# Aprendizaje no Supervisado

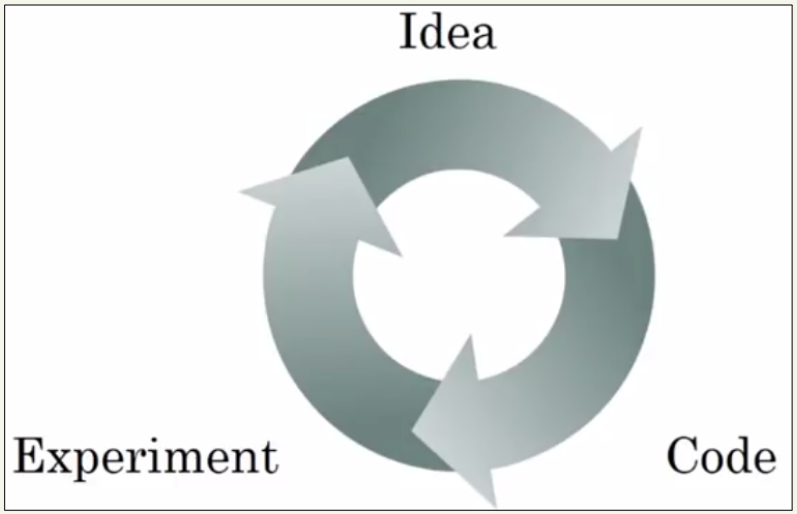
**Introducción a la Inteligencia Artificial**

# Agenda

1. Autoencoders
2. Principal Components Analysis (PCA)
3. K-Means

# Aprendizaje Estadístico

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

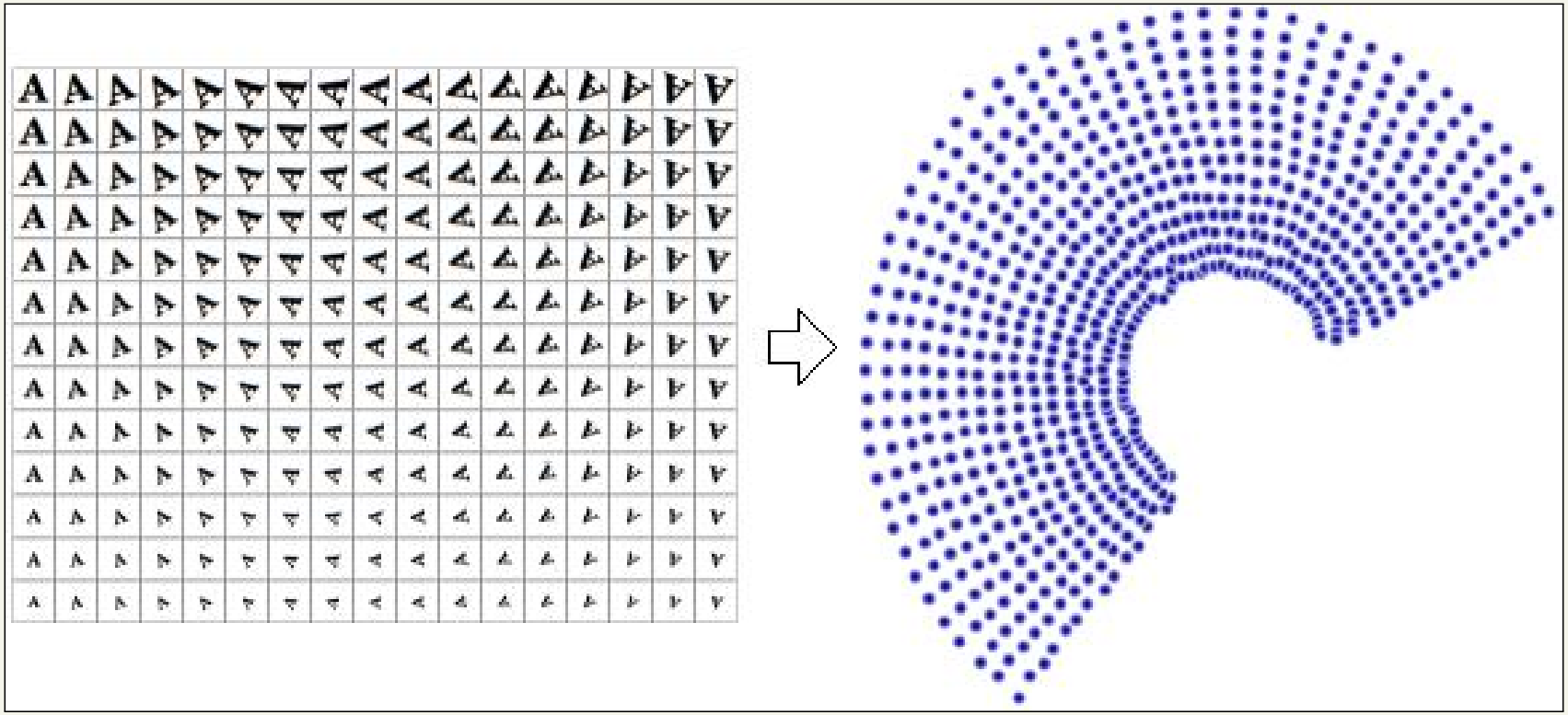



## TIPOS DE APRENDIZAJES

- 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.

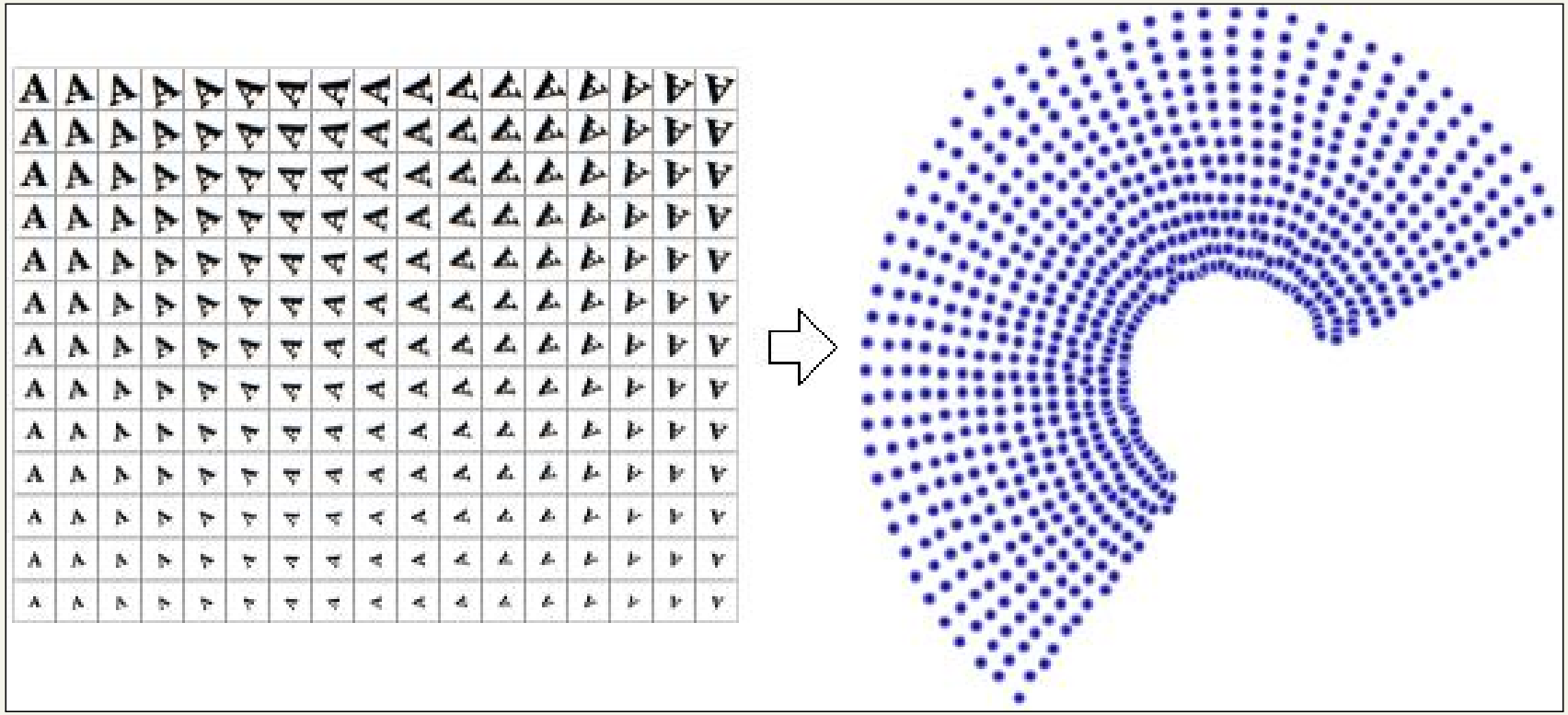
# Manifold

## ¿Cuál es la dimensión efectiva de los datos?

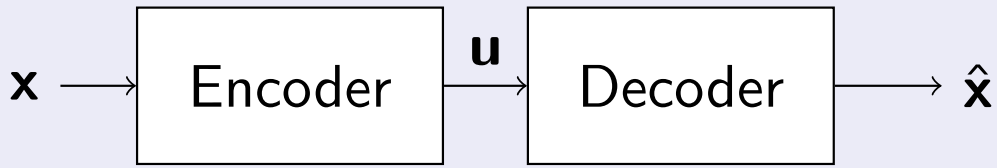


# Manifold

## ¿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 $\mathcal{T} : \mathbb{R}^{d_x} \to \mathbb{R}$ 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

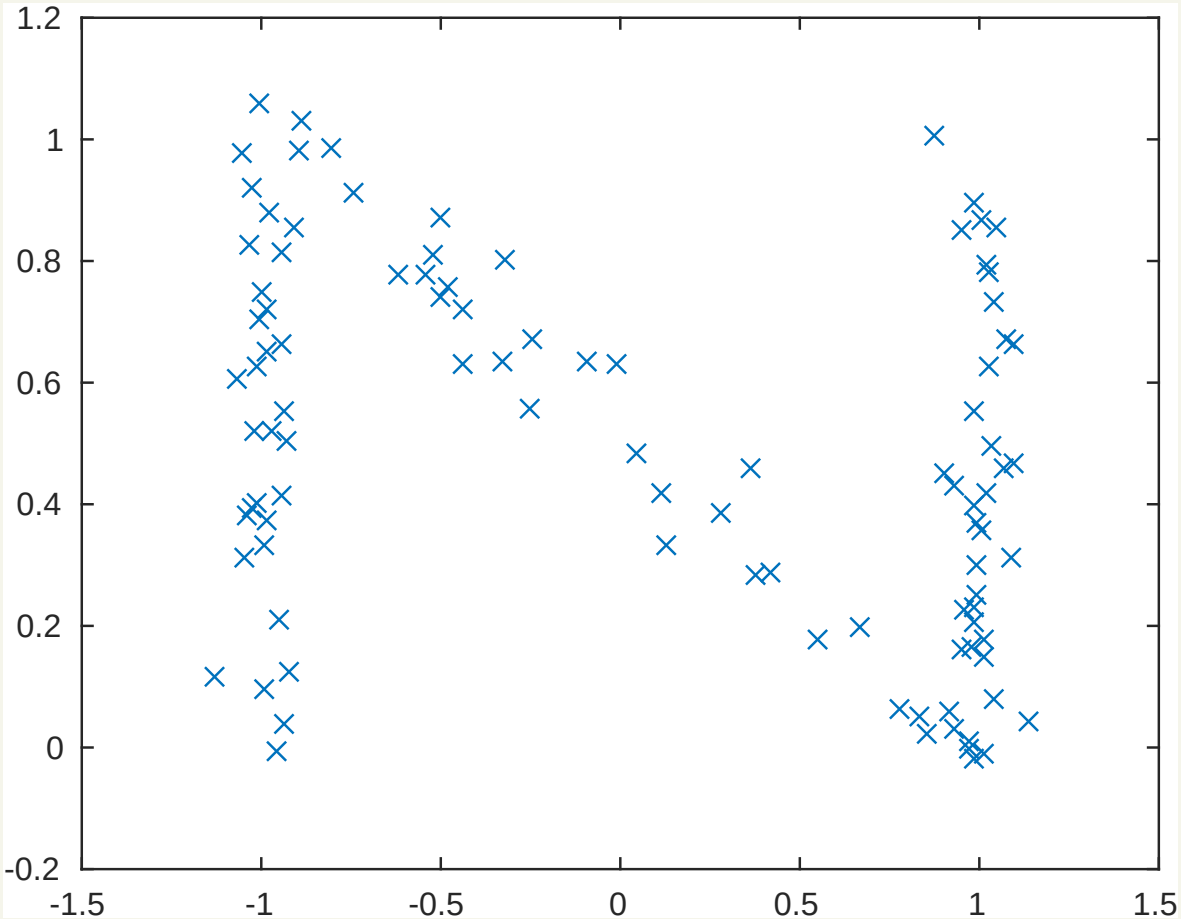
| Bajo ECM para cualquier tipo de entrada | Bajo ECM para los sets de entrenamiento y testeo | Bajo ECM solamente en el set de entrenamiento |
|---|---|---|
| ✗ | ✓ | ✗ |

**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.

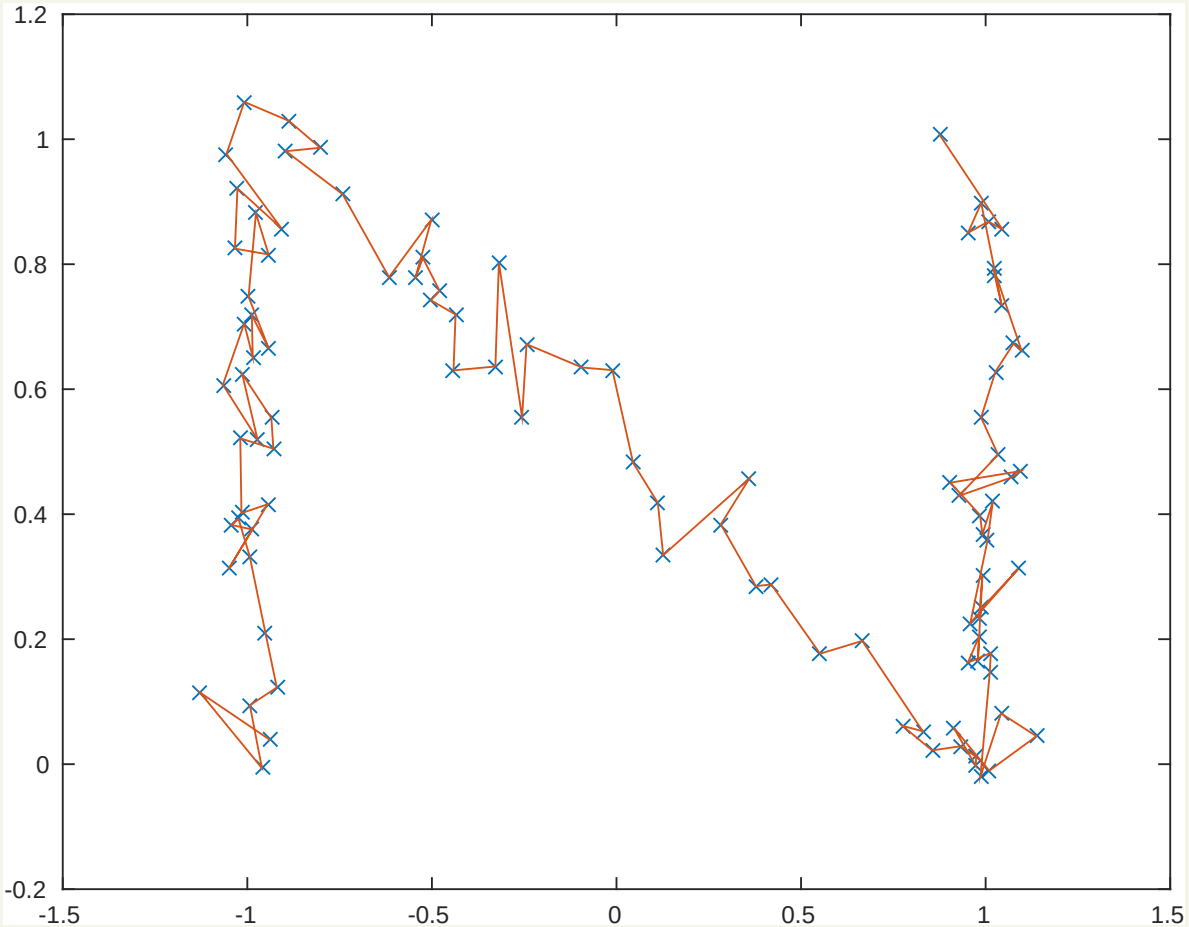
# Manifold

## Regularización de autoencoders



# Manifold

## Regularización de autoencoders



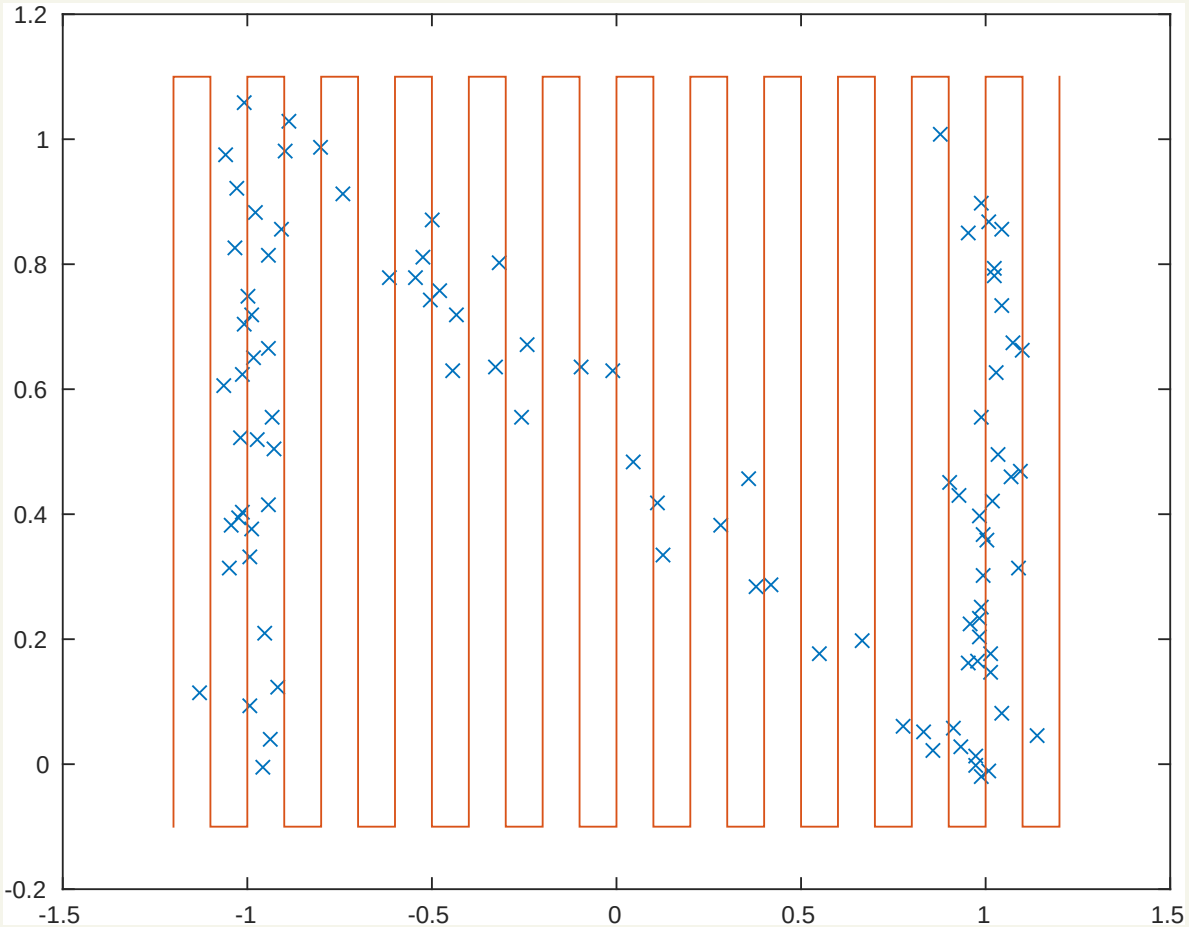

**OVERFITTING**

No hay aprendizaje, se están memorizando las muestras.

**Necesito regularización**

# Manifold

## Regularización de autoencoders



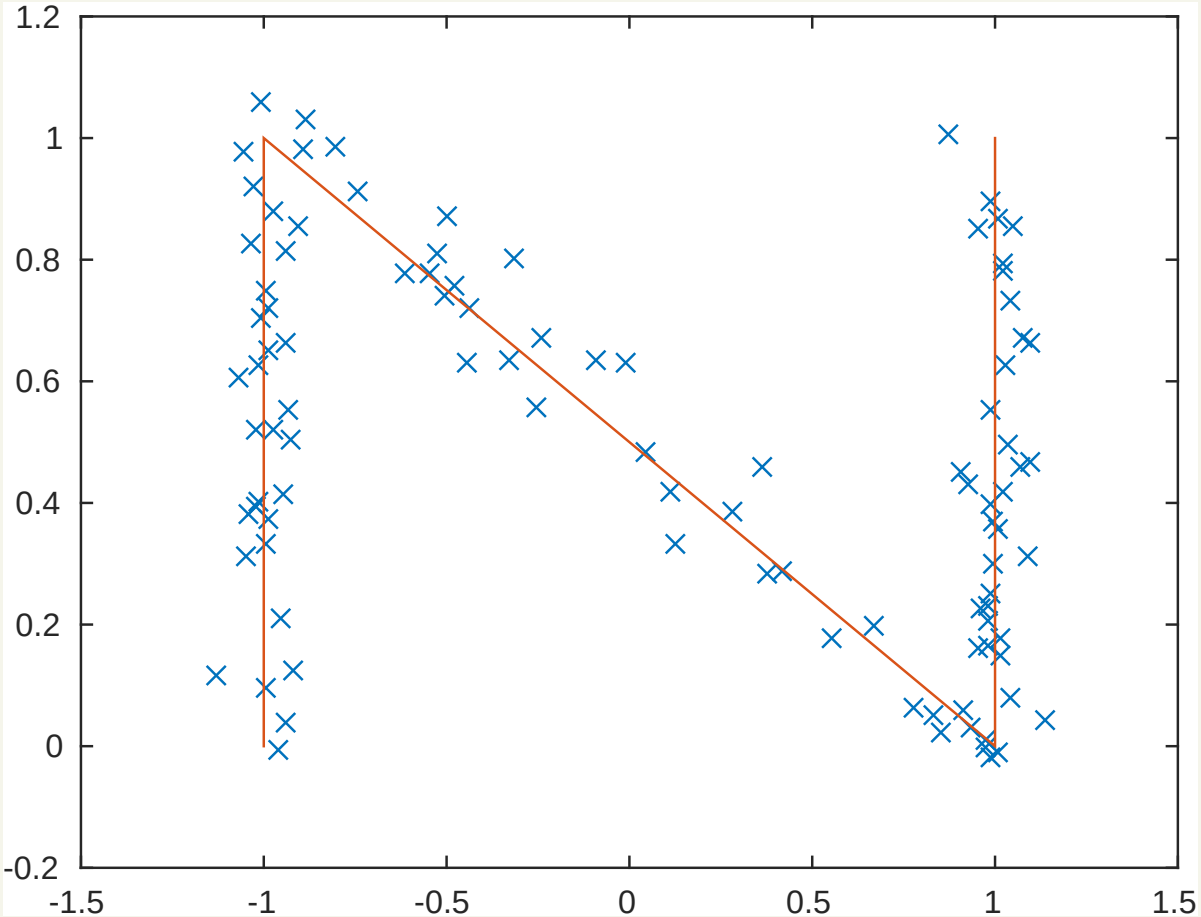

**IDENTIDAD**

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

**Necesito regularización**

# Manifold

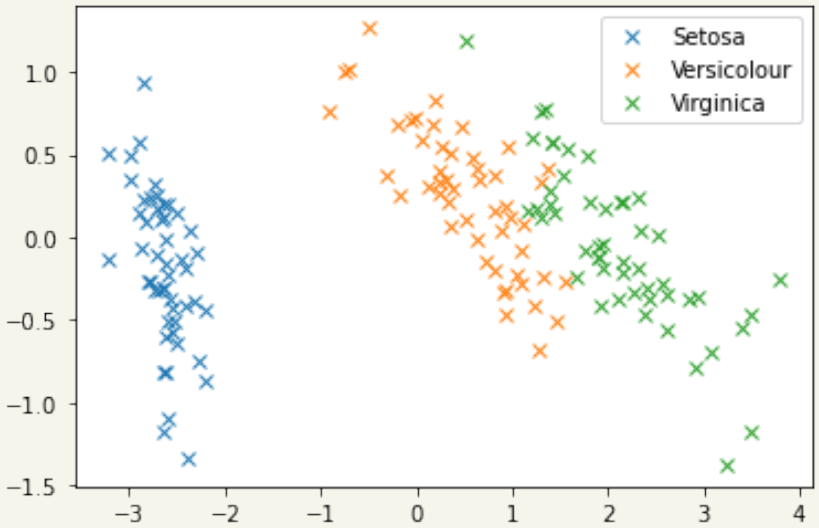
## Regularización de autoencoders




# Algunas Aplicaciones

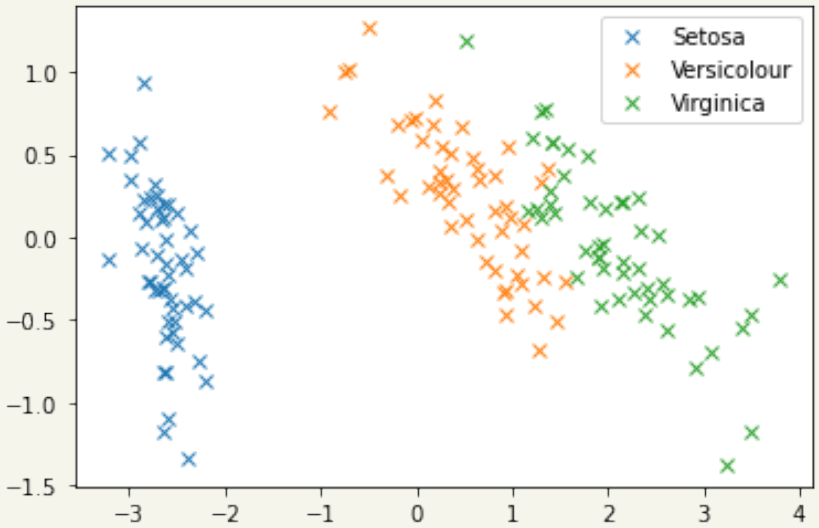
- Para efectuar una inferencia más precisa
- Para pre-procesar los datos
- Para detectar anomalías

# Inferencia

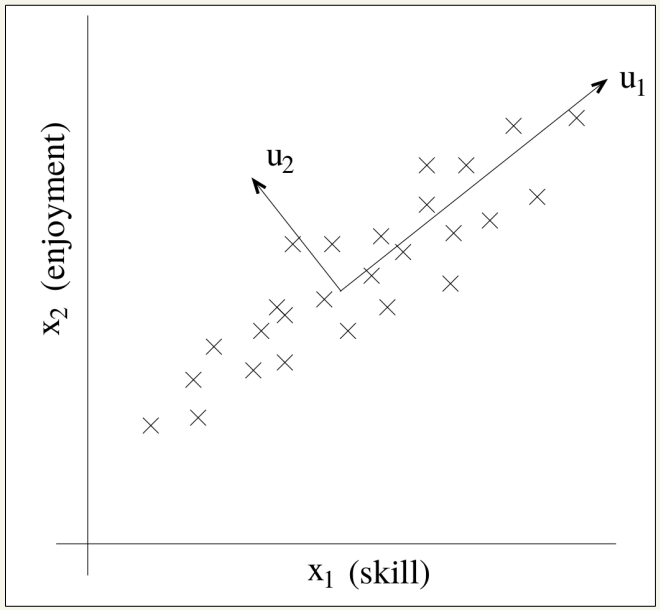



Visualizar en un gráfico 2d o 3d para explicar algunos fenómenos (iris dataset)

# Inferencia




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

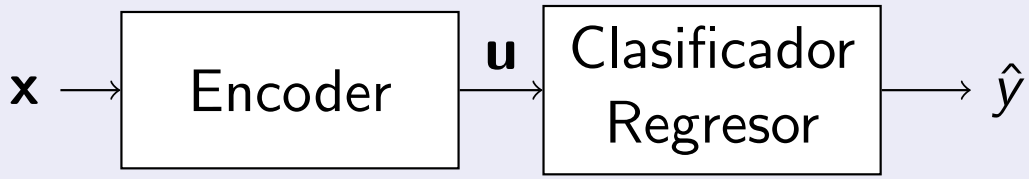



Generar alguna métrica que combine variables muy distintas entre si (radio-controlled helicopters)

# Pre-processing

## Preprocesing: Opción 1

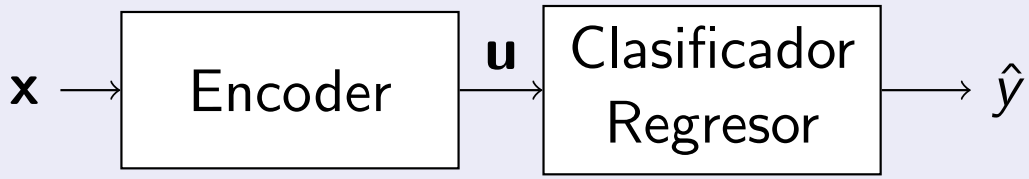
Entrenar el autoencoder y luego usar las muestras en el espacio latente para entrenar el clasificador/regresor.



# Pre-processing

## Preprocesing: Opción 1

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



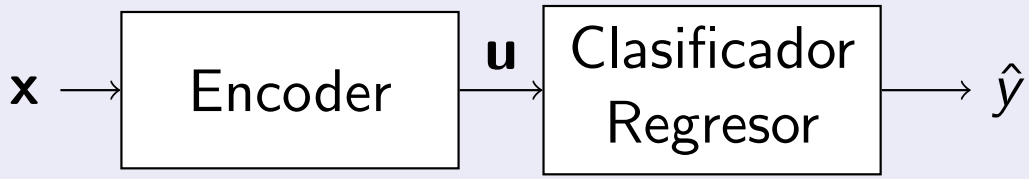

## Preprocesing: Opción 2

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

# Pre-processing

## Preprocesing: Opción 1

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




## Preprocesing: Opción 2

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


## Semi-supervise learning

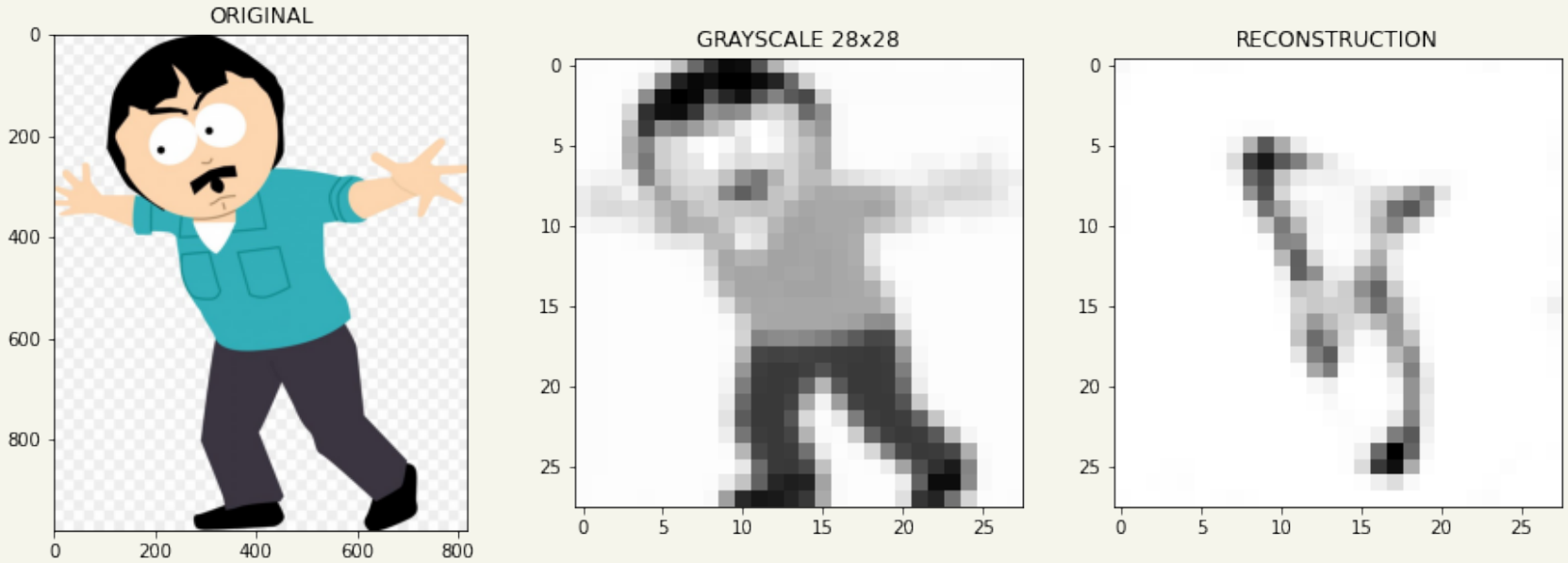
Puedo usar las muestras no supervisadas para entrenar el autoencoder y las supervisadas para el clasificador o el regresor final.

# Detección de anomalías

## Paradigma

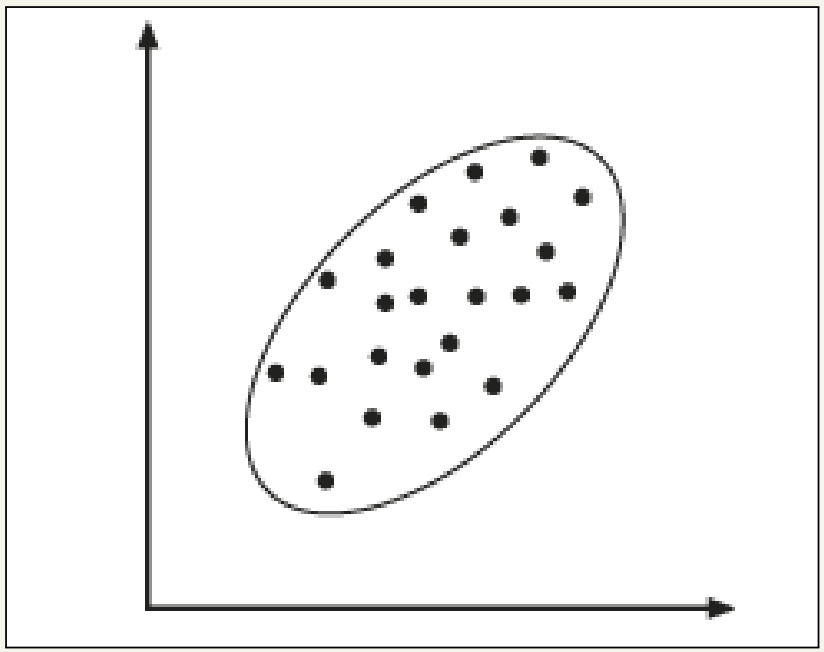
Durante el entrenamiento un autoencoder aprende patrones en los datos para reconstruirlos con cierta facilidad. Entonces es de esperar que una muestra que no cumpla los patrones aprendidos sea más dificil de reconstruir.

**EJEMPLO AUTOENCODER ENTRENADO CON MNIST:**



# Principal Components Analysis

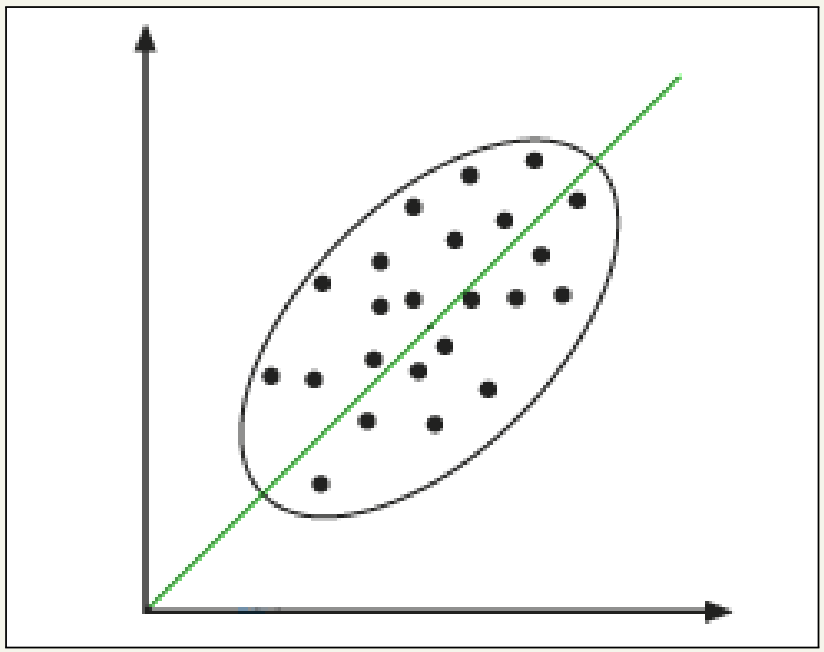
## Reducción lineal



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

# Principal Components Analysis

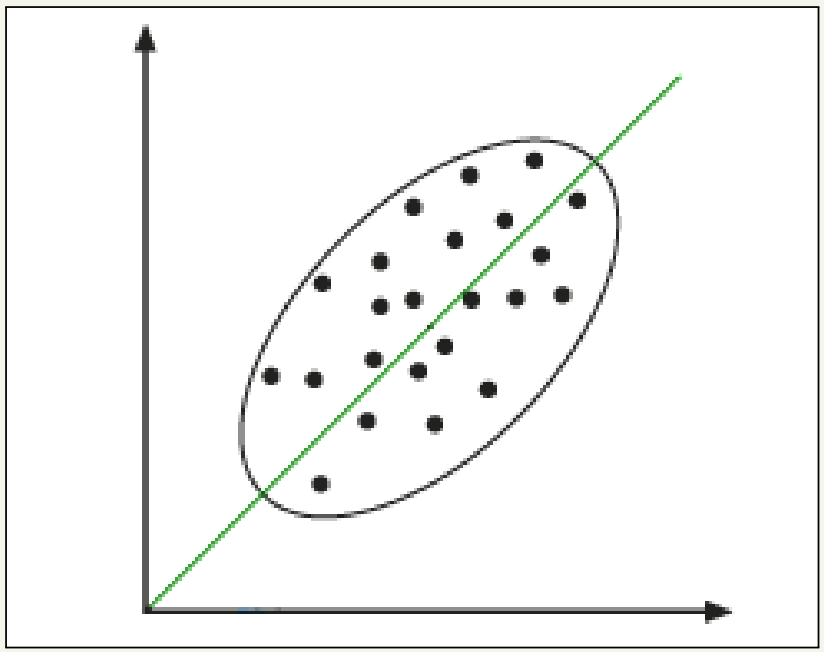
## Reducción lineal



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

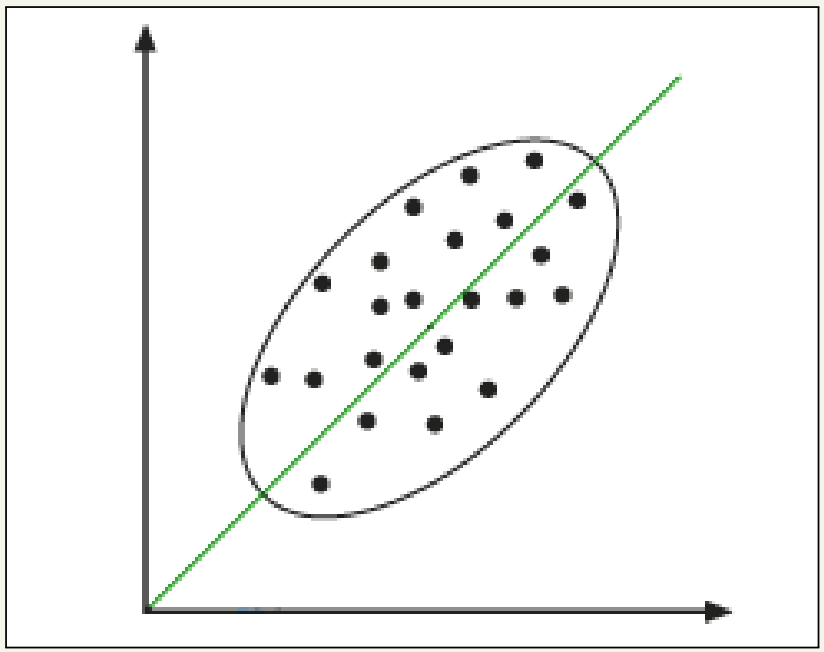
$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_j^{(i)}$$

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

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 autoversor $\mathbf{v}_1$**

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

---

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

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:

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

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

# Principal Components Analysis

## Algunas cuentas

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

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

# Principal Components Analysis

## Algunas cuentas

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

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

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

# Principal Components Analysis

## Algunas cuentas

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

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

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

---

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

# Principal Components Analysis

## Algunas cuentas

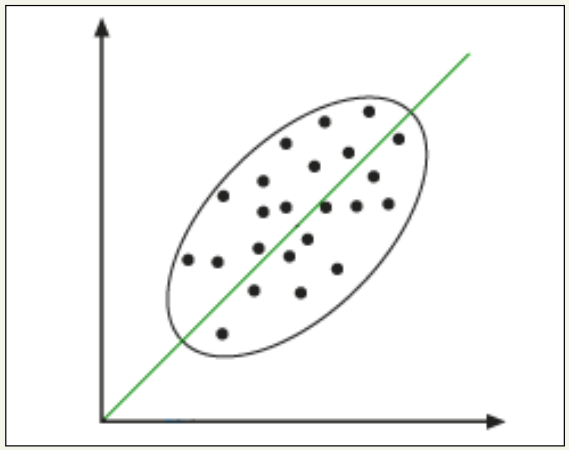
$$J(\mathbf{v}_1) = \mathbf{v}_1^T \boldsymbol{\Sigma} \mathbf{v}_1 - \lambda \left( \mathbf{v}_1^T \mathbf{v}_1 - 1 \right)$$

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

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

El problema de optimización pasa a ser de la forma

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



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.

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

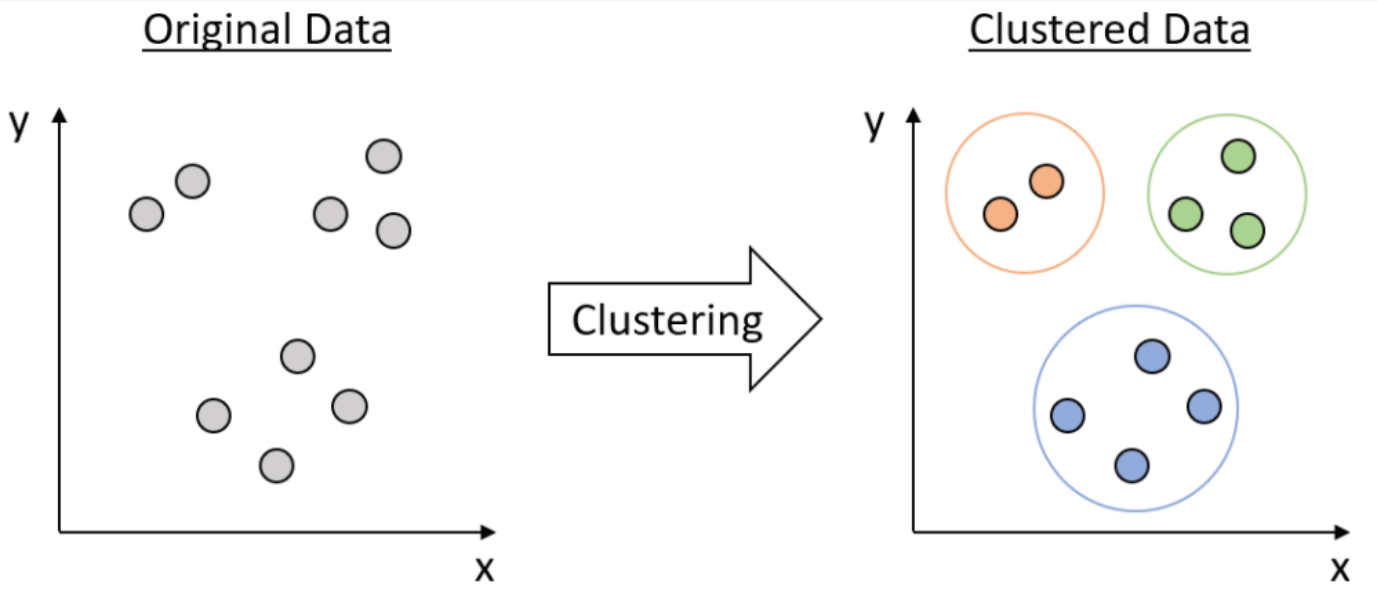
$$\mathbf{u} = \mathbf{V} \cdot \mathbf{x}, \qquad \hat{\mathbf{x}} = \mathbf{V}^T \cdot \mathbf{u}$$

# Outline

# Clustering

## Clustering

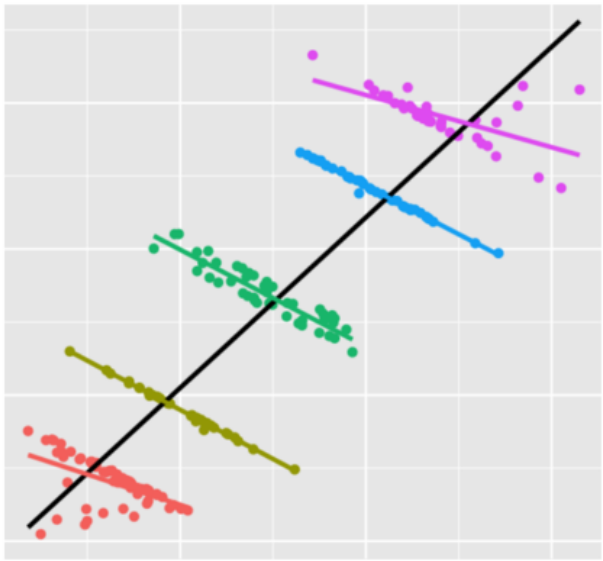
Estos algoritmos son la versión no supervisada de la clasificación. Su objetivo es agrupar muestras de manera de tener un mayor entendimiento del *manifold*.



# 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 | $\geq 80$ | 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

**procedure** $\text{KMeans}(X, K)$
**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 $y \in \{1, \cdots, K\}^n$ etiquetas.
    Inicializar $\mu$ con el valor de $K$ columnas de $X$ elegida al azar.
    **repeat**
        $y[i] = \arg\min_k \|X[i,:] - \mu[k,:]\|$ ▷ Con $i = 1, \cdots, n$.
        $\mu[k,:] = \mathbb{E}\left[X[y == k,:]\right]$ ▷ Con $k = 1, \cdots, K$
    **until** convergencia
    **Return:** $\mu$ e $y$
**end procedure**

# Algoritmo K-Means