

Introducción a la Inteligencia Artificial

Clase 6



Índice

1. Motivación
 - a. Aprendizaje No supervisado
 - b. Aplicaciones
2. kMeans
3. Teoría - Principal Component Analysis
 - a. Concepto
 - b. Demostración Matemática

Algoritmos no supervisados

Aprendizaje no supervisado

en ml sup.
puedo definir
un concepto de
error

en ml no sup.
no tenemos un concepto
claro de error, si no
que vamos a usar
medidas de
relajación

Machine Learning Supervisado	Machine Learning no Supervisado
Proceso aleatorio \bar{X}, y	Proceso aleatorio \bar{X}
$\hat{f}_{y/\bar{x}}(y \bar{x})?$ \longrightarrow Bayes y M.V.	$\hat{f}_{\bar{x}}(\bar{x})?$ \longrightarrow Bayes y M.V.
Inferencias, predicciones	Clusterización, Reducción Dimensionalidad

$\bar{X} \rightarrow$ datos (regresores, features)

$y \rightarrow$ labels / val. a inferir

$y \in \mathbb{R} \vee y \in \mathbb{C} \quad \pi_i \quad \mathcal{L}(y, \hat{y})$

métricas de relajación / equilibrio
métricas de desigualdad

en k means $U = \frac{\text{varianza intra grupo}}{\text{varianza entre grupos}}$

Aplicaciones Generales

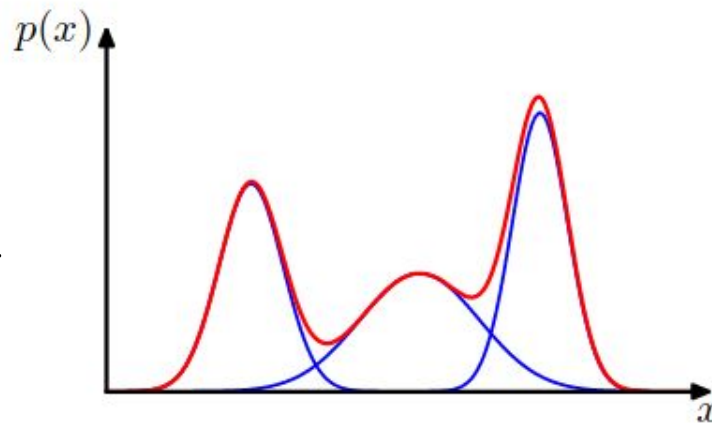
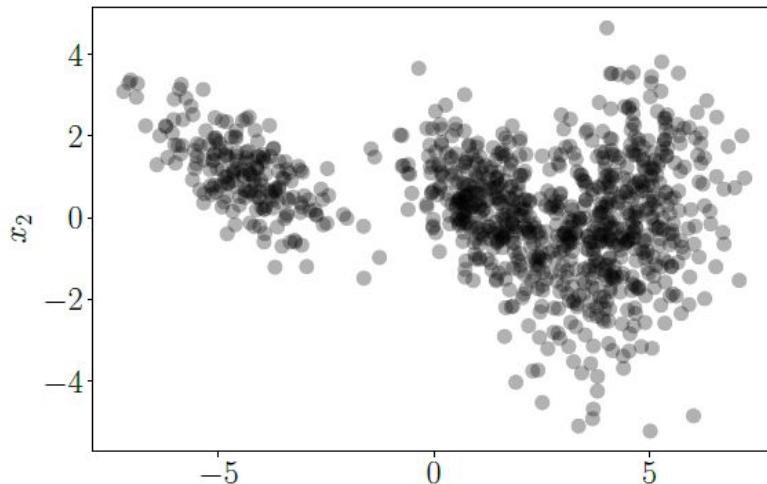
- Data Mining
- Pattern Recognition
- Statistical Analysis

Aplicaciones Específicas

- Density Estimation
- Clustering
- Anomaly Detection
- Object Tracking
- Speech Feature Extraction

• Reducción de dimensionalidad

Anomaly \neq outlier

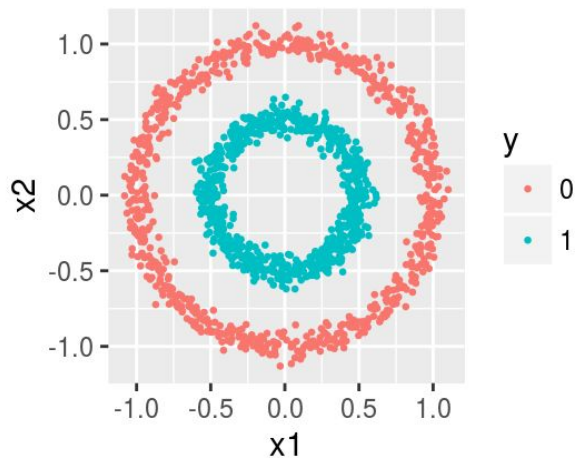


Clustering

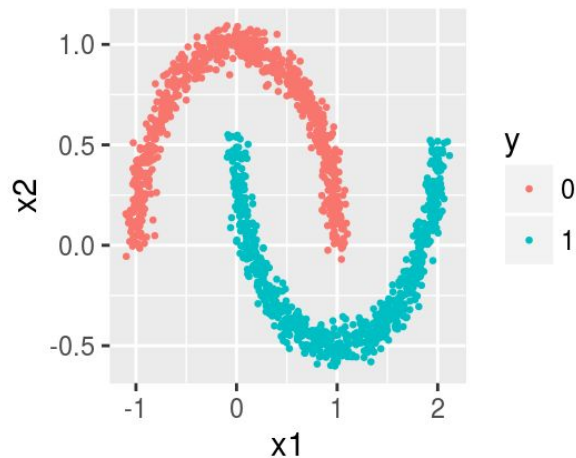
La clusterización o clustering, es el proceso de agrupar objetos en grupos de manera que sean más similares entre sí que con los objetos de otros clusters.

Para generar estos grupos existen diferentes técnicas y diferentes medidas de similaridad.

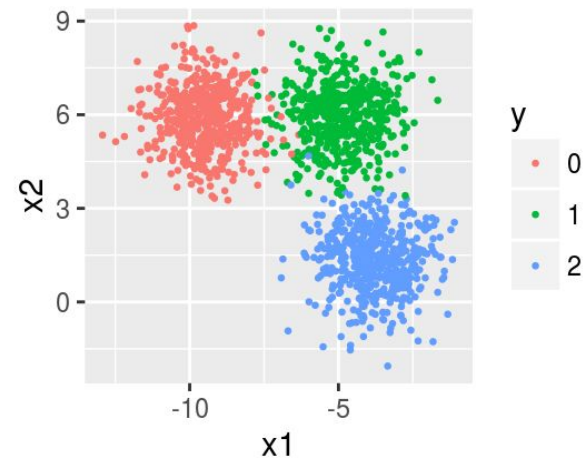
Circles



Moons



Blobs

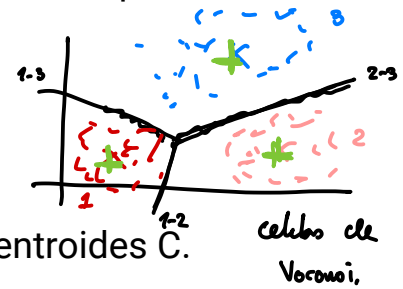


kMeans \rightarrow parameters (n_clusters, c_init, solver, metric)

K-means es uno de los algoritmos más básicos en Machine Learning no supervisado. Es un algoritmo de **clusterización**, que agrupa los datos que comparten características similares. Recordemos que entendemos datos como n realizaciones del vector aleatorio X .

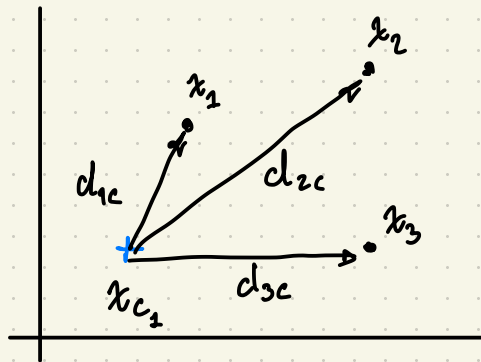
El algoritmo K-means funciona de la siguiente manera:

1. El usuario selecciona la cantidad de clusters a crear (n).
2. Se seleccionan n elementos aleatorios de X como posiciones iniciales de los centroides C .
3. Se calcula la distancia entre todos los puntos en X y todos los puntos en C .
4. Para cada punto en X se selecciona el centroide más cercano de C .
5. Se recalculan los centroides C a partir de usar las filas de X que pertenecen a cada centroide.
6. Se itera entre 3 y 5 una cantidad fija de veces o hasta que la posición de los centroides no cambie.



Implementar la función `def k_means(X, n)` de manera tal que al finalizar devuelva la posición de los centroides y a qué cluster pertenece cada fila de X .

Hint: para (2) utilizar funciones de `np.random`, para (3) y (4) usar los ejercicios anteriores, para (5) es válido utilizar un `for`. Iterar 10 veces entre (3) y (5).



$d(x_i, x_c)$ \rightsquigarrow Euclidean ✓
 \rightsquigarrow Manhattan
 \rightsquigarrow city block
 \rightsquigarrow Mahalanobis

K means es valido para

$$\bar{X} \in \mathbb{R}^{n \times p}$$

②, $p \geq 1$ ②

\hookrightarrow preferible
 mayor estricto

$$d(x_i, x_c) = \sum_{j=1}^p (x_{ij} - x_{cj})^2$$

$$= \|x_i - x_c\|^2$$

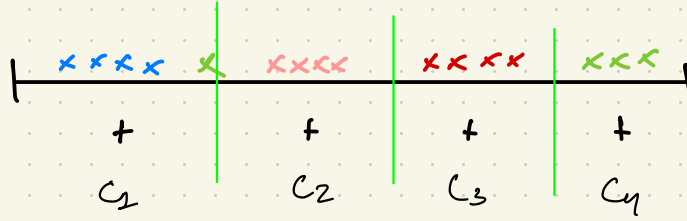
$$W(c) = \frac{1}{2} \sum_{k=1}^K \sum_{c(i)=k} \sum_{c(j)=k} \|x_i - x_j\|^2 \rightsquigarrow \min W?$$

pag 329 Elements of Statistical Learning

① Si mi x tiene categorias puedo usar K-proto.

② porque $p \geq 1$:

si $p = 1$



tenemos un problema de conv. con los centroides
posibles soluciones:

- + mejorar c_{init}
- + cambiar d_{ij} para ponderar dist. pequeñas
- + introducir variaciones
- + modifico el solver

Alternativas: discretización \rightarrow por ancho fijo
 \hookrightarrow u freq. fija.

buscamos: $\arg \min_S \sum_{i=1}^K \sum_{x \in S_i} \|x - \mu_i\|^2 = \arg \min_S \sum_{i=1}^K |S_i| \text{Var}_{S_i}$



media del
cluster o
el centroide

donde $\|\cdot\|^2$ es la norma, $|\cdot|$ es el conteo de muestras.

$$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x$$

$$\arg \min_S \sum_{i=1}^K \frac{1}{|S_i|} \sum_{x, y \in S_i} \|x - y\|^2 \quad \text{BCSS}$$

between cluster sum of squares

Algoritmo estándar:

- etapa de asignación: Aca labelamos basado en su centroide más cercano. (en un paso inicial lo hacemos basado en c_{init})

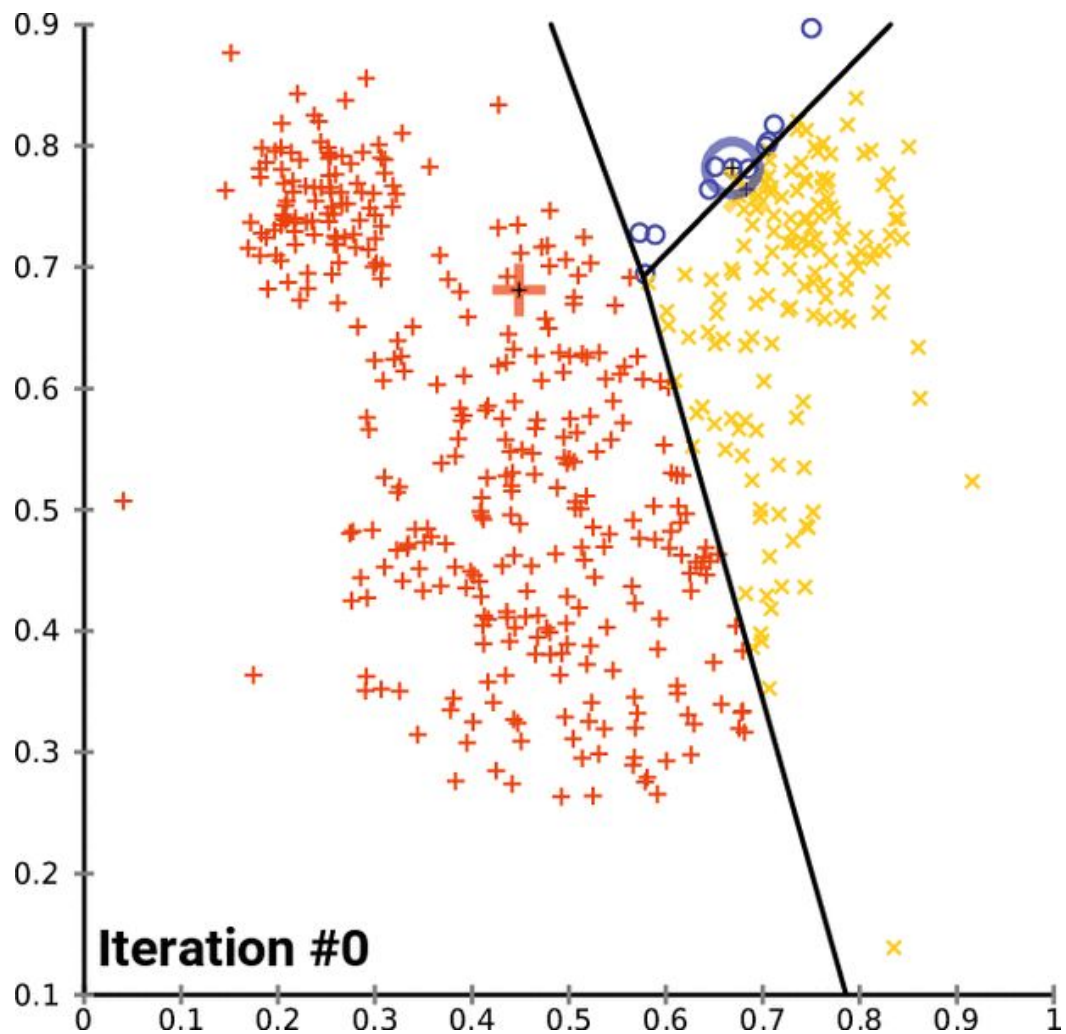
$$S_i^{(t)} = \left\{ x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \quad \forall j, 1 \leq j \leq k \right\}$$

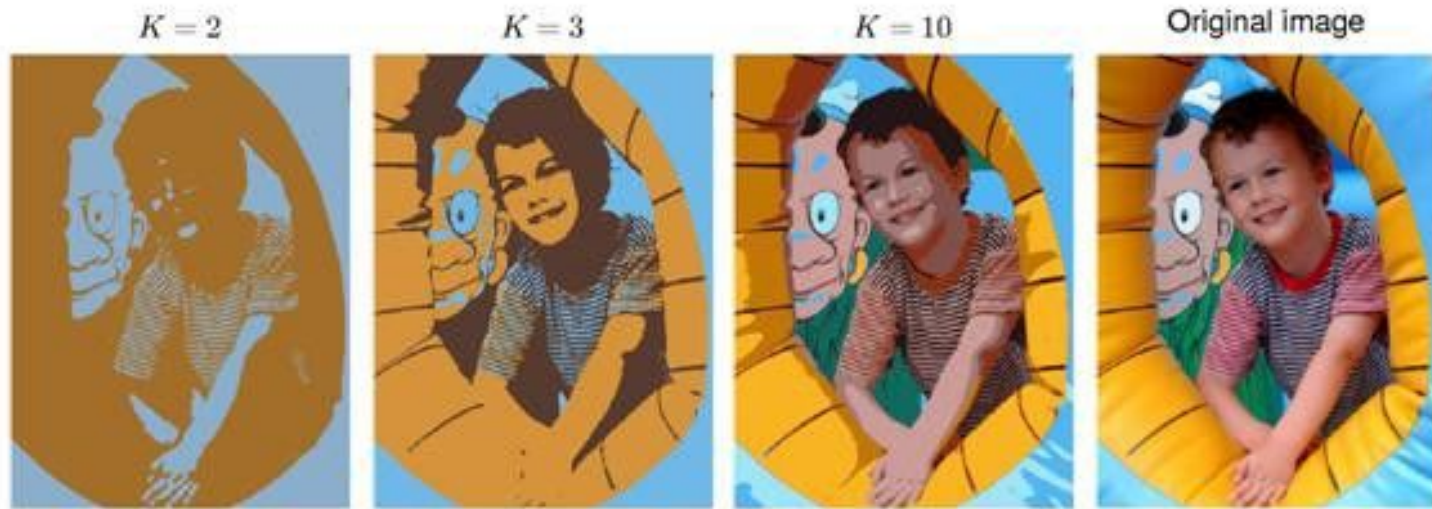
- proceso de update: recalculamos la media de cada cluster:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \cdot \sum_{x_j \in S_i^{(t)}} x_j$$

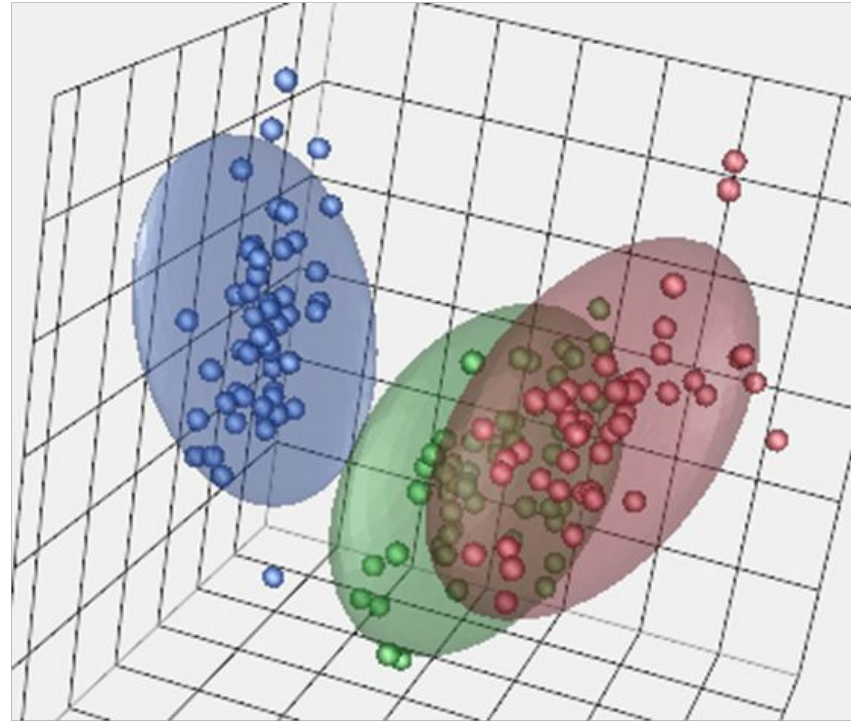
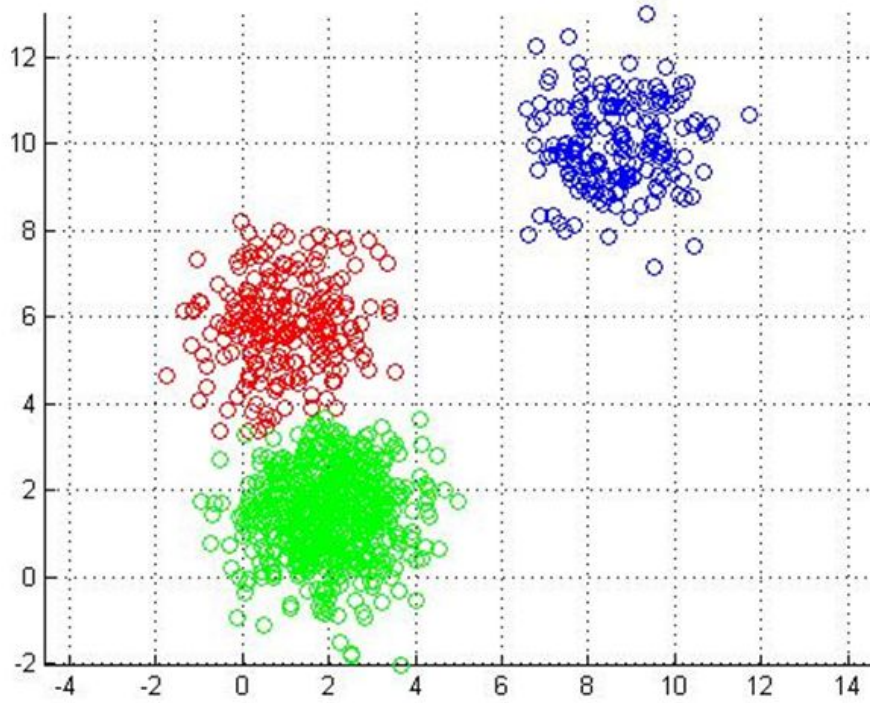
$$\Delta m = m_i^{(t+1)} - m_i^{(t)} \sim 0$$

• Si necesitamos extender la capacidad de clusterización con Var. no numéricas \rightarrow K-proto, K-median, K-medoids





kMeans - Image segmentation



kMeans en R3

Reducción de dimensionalidad

$$\mathbb{R}^{n \times p} \mapsto \mathbb{R}^{m \times p} \quad m < n$$

El objetivo de los modelos de reducción de dimensionalidad es encontrar una “mejor” representación de los datos.

Con “mejor” nos referimos a una representación que preserve la mayor cantidad de información posible de los datos, bajo una determinada penalidad o restricción, que haga que la representación sea más accesible o simple.

Ejemplos de representaciones más simples:

- Representación de menor dimensionalidad
- Representación sparsa
- Representación independiente

Ingeniería de Features - PCA

En ocasiones los datos de entrada tienen muchas features y se torna costoso en tiempo y recursos entrenar modelos de ML con todo el dataset. En la práctica se pueden utilizar técnicas de reducción de la dimensión no supervisadas como PCA (Principal Component Analysis).

Casos de Uso

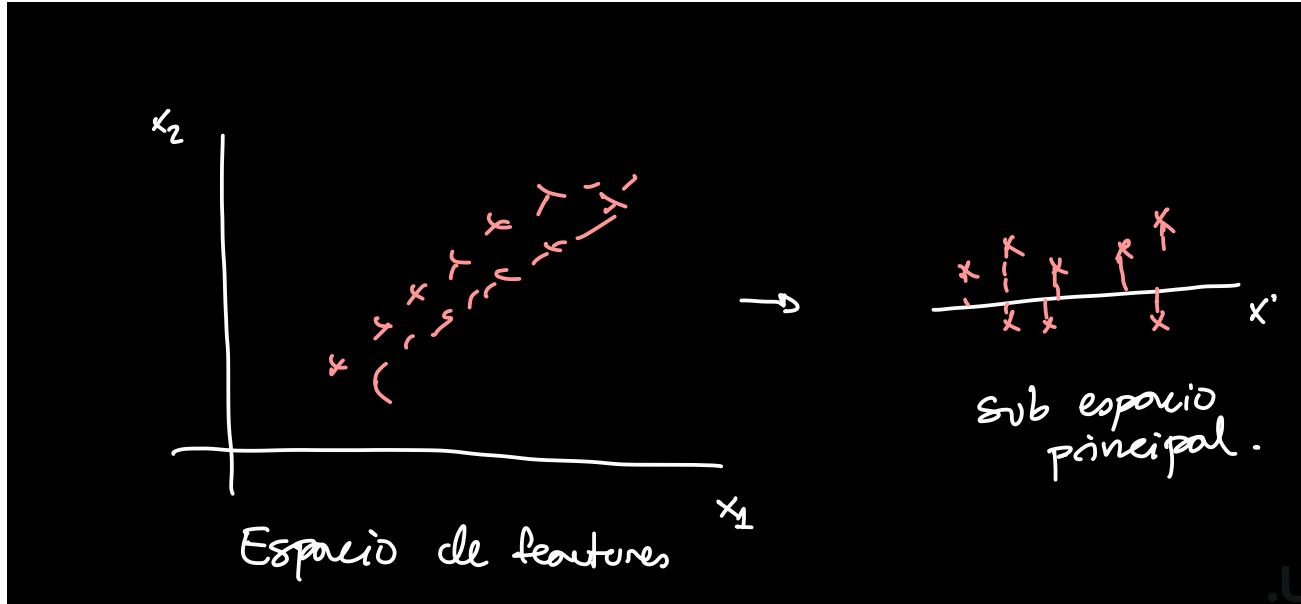
- Compresión de datos
- Identificación de patrones
- Factores latentes
- Visualización

Conocimientos Previos

- Bases y cambio de bases
- Proyecciones
- Valores y vectores propios
- Distribución gaussiana
- Optimización con restricciones

PCA

Queremos encontrar proyecciones ... de observaciones de datos ..., que sean lo más similares posibles a los originales, pero con significativamente menos dimensiones.



PCA

Dado un dataset i.i.d:

$$\chi = \{x_1, \dots, x_N\}, x_N \in \mathbb{R}^D$$

con **media cero**, la matriz de covarianza es: ①

$$S = \frac{1}{N} \sum_{n=1}^N x_n x_n^T$$

Definimos transformaciones lineales:

$$z_n = B^T x_n \in \mathbb{R}^M \quad M \leq D$$

$$B = [b_1, \dots, b_m] \in \mathbb{R}^{D \times M}, b_i^T b_j = 0 \quad \forall i \neq j \quad (\|b_i\| = 1)$$

me dice que
solo soporte
var. numéricas

x_{11}	\dots	\dots	x_{1n}
\dots	\dots	\dots	\dots
x_{d1}	\dots	\dots	x_{dn}

χ

① me dice que χ tiene que estar escalado.

feat 1	feat 2	...	feat
.	.		.
.	.		.
.	.		.
.	.		.
.	.		.

CP1	CP2	CP3	...	CPN
-----	-----	-----	-----	-----

$$D \in \mathbb{R}^{n \times m}$$

$\xrightarrow{\text{PCA}}$

$$D_{\text{PCA}} \in \mathbb{R}^{n \times m}$$

$$\Sigma_{\text{PCA}} = \text{diag}(\vec{b})$$

en genl se atleja
en un mejor modelo (lineal)

PCA

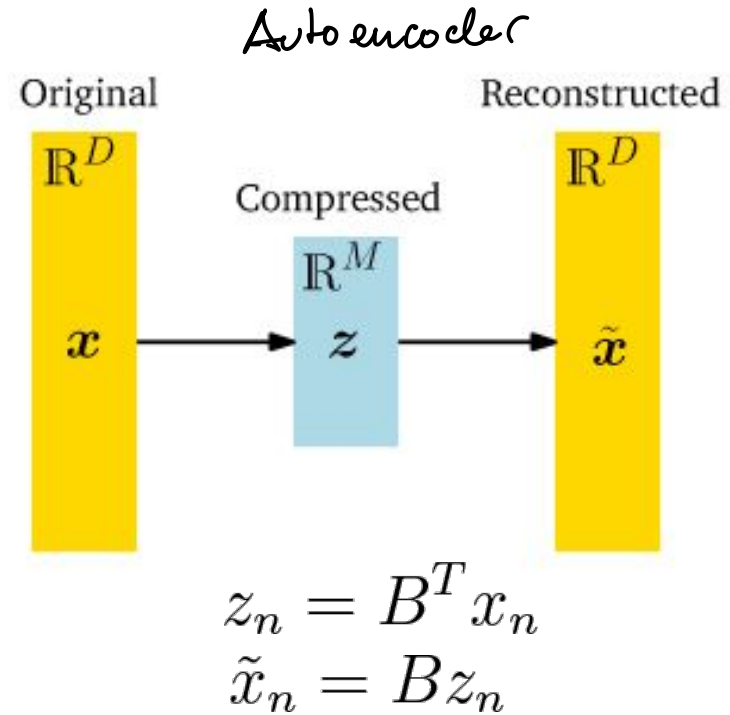
Buscamos un subespacio

$$U \subseteq \mathbb{R}^D / \dim(U) = M < D$$

donde proyectar los datos. Es decir encontrar para:

$$\tilde{x}_n \in \mathbb{R}^D \begin{cases} \rightarrow z_n \\ \rightarrow [b_1, \dots, b_m] \end{cases}$$

- i. Enfoque de máxima varianza
- ii. Enfoque de error de reconstrucción mínimo
- iii. Enfoque de variables latentes



Jamboard - Desarrollo Matemático PCA

- [Introducción](#)
- [Enfoque de maximización de varianza](#)
- [Enfoque de minimización de error de reconstrucción](#)
- [Enfoque por variables latentes](#)

Desarrollo matemático de PCA:

buscamos una proyección \tilde{x}_n de mis datos originales x_n / $\dim(x_n) \leq \dim(x_n)$

• \mathcal{X} es el dataset i.i.d $\wedge x \in \mathbb{R}^D \wedge \vec{\mu}_{\mathcal{X}} = \vec{0}$

• matriz de cov es: $S = \frac{1}{N} \sum x_n x_n^t$

Método de máxima Varianza: buscamos maximizar la varianza en una dimensión inferior

Partimos con una columna de B ($\mathbb{R}^{M \times D}$), $b_1 \in \mathbb{R}^D$

Lo maximizamos la varianza de z_1 de $\tilde{z} \in \mathbb{R}^M$:

$$\text{Var}[z] = \text{Var}[B^t (x - \mu)] = \text{Var}[B^t x - B^t \mu] = \text{Var}[B^t x]$$

$$\text{Var}_1 = \text{Var}[\tilde{z}_{1n}] = \frac{1}{N} \sum_{n=1}^N \tilde{z}_{1n}^2 ;$$

$$\tilde{z}_{1n} = b_1^t x_{1n}$$

$$\text{Var}_1 = \frac{1}{N} \sum_{i=1}^N (b_1^t \cdot x_n)^2 = \frac{1}{N} \sum_{i=1}^N (b_1^t x_n)^t (b_1^t x_n)$$

↳ proyección ortogonal de x_1 en el subesp. unidimensional formado por b_1

$$= \frac{1}{N} \sum_{n=1}^N b_1^t x_n x_n^t b_1 = b_1^t \underbrace{\frac{1}{N} \sum_{n=1}^N x_n x_n^t}_S b_1$$

$$\text{Var}_1 = b_1^t S \cdot b_1$$

si aumento $b_1 \Rightarrow$ incremento Var_1

objetivo: $\max_b b_1^t S b_1$, $\|b_1\|=1$ maximización condicionada

$$L(b_1, \lambda_1) = b_1^t S b_1 + \lambda_1 (1 - b_1^t b_1)$$

$$\begin{cases} \partial_{\lambda_1} L = 0 \rightarrow 1 - b_1^t b_1 = 0 \rightarrow b_1^t b_1 = 1 \\ \partial_{b_1} L = 0 \rightarrow \textcircled{2} \end{cases}$$

$$(2) \rightarrow 2 b_1^t S - 2 \lambda_1 b_1^t = 0$$

$$(b_1^t S)^t = (\lambda_1 b_1^t)^t$$

$$S^t \cdot b_1 = b_1 \lambda_1^t$$

(*) ↓

$$S \cdot b_1 = \lambda_1 b_1$$

def. autovalor

$$\leadsto S \cdot b_1 = \lambda \cdot b_1$$

vector propio de S

↑ Valor propio

(*) puedo hacer esto porque S es simétrica

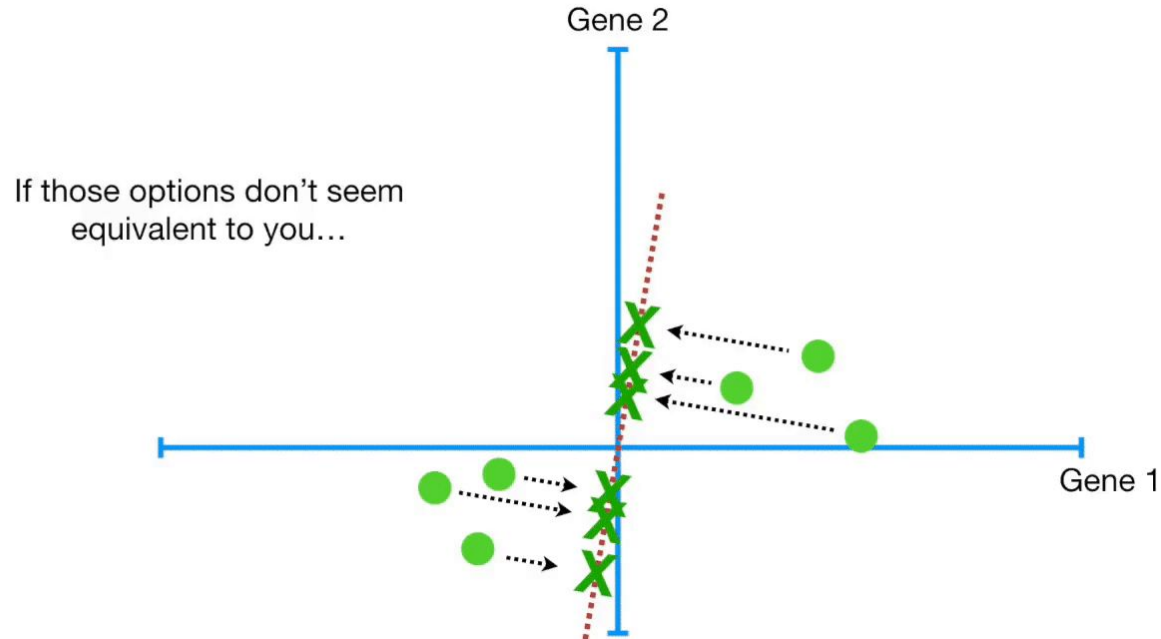
\leadsto seleccionar los autovectores asociados a los m autovalores más grandes de la matriz de covarianza.

con esto: $\begin{cases} - \text{Varianza explicada: } \sum_{i=1}^m \lambda_i \rightarrow \% \text{ de la varianza total.} \\ - \text{ " perdida: } \sum_{j=m+1}^D \lambda_j \end{cases}$

M es un parámetro de nuestro modelo.

PCA

Comparación métodos 1 y 2.



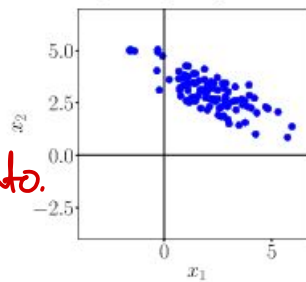
PCA

Pasos principales: 0. *Validar tipo de dato.*

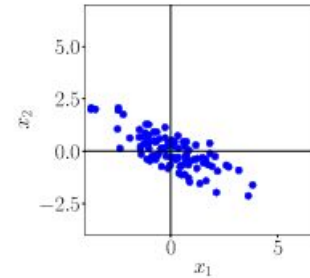
1. Centramos los datos
2. Estandarización
3. Autovalores de la matriz de covarianza
4. Proyección

$$z_n = B^T x_n$$

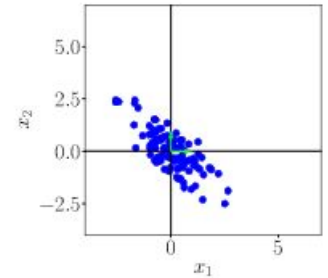
3'. *selección los m autovectores*



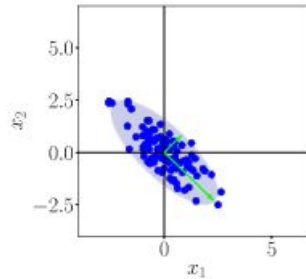
(a) Original dataset.



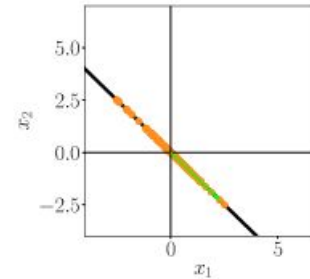
(b) Step 1: Centering by subtracting the mean from each data point.



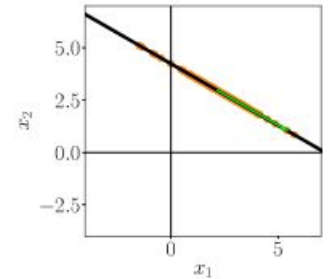
(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.



(d) Step 3: Compute eigenvalues and eigenvectors (arrows) of the data covariance matrix (ellipse).



(e) Step 4: Project data onto the principal subspace.



(f) Undo the standardization and move projected data back into the original data space from (a).

PCA

Derivaciones

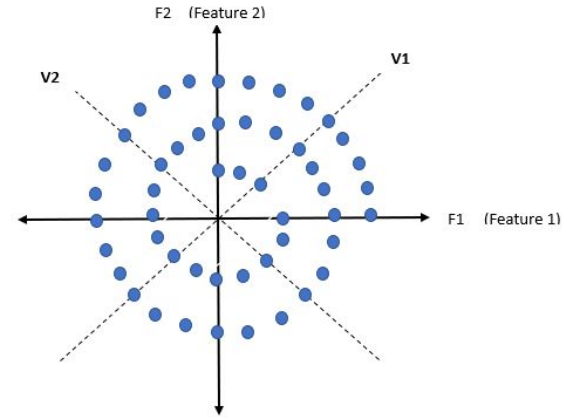
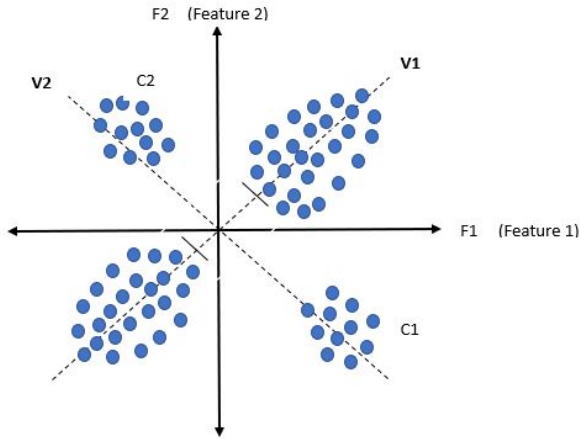
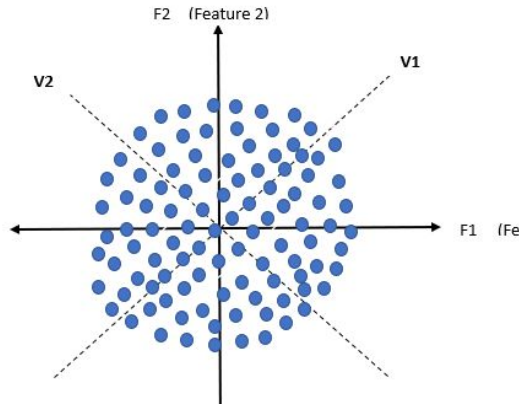
$$Z_n = \underset{\text{e fn. kernel}}{\phi(B^t x_n)} \leftarrow \text{Kernel-PCA}$$

- Si en PCA cambiamos el mapeo lineal por uno no-lineal, obtenemos un auto-encoder. Si el mapeo no-lineal es una red neuronal, tenemos un deep auto-encoder.
- Cuando la varianza del ruido gaussiano es cero, PPCA \rightarrow PCA.
- Si para cada dimensión, el ruido tiene una varianza distinta \rightarrow Factor Analysis.
- Si cambiamos la distribución a priori de z por una no gaussiana \rightarrow ICA

$$X_n = X + \epsilon \rightarrow \epsilon \sim \mathcal{N}(\bar{\phi}, \bar{\Sigma})$$

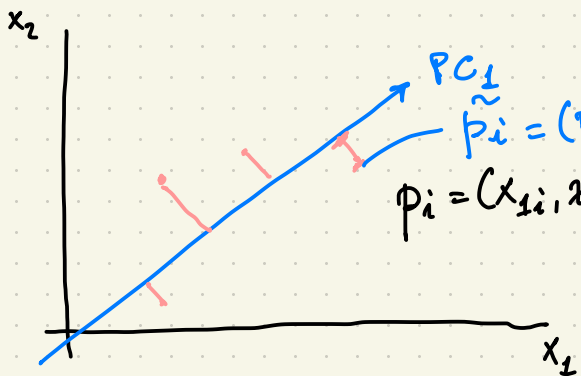
PCA

Limitaciones



Segunda Forma de derivación de PCA:

Minimización del error de reducción:



Si proyectamos un pto. sobre una dirección de la reconstrucción vamos a poder definir la posición del pto \tilde{p}

p_i : pto. original

\tilde{p}_i : pto. proyectado

} Vamos a minimizar $\|p_i - \tilde{p}_i\|$

Vamos a suponer \exists B base **orthonormal** de \mathbb{R}^D con esto decimos que $\forall x \in \mathbb{R}^D$ podemos asegurar:

$$\vec{x} = \sum_{d=1}^D \alpha_d \cdot \bar{b}_d = \underbrace{\sum_{i=1}^M \alpha_i \bar{b}_i}_{\tilde{x}} + \sum_{d=M+1}^D \alpha_d \bar{b}_d$$

queremos encontrar $\tilde{x} = \sum_{i=1}^m \alpha_i \cdot \bar{b}_i \in U \subseteq \mathbb{R}^D$ tal que podamos maximizar la similitud entre \tilde{x} y x :

$$\tilde{x} = \sum_{i=1}^m \tilde{z}_i b_i = \bar{B} \cdot \tilde{z}$$

buscamos minimizar el MSE $\|x - \tilde{x}\|^2$:

①. optimizar \tilde{z}_n para una dada base.

②. encontrar una base óptima.

$$J_M = \frac{1}{N} \sum_{i=1}^N \|x_n - \tilde{x}_n\|^2$$

$$\frac{\partial J_M}{\partial \tilde{z}_{in}} = \frac{\partial J_M}{\partial \tilde{x}_n} \cdot \frac{\partial \tilde{x}_n}{\partial \tilde{z}_{in}}$$

$$= \left(-\frac{2}{N} \cdot (x_n - \tilde{x}_n)^t \right) \cdot \left(\frac{\partial \tilde{x}_n}{\partial \tilde{z}_{in}} \left(\sum_{i=1}^m \tilde{z}_{in} \cdot b_i \right) \right) = -\frac{2}{N} (x_n - \tilde{x}_n)^t \cdot b_i$$

$$\begin{aligned}\partial_{z_{in}} J_M &= -\frac{2}{N} \left(x_n - \sum_{i=1}^M z_{in} \cdot b_i \right)^t \cdot b_i = -\frac{2}{N} \left(x_n^t b_i - \underbrace{z_{in} \cdot b_i^t \cdot b_i}_{=1} \right) \\ &= -\frac{2}{N} (x_n^t b_i - z_{in})\end{aligned}$$

$$\partial_{z_{in}} J_M = 0 \Rightarrow -\frac{2}{N} (x_n^t b_i - z_{in}) = 0 \rightarrow \boxed{z_{in} = b_i x_n^t}$$

los z_{in} son las coord. que vamos a encontrar para cada vector $(b_i x_n^t)$

• busquemos ahora la base óptima:

$$\tilde{x}_n = \sum_{n=1}^M z_{nn} \cdot b_n = \sum_{n=1}^M (x_n^t b_n) b_n = \sum_{n=1}^M (b_n b_n^t) \cdot x_n$$

$$X = \left(\sum_{n=1}^M b_n b_n^t \right) \cdot X_n + \left(\sum_{j=M+1}^D b_j b_j^t \right) \cdot x_j$$

$$X = (x_1, x_2, \dots, x_m, \dots, x_n)$$

$$\tilde{X} = (x_1', x_2', \dots, x_m', 0, \dots, 0)$$

$$X - \tilde{X} = (0, 0, \dots, \underbrace{x_{m+1}, \dots, x_n})$$

distancia (error) : $X_m - \tilde{X}_m = \sum_{j=M+1}^D b_j b_j^t \cdot x_j = \sum_{j=M+1}^D (x_m^t \cdot b_j) \cdot b_j$

$$J_m = \frac{1}{N} \sum_n \|X_n - \tilde{X}_n\|^2 = \frac{1}{N} \cdot \sum_{n=1}^N \left\| \sum_{j=M+1}^D (x_n^t \cdot b_j) b_j \right\|^2$$

$$= \sum_{j=M+1}^D b_j^t \left(\frac{1}{N} \sum_{n=1}^N x_n \cdot x_n^t \right) \cdot b_j$$

S

$$= \sum_{j=M+1}^D b_j^t S b_j = \sum_{j=M+1}^D \text{tr}(b_j^t S b_j) = \sum_{j=M+1}^D \text{tr}(S b_j^t \cdot b_j)$$

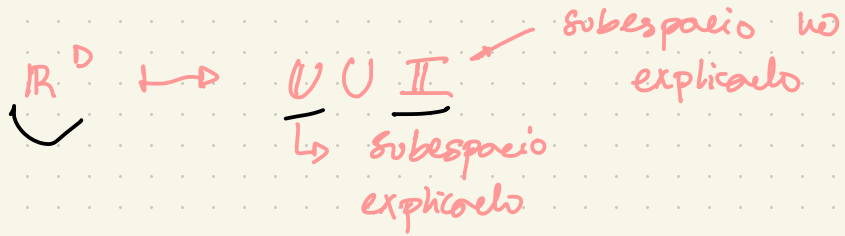
$$= \text{tr} \left(\underbrace{\sum_{j=M+1}^D b_j b_j^t}_{\text{matriz de proyección}} \cdot S \right) \rightarrow \text{el error de reconstrucción se puede pensar como la matriz } S \text{ proyectada sobre el complemento ortogonal de } U.$$

↳ Subev. principal explicada

$$D = U U^T \bar{U}$$

↳ subev. perdidos

A) reducir dimensiones:

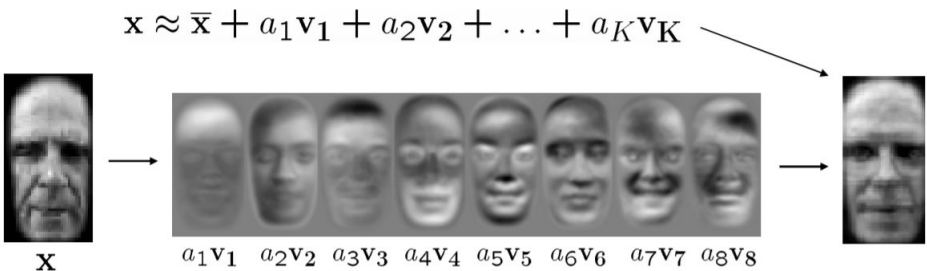
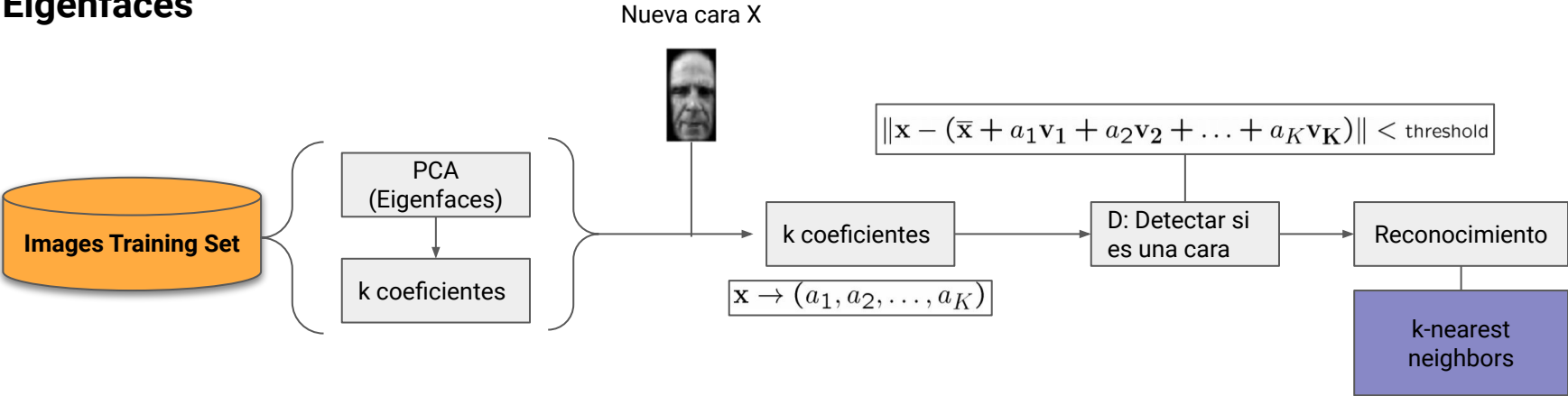


para calcular PCA, nosotros podemos elegir maximizar U o minimizar II .

PCA - Ejemplo

PCA

Eigenfaces



Bibliografía

- The Elements of Statistical Learning | Trevor Hastie | Springer
- An Introduction to Statistical Learning | Gareth James | Springer
- Deep Learning | Ian Goodfellow | <https://www.deeplearningbook.org/>
- Stanford | CS229T/STATS231: Statistical Learning Theory | <http://web.stanford.edu/class/cs229t/>
- Mathematics for Machine Learning | Deisenroth, Faisal, Ong
- Artificial Intelligence, A Modern Approach | Stuart J. Russell, Peter Norvig