{N}(0, I)$  (de dimensión  $d_z$ ) y  $\epsilon \sim \mathcal{N}(0, \Psi)$  (de dimensión  $d_x$ ) con  $\Psi$  una matriz diagonal y con  $Z$  y  $\epsilon$  independientes. En este caso,  $\theta = \{\mu, W, \Psi\}$ .

cte



## Expectación

$Z$  es una normal,  $X|Z$  es una combinación de  $Z$ , es una normal  
 $X|Z \sim \mathcal{N}(\mu + w_z, \Psi)$

Dado que la conjunta entre  $(X, Z)$  es una normal multivariada, la condicional de  $Z|X = x$  también lo será. Es decir que  $p(z|x, \theta)$  se caracterizará por una media (recta función de  $x$ ) y una matriz de covarianza constante (no depende de  $x$ ). paso E: Calcular  $Z|X=x \sim \mathcal{N}()$

# Aplicación: Factor Analysis

## Maximización

Como la marginal  $p(Z)$  no depende de  $\theta$ , la maximización se reduce a:

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i | Z, \theta) | X_i]$$

$\log p(x, z | \text{tita}) = \cancel{\log p(z)} + \log(p(x|z, \text{tita}))$

# Aplicación: Factor Analysis

## Maximización

Como la marginal  $p(Z)$  no depende de  $\theta$ , la maximización se reduce a:

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i|Z, \theta)|X_i]$$

## Encoder - Decoder

Una vez entrenado el algoritmo, se utiliza  $\mathbb{E}[Z|X = x, \theta]$  como *encoder* y  $\mathbb{E}[X|Z = z, \theta]$  como *decoder*.

$$\mathbb{E}[X|Z = z] = \mu + Wz$$

# Aplicación: Factor Analysis

## Maximización

Como la marginal  $p(Z)$  no depende de  $\theta$ , la maximización se reduce a:

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i|Z, \theta)|X_i]$$

## Encoder - Decoder

Una vez entrenado el algoritmo, se utiliza  $\mathbb{E}[Z|X = x, \theta]$  como *encoder* y  $\mathbb{E}[X|Z = z, \theta]$  como *decoder*.

## Probabilistic PCA

Si en lugar de pedir que  $\Psi$  sea diagonal se pide que todas las varianzas coincidan  $\Psi = \sigma^2 \mathbf{I}$ , se obtiene un algoritmo conocido como Probabilistic PCA.

# Esquema de despejes de ecuaciones

- 1 Hallar la distribución condicional de  $X|Z = z$ .
- 2 Hallar la distribución marginal de  $X$ .  $X \sim N(\mu, w.w^T + \Psi)$
- 3 Hallar la distribución conjunta de  $(X, Z)$ .
- 4 E-Step (los parámetros se asumen conocidos y se calcula la distribución): Hallar la distribución condicional de  $Z|X = x$ . Bautizar a los principales momentos:  $m_i = \mathbf{E}[Z|X = x_i]$ ,  $\Sigma_i = \text{var}(Z|X = x_i)$ .
- 5 M-Step (la distribución anterior se asume conocida y se calculan los parámetros): Calcular la esperanza de  $\log(p(X|Z, \theta))$ , bajo  $Z|X$ .
- 6 Plantear la función a maximizar.
- 7 Para simplificar el análisis asumir que  $\mu = \frac{1}{n} \sum_{i=1}^n x_i$  (no varía en las iteraciones). ¿Puede decir algo sobre  $\sum_{i=1}^n m_i$ ?
- 8 Derivar respecto de  $W$  e igualar a cero. Despejar  $W$ .
- 9 Derivar respecto de  $\Psi$  e igualar a cero. Despejar  $\Psi$ .

---

Petersen and Pedersen - "Matrix Cookbook" (tenerlo a mano en todo el momento).  
Andrew Ng - "Lecture notes: Factor Analysis" (chequear resultados).