



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica



# Tema 3. Reducción de dimensionalidad

Percepción (PER)

Curso 2019/2020

Departamento de Sistemas Informáticos y Computación

# Índice

- 1 Introducción ▷ 3
- 2 Proyecciones lineales ▷ 9
- 3 Vectores propios ▷ 17
- 4 Principal Component Analysis (PCA) ▷ 25
- 5 Aplicación de PCA: EigenFaces ▷ 39

# Índice

- 1 *Introducción* ▷ 3
- 2 Proyecciones lineales ▷ 9
- 3 Vectores propios ▷ 17
- 4 Principal Component Analysis (PCA) ▷ 25
- 5 Aplicación de PCA: EigenFaces ▷ 39

# Introducción

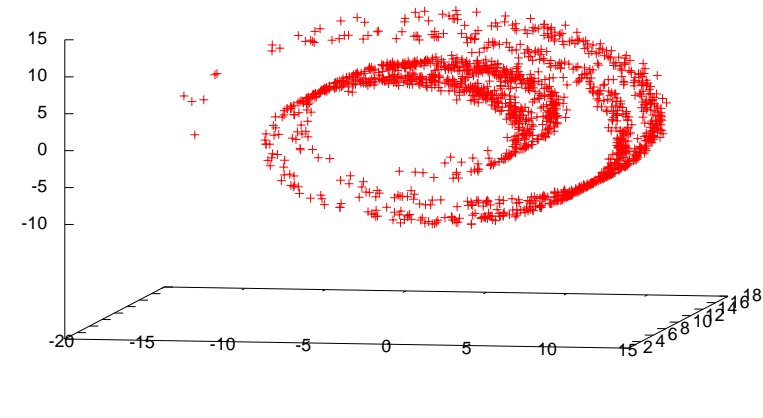
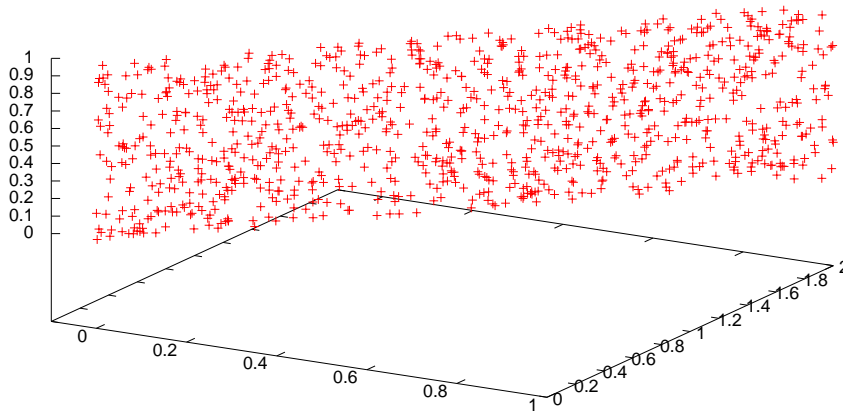
- La representación vectorial de un objeto  $\mathbf{x}$  puede ser de muy alta dimensionalidad  $\mathbf{x} \in \mathbb{R}^D$  siendo  $D > 1000$
- En un espacio de alta dimensionalidad las técnicas de aprendizaje se ven afectadas por lo que se conoce como la *maldición de la dimensionalidad*<sup>1</sup>
- Razones para aplicar técnicas de reducción de dimensionalidad:
  - La dimensionalidad intrínseca puede ser menor que la de la representación
  - Las componentes de cada vector pueden presentar fuertes correlaciones
  - Reducción de parámetros a estimar en el clasificador mejorando su estimación
  - Por eficiencia de tiempo de cómputo y ocupación en memoria
  - Eliminación del ruido durante la adquisición de datos

---

<sup>1</sup>En inglés, *curse of dimensionality*.

# Dimensionalidad intrínseca de los datos

El espacio de representación puede ser  $\mathbb{R}^D$  pero los objetos tienen una representación intrínseca  $\mathbb{R}^{D'}$  con  $D' < D$



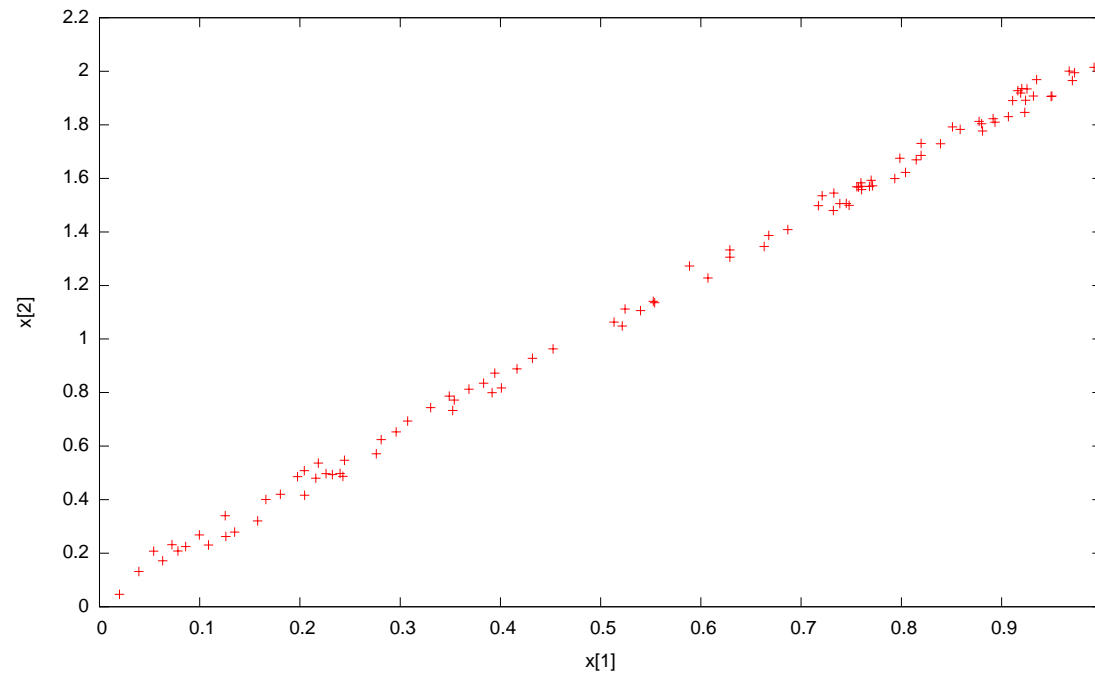
En ambos ejemplos  $D = 3$  y  $D' = 2$

# Correlaciones entre componentes

Algunas dimensiones están correladas y no aportan información extra

Ejemplo  $\mathbf{x} \in \mathbb{R}^2$ :

- $x_2 = 2x_1 \rightarrow$  Dimensionalidad intrínseca  $D' = 1$
- $x_2 = 2x_1 + \epsilon$  con  $\epsilon$  independiente de  $x_1 \rightarrow D' = 2$   
pero  $x_2$  no aporta mucha más información que  $x_1$



# Estimación de los parámetros del clasificador

- Sea un clasificador por funciones discriminantes lineales (FDLs)

- $G \equiv \{g_1, g_2, \dots, g_C\}$ ,  $g_c(\mathbf{x}) = \mathbf{w}_c \cdot \mathbf{x} + b$

- Si  $\mathbf{x} \in \mathbb{R}^D$ , hay  $C \times (D + 1)$  parámetros a estimar

- Ejemplo:

Clasificar mediante FDLs las caras de 100 individuos. De cada individuo se tienen 4 imágenes. Cada imagen consta de  $75 \times 75$  píxeles. Considerando una representación geométrica global tenemos que:

$$100 \times (5625 + 1) = 562600 \text{ parámetros a estimar con sólo 400 muestras}$$

- Problema:

Un espacio de  $75 \times 75$  dimensiones se puede considerar *vacío* con sólo 400 ejemplos. Es más, para que dejara de considerarse *vacío* haría falta un número de muestras que nunca vamos a tener disponibles

# Reducción de dimensionalidad

- Proceso para reducir la dimensionalidad del espacio de representación de los datos
- Siendo  $E = \mathbb{R}^D$  el espacio de representación de los datos, la reducción de dimensionalidad se puede ver como una función:

$$f(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^k, \quad k \ll D$$

- Así, dada una representación  $\mathbf{x} \in \mathbb{R}^D$ , tendremos  $\mathbf{x}' = f(\mathbf{x})$ ,  $\mathbf{x}' \in \mathbb{R}^k$



# Índice

- 1 Introducción ▷ 3
- 2 *Proyecciones lineales* ▷ 9
- 3 Vectores propios ▷ 17
- 4 Principal Component Analysis (PCA) ▷ 25
- 5 Aplicación de PCA: EigenFaces ▷ 39

# Proyecciones lineales

- Sólo consideraremos proyecciones lineales
- Son el resultado de multiplicar  $\mathbf{x}$  por una *matriz de proyección*  $W$

$$\begin{array}{ccccc} \mathbf{x}' & = & W^t & \cdot & \mathbf{x} \\ k \times 1 & & k \times D & & D \times 1 \end{array}$$

donde  $W \in \mathbb{R}^{D \times k}$  y  $\mathbf{x} \in \mathbb{R}^D$ ,  $k < D$

- La proyección resulta en un vector  $\mathbf{x}' \in \mathbb{R}^k$
- Ejemplo para  $D = 3$  y  $k = 2$

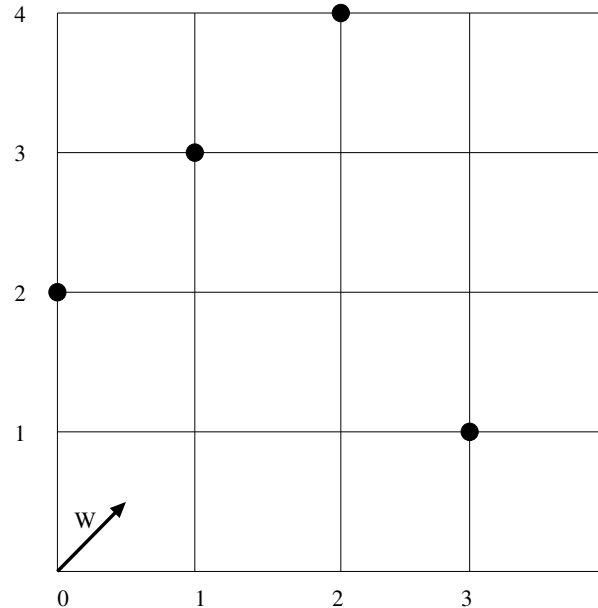
$$\begin{array}{ccccc} \begin{pmatrix} 2 \\ 0 \end{pmatrix} & = & \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} & \cdot & \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} \\ \mathbf{x}' & = & W^t & \cdot & \mathbf{x} \end{array}$$

# Proyecciones lineales

- Sean los siguientes puntos en un espacio  $\mathbb{R}^2$ :

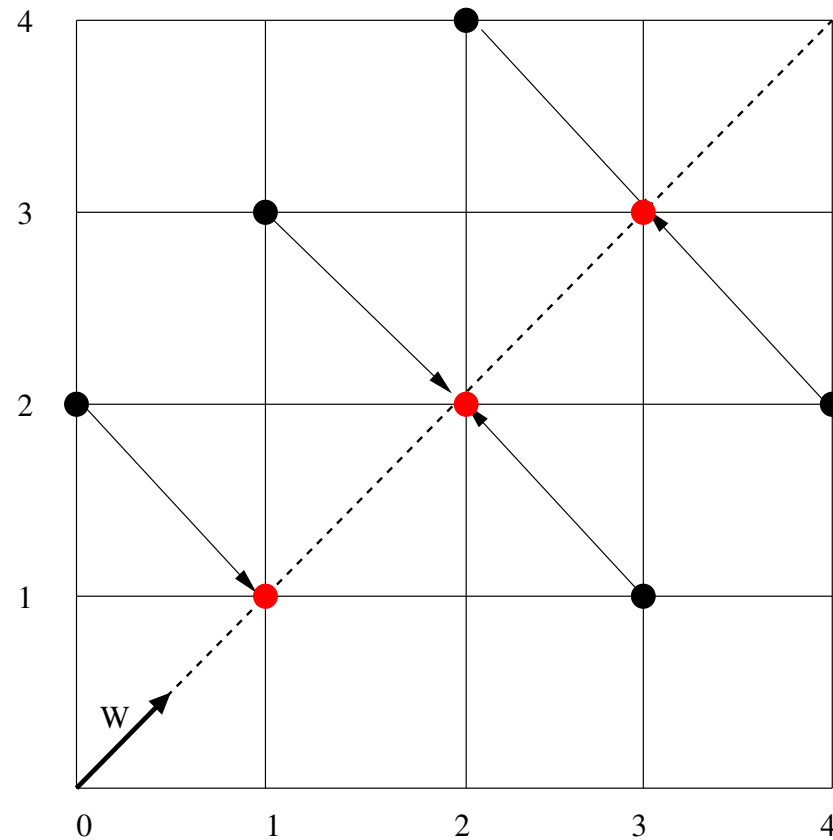
$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \quad \mathbf{x}_2 = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \quad \mathbf{x}_3 = \begin{pmatrix} 2 \\ 4 \end{pmatrix} \quad \mathbf{x}_4 = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \quad \mathbf{x}_5 = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

- Calculemos la proyección con  $W = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$



# Proyecciones lineales

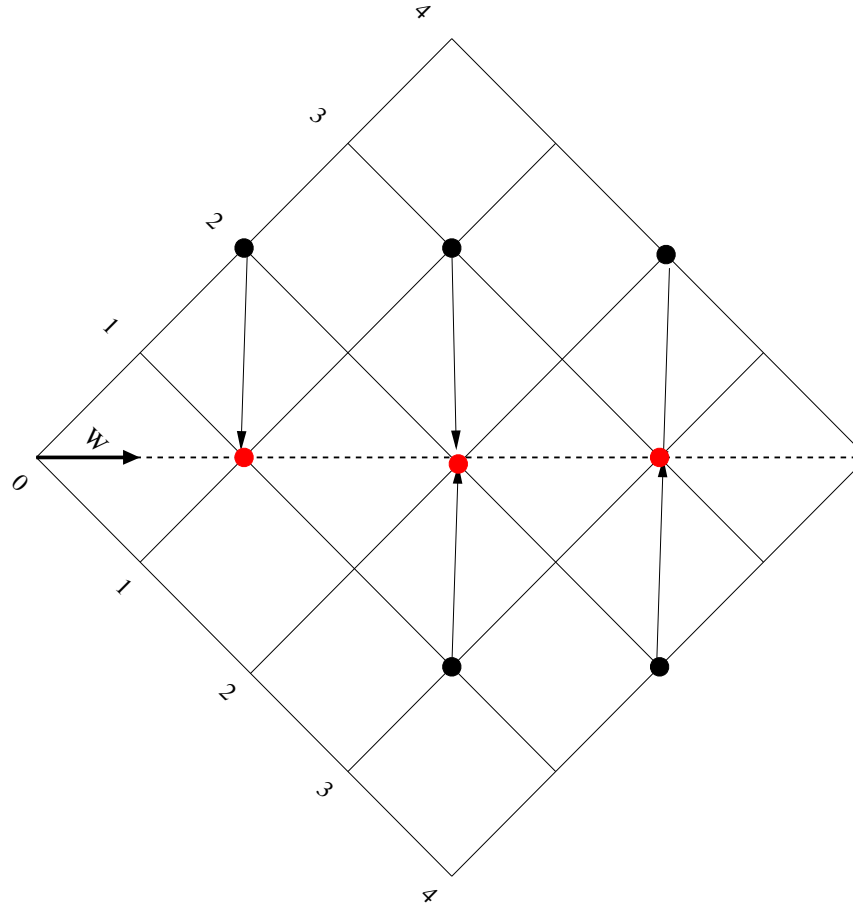
¿Qué está pasando con  $x_2$  y  $x_4$ ?



Proyección ortogonal a  $W$

# Proyecciones lineales

Desde el punto de vista de  $W$ , esta reducción de dimensionalidad es una proyección sobre la recta definida por dicha matriz  $W = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$



# Proyecciones lineales: interpretación geométrica

- Una proyección es un producto escalar de los vectores columna  $\mathbf{w}$  de  $W$  por  $\mathbf{x}$

$$\mathbf{w}^t \cdot \mathbf{x} = |\mathbf{w}| \cdot |\mathbf{x}| \cdot \cos \alpha$$

donde  $\alpha$  es el ángulo definido entre  $\mathbf{w}$  y  $\mathbf{x}$

- Calculemos ahora la proyección con  $W = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$  de:

$$\mathbf{x}_1 = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \quad \mathbf{x}_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \quad \mathbf{x}_3 = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \quad \mathbf{x}_4 = \begin{pmatrix} 0 \\ -2 \end{pmatrix} \quad \mathbf{x}_5 = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$$

- ¿Cómo podríamos interpretar el signo y magnitud del resultado?

# Proyecciones lineales: interpretación geométrica

- En toda proyección lineal la dimensión  $i$ -ésima de  $\mathbf{x}'$  se calcula como:

$$\mathbf{x}'_i = \mathbf{w}_i^t \cdot \mathbf{x}$$

donde  $\mathbf{w}_i$  es la columna  $i$ -ésima de  $W$

- En la práctica,  $\mathbf{x}$  se multiplica por cada uno de los vectores columna de  $W$
- El producto escalar  $\mathbf{w}_i^t \cdot \mathbf{x}$  normalizado es la similitud coseno entre  $\mathbf{w}_i$  y  $\mathbf{x}$ :

$$\cos \alpha = \frac{\mathbf{w}_i^t \cdot \mathbf{x}}{|\mathbf{w}_i| \cdot |\mathbf{x}|}$$

# Proyecciones lineales y reducción de dimensionalidad

- En las reducciones de dimensionalidad lineales se busca una  $W$  adecuada
- La nueva representación en  $E' \equiv \mathbb{R}^k$  debería cumplir las propiedades de:
  - *Continuidad*: cosas cercanas permanecen cercanas
  - *Discriminativa*: datos de distintas clases están separados
  - *Invarianza*: a transformaciones usuales en el espacio original
- Una proyección lineal con estas propiedades es en general difícil de hallar
- Propuesta:
  - Búsqueda de la  $W$  adecuada como un problema de optimización
  - Selección de criterio de optimización según una cierta propiedad deseable
- Este problema de optimización requiere el uso de **vectores y valores propios** (*eigenvectors* y *eigenvalues*, respectivamente)

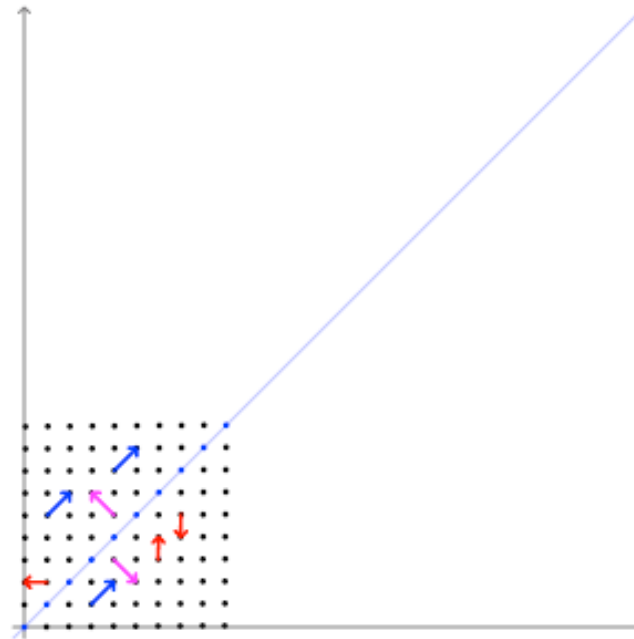


# Índice

- 1 Introducción ▷ 3
- 2 Proyecciones lineales ▷ 9
- 3 *Vectores propios* ▷ 17
- 4 Principal Component Analysis (PCA) ▷ 25
- 5 Aplicación de PCA: EigenFaces ▷ 39

# Vectores propios de una matriz

Ejemplo de transformación lineal ( $\mathbf{x}' = W^t \mathbf{x}$  donde  $\mathbf{x}', \mathbf{x} \in \mathbb{R}^2$ ):



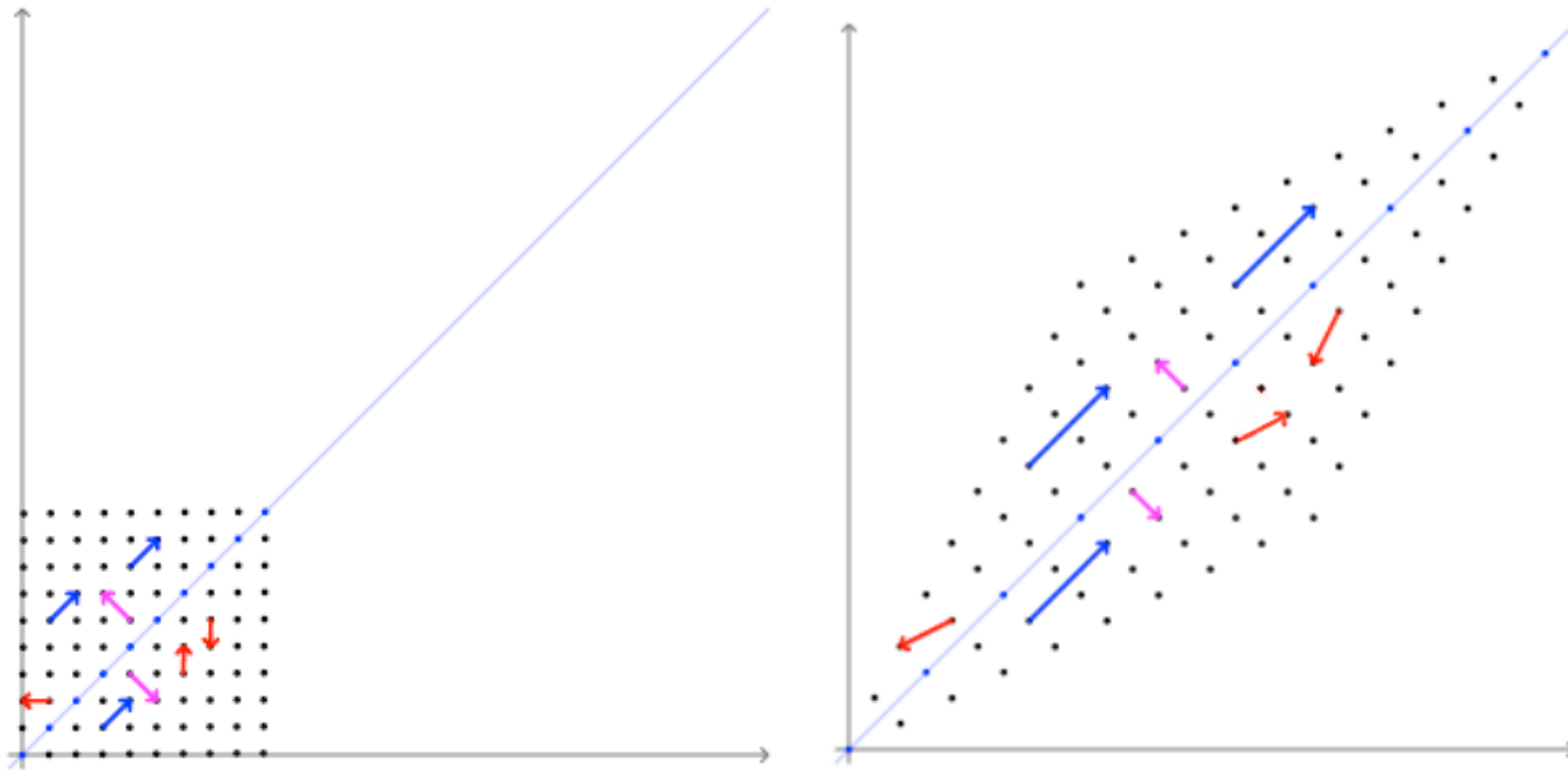
incluyendo vectores

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \mathbf{x}_2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \quad \mathbf{x}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \mathbf{x}_4 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad \mathbf{x}_5 = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad \mathbf{x}_6 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Below the vectors, there are colored arrows pointing to them: a blue arrow under  $\mathbf{x}_1$ , a red arrow under  $\mathbf{x}_2$ , a red arrow under  $\mathbf{x}_3$ , a pink arrow under  $\mathbf{x}_4$ , a red arrow under  $\mathbf{x}_5$ , and a pink arrow under  $\mathbf{x}_6$ .

# Vectores propios de una matriz

Si aplicamos la transformación lineal con  $W = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$  tenemos:



¿Qué vectores no son modificados en dirección, aunque puede que en magnitud?

# Vectores propios de una matriz

- Observamos que los vectores

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

son simplemente escalados pero no rotados

- Dichos vectores son **vectores propios** (*eigenvectors*) de la matriz  $W$
- Se dice que  $\mathbf{x}$  es un vector propio de  $W$  si:

$$W^t \mathbf{x} = \lambda \mathbf{x}$$

con  $\lambda \in \mathbb{R}$

- A este escalar se le conoce como **valor propio** (*eigenvalue*)

# Vectores propios de una matriz. Problema

- Muchas técnicas usan el cálculo de vectores y valores propios de una matriz
- Los valores propios se interpretan según qué represente esa matriz
- Encontrar los vectores propios supone resolver el sistema:

$$W^t \mathbf{x} = \lambda \mathbf{x}$$

o lo que es lo mismo

$$W^t \mathbf{x} - \lambda \mathbf{x} = 0 \quad \longrightarrow \quad (W^t - \lambda I) \mathbf{x} = 0$$

- Por simplicidad en la presentación del cálculo de vectores y valores propios utilizaremos la notación  $W$  sin transponer

$$W \mathbf{x} - \lambda \mathbf{x} = 0 \quad \longrightarrow \quad (W - \lambda I) \mathbf{x} = 0$$

# Vectores propios de una matriz. Solución

- Según un teorema fundamental del álgebra lineal:

$$(W - \lambda I)\mathbf{x} = 0 \quad \Leftrightarrow \quad \det(W - \lambda I) = 0$$

- Pasos de resolución:
  1. Calcular valores propios  $\lambda$  (los que hacen que  $\det(W - \lambda I) = 0$ )
  2. Utilizar  $\lambda$  para resolver el sistema lineal  $(W - \lambda I)\mathbf{x} = 0$
- Por lo tanto cada valor propio tiene asociado un vector propio:

$$(W - \lambda_1 I)\mathbf{x}_1 = 0$$

...

$$(W - \lambda_n I)\mathbf{x}_n = 0$$

- El valor 0 denota un vector  $D$ -dimensional de ceros
- $n = D$  si la matriz  $W \in \mathbb{R}^{D \times D}$  es de rango completo

## Vectores propios de una matriz. Ejemplo

Sea  $W = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$  vamos a calcular sus valores y vectores propios

$$\det(A - \lambda I) = 0 \rightarrow \det \left( \begin{pmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{pmatrix} \right) = 0 \rightarrow (1-\lambda)^2 - 4 = 0 \rightarrow \boxed{\lambda = \{-1, 3\}}$$

Para  $\lambda_1 = -1$ , encontrar  $\mathbf{x}_1$  que cumpla  $(W - \lambda_1 I)\mathbf{x}_1 = 0$

$$\begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \rightarrow \begin{cases} (2x_{11} + 2x_{12}) = 0 \\ (2x_{11} + 2x_{12}) = 0 \end{cases} \rightarrow \mathbf{x}_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Para  $\lambda_2 = 3$ , encontrar  $\mathbf{x}_2$  que cumpla  $(W - \lambda_2 I)\mathbf{x}_2 = 0$

$$\begin{pmatrix} -2 & 2 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} x_{21} \\ x_{22} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \rightarrow \begin{cases} (-2x_{21} + 2x_{22}) = 0 \\ (2x_{21} - 2x_{22}) = 0 \end{cases} \rightarrow \mathbf{x}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

En realidad  $\mathbf{x}_1$  y  $\mathbf{x}_2$  definen una *base*, ya que existen infinitos vectores propios.

# Consideraciones prácticas

En general, no existe solución cerrada para polinomios de orden  $> 4$  (Teorema de Abel-Ruffini)

Existen algoritmos numéricos iterativos para encontrar soluciones:

- QR
- Householder
- Lanczos



# Índice

- 1 Introducción ▷ 3
- 2 Proyecciones lineales ▷ 9
- 3 Vectores propios ▷ 17
- 4 *Principal Component Analysis (PCA)* ▷ 25
- 5 Aplicación de PCA: EigenFaces ▷ 39

# PCA: Principal Component Analysis

- PCA es probablemente la técnica de reducción de dimensionalidad más conocida
- Es una técnica de reducción de dimensionalidad *no supervisada* (la capacidad discriminativa entre clases no se tiene en cuenta)
- Es muy empleada por preservar la mayor parte de la *varianza* de los datos, que se asume que incluye la capacidad de discriminar entre clases
- Se suele emplear como un preproceso previo a otras técnicas de reducción de dimensionalidad supervisadas

# Problema de optimización

- Objetivo PCA: encontrar una matriz de proyección  $W$  que minimice el error de reconstrucción
- Sea un conjunto de datos  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  con  $\mathbf{x}_i \in \mathbb{R}^D$  y  $\bar{\mathbf{x}} = \frac{1}{n} \sum_i \mathbf{x}_i$
- Dado un vector  $\mathbf{x}_i \in \mathbb{R}^D$ , el proceso de reconstrucción es:
  - Se resta  $\bar{\mathbf{x}}$  a  $\mathbf{x}_i$  y se proyecta el resultado al espacio reducido  $\mathbb{R}^k$

$$\mathbf{x}'_i = W^t(\mathbf{x}_i - \bar{\mathbf{x}}) \quad W \in \mathbb{R}^{D \times k} \quad y \quad \mathbf{x}'_i \in \mathbb{R}^k$$

- Se proyecta  $\mathbf{x}'_i$  al espacio original  $\mathbb{R}^D$  y se suma  $\bar{\mathbf{x}}$

$$\hat{\mathbf{x}}_i = \bar{\mathbf{x}} + V^t \mathbf{x}'_i \quad V \in \mathbb{R}^{k \times D} \quad y \quad \hat{\mathbf{x}}_i \in \mathbb{R}^D$$

- El error de reconstrucción del conjunto de datos al proyectar a  $\mathbb{R}^k$  es:

$$\text{error}_k = \sum_{i=1}^n \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 = \sum_{i=1}^n (\mathbf{x}_i - \hat{\mathbf{x}}_i)^t (\mathbf{x}_i - \hat{\mathbf{x}}_i)$$

# Problema de optimización

- $W$  está formada por  $k$  vectores *columna*  $\mathbf{w}_j \in \mathbb{R}^{D \times 1}$
- $W$  define una base ortonormal donde  $W^t W = I$ :

$$\begin{aligned}\mathbf{w}_j^t \mathbf{w}_j &= 1 \\ \mathbf{w}_j^t \mathbf{w}_i &= 0 \quad \text{si } i \neq j\end{aligned}$$

- Con estas condiciones  $V = W^t$  minimiza el error de reconstrucción, es decir:

$$\hat{\mathbf{x}}_i = \bar{\mathbf{x}} + W W^t (\mathbf{x}_i - \bar{\mathbf{x}}) \quad \text{con } W \in \mathbb{R}^{D \times k}$$

- Objetivo: encontrar matriz de proyección  $W$  que minimice error de reconstrucción

$$\widehat{W} = \underset{W}{\operatorname{argmin}} \sum_{i=1}^n (\mathbf{x}_i - \hat{\mathbf{x}}_i)^t (\mathbf{x}_i - \hat{\mathbf{x}}_i)$$

# Problema de optimización

- Operando convenientemente, equivale a:

$$\begin{aligned}\widehat{W} &= \operatorname{argmin}_W \sum_{i=1}^n (\mathbf{x}_i - \hat{\mathbf{x}}_i)^t (\mathbf{x}_i - \hat{\mathbf{x}}_i) = \operatorname{argmin}_W \sum_{j=k+1}^D \mathbf{w}_j^t \Sigma_{\mathcal{X}} \mathbf{w}_j \\ &= \operatorname{argmax}_W \sum_{j=1}^k \mathbf{w}_j^t \Sigma_{\mathcal{X}} \mathbf{w}_j\end{aligned}$$

donde  $\Sigma_{\mathcal{X}}$  es la matriz de covarianza de los datos.

- Por simplicidad, consideramos un único vector  $\mathbf{w}$  que minimiza el error de reconstrucción cuando proyectamos a una única dimensión

$$\hat{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w} \in \mathbb{R}^D} \mathbf{w}^t \Sigma_{\mathcal{X}} \mathbf{w} \quad \text{sujeto a que} \quad \mathbf{w}^t \mathbf{w} = 1$$

# Problema de optimización

- Maximización con restricciones, resolución por *multiplicadores de Lagrange*

$$\hat{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w} \in \mathbb{R}^D} \max_{\lambda \in \mathbb{R}} \mathbf{w}^t \Sigma_{\mathcal{X}} \mathbf{w} + \lambda (1 - \mathbf{w}^t \mathbf{w})$$

- Tras derivar respecto a  $\mathbf{w}$  y  $\lambda$ , e igualar a cero, se tiene que

$$\Sigma_{\mathcal{X}} \mathbf{w} = \lambda \mathbf{w}$$

- Es decir,  $\mathbf{w}$  es un vector propio de la matriz de covarianza de los datos  $\Sigma_{\mathcal{X}}$  y  $\lambda$  su valor propio asociado.

Detalles de los cálculos en documento en PoliformaT

# Problema de optimización

- ¿Qué vectores propios se deben usar en la proyección para minimizar el error de reconstrucción?
- Para minimizar el error de reconstrucción al proyectar a una dimensión:

$$\hat{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w} \in \mathbb{R}^D} \max_{\lambda \in \mathbb{R}} \mathbf{w}^t \Sigma_{\mathcal{X}} \mathbf{w} + \lambda (1 - \mathbf{w}^t \mathbf{w})$$

- Como  $\Sigma_{\mathcal{X}} \mathbf{w} = \lambda \mathbf{w}$  y  $\mathbf{w}^t \mathbf{w} = 1$

$$\hat{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w} \in \mathbb{R}^D} \max_{\lambda \in \mathbb{R}} \mathbf{w}^t \lambda \mathbf{w} + \lambda (1 - \mathbf{w}^t \mathbf{w}) = \operatorname{argmax}_{\mathbf{w} \in \mathbb{R}^D} \max_{\lambda \in \mathbb{R}} \lambda$$

- Por tanto, se minimiza el error de reconstrucción al proyectar a una única dimensión cuando se toma el vector propio  $\mathbf{w}$  asociado al mayor valor propio

# Problema de optimización

- Para obtener el segundo vector propio que minimiza el error de reconstrucción al proyectar a dos dimensiones, lo expresaríamos como:

$$\hat{\mathbf{w}}_2 = \operatorname{argmax}_{\mathbf{w}_2 \in \mathbb{R}^D} \mathbf{w}_2^t \Sigma_{\mathcal{X}} \mathbf{w}_2 \quad \text{sujeto a que} \quad \mathbf{w}_2^t \mathbf{w}_2 = 1 \quad \text{y} \quad \mathbf{w}_1^t \mathbf{w}_2 = 0$$

es decir, el segundo vector de proyección debe ser unitario y ortogonal a  $\mathbf{w}_1$

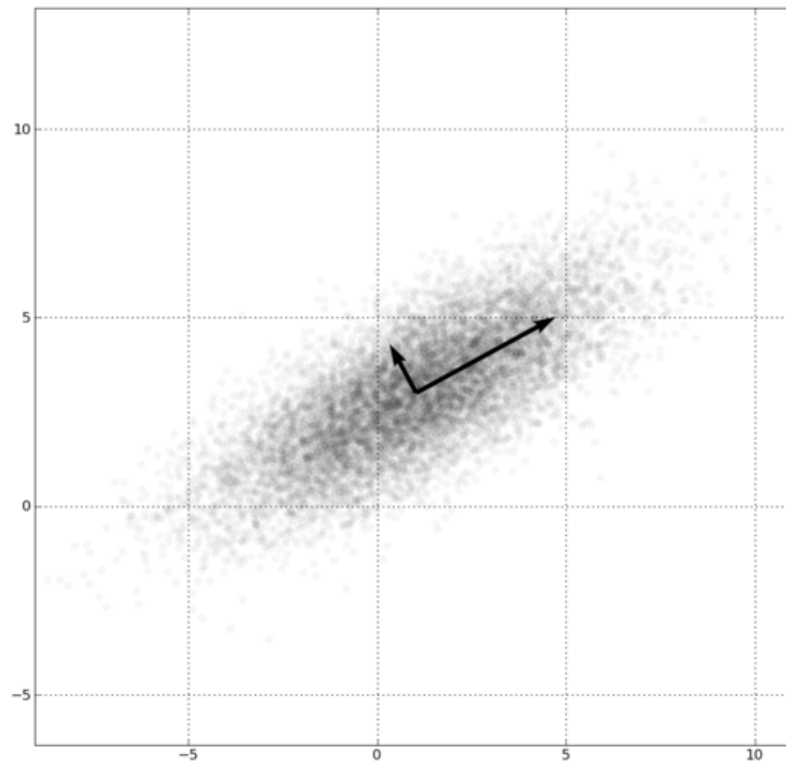
- Proceso iterativo de cálculo de vectores de proyección que se resume en:
  - Cada vector propio  $\mathbf{w}_j$  tienen un valor propio asociado  $\lambda_j$
  - $\widehat{W}$  son los vectores propios de  $\Sigma_{\mathcal{X}}$  de mayor a menor valor propio
  - $\Sigma_{\mathcal{X}}$  puede tener hasta  $D$  vectores propios (si es de rango completo)
- El error de reconstrucción cuando se proyecta a  $k$  dimensiones es:

$$\text{error}_k = \sum_{j=k+1}^D \mathbf{w}_j^t \Sigma_{\mathcal{X}} \mathbf{w}_j = \sum_{j=k+1}^D \mathbf{w}_j^t \lambda_j \mathbf{w}_j = \sum_{j=k+1}^D \lambda_j$$



# Interpretación gráfica de PCA

La proyección PCA se hace en la dirección de los vectores propios de mayor a menor valor propio, es decir, en las direcciones del espacio de mayor a menor varianza



- El primer vector propio  $w_1$  indica la dirección del espacio donde hay mayor varianza ( $\lambda_1$ ) en los datos
- El segundo vector propio  $w_2$  indica una dirección del espacio ortogonal a  $w_1$  donde reside la mayor cantidad de varianza restante (no capturada en  $\lambda_1$ )

# Problema de optimización. Resumen

- Para *minimizar el error de reconstrucción* a  $k$  dimensiones se proyecta con  $W = (\mathbf{w}_1 \dots \mathbf{w}_k)$  que son los  $k$  vectores propios de  $\Sigma_{\mathcal{X}}$  con *mayor valor propio asociado*
- Dado  $\mathbf{y} \in \mathbb{R}^D$  para proyectarlo a  $k$  dimensiones

$$\begin{array}{ccccc} \mathbf{y}' & = & W^t & \cdot & (\mathbf{y} - \bar{\mathbf{x}}) \\ k \times 1 & & k \times D & & D \times 1 \end{array}$$

siendo  $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$

# Algoritmo PCA

- Entrada:  $n, D, k, \mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$
- Salida:  $W, \bar{\mathbf{x}}$
- Algoritmo:
  1. Calcular la media de los datos:  $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$
  2. Restar a todos los datos la media:  $\mathbf{x}_i \leftarrow \mathbf{x}_i - \bar{\mathbf{x}}$
  3. Calcular la matriz de covarianza:  $\Sigma_{\mathcal{X}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i (\mathbf{x}_i)^t$
  4. Encontrar todos los vectores propios de  $\Sigma_{\mathcal{X}}$
  5. Ordenarlos descendientemente según los valores propios asociados
  6. Definir  $W$  como la matriz con los  $k$  primeros vectores propios
- Importante: para aplicar la proyección lineal a cualquier otro dato  $\mathbf{y} \in \mathbb{R}^D$  hay que restarle previamente la media estimada en el paso 1

# Diagonalización y varianza

- En general PCA es el proceso que diagonaliza la matriz de covarianzas:

$$\Sigma_{\mathcal{X}} W = W \Lambda \quad \rightarrow \quad \Sigma_{\mathcal{X}} = W \Lambda W^t \rightarrow \quad \Lambda = W^t \Sigma_{\mathcal{X}} W$$

donde:

- $\Sigma_{\mathcal{X}} \in \mathbb{R}^{D \times D}$  es la matriz de covarianza de los datos
  - $W \in \mathbb{R}^{D \times D}$  es la matriz con todos los vectores propios
  - $\Lambda \in \mathbb{R}^{D \times D}$  es una matriz diagonal con los todos valores propios
- $\Lambda$  es la matriz de covarianzas de los datos proyectados por  $W \in \mathbb{R}^{D \times D}$
  - Los datos proyectados están decorrelados (covarianzas nulas), y la varianza de estos datos en la dimensión  $j$  es  $\Lambda_{jj} = \lambda_j$
  - La varianza total acumulada de los datos proyectados a  $k$  dimensiones es:

$$\sum_{j=1}^k \Lambda_{jj} = \sum_{j=1}^k \lambda_j$$

# Consideraciones prácticas

- Sean:
  - $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ , datos de entrenamiento
  - $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ , media de dichos datos
  - $A_{D \times n} = (\mathbf{x}_1 - \bar{\mathbf{x}}, \mathbf{x}_2 - \bar{\mathbf{x}}, \dots, \mathbf{x}_n - \bar{\mathbf{x}})$
- $\Sigma_{\mathcal{X}}$  puede expresarse como  $\Sigma_{\mathcal{X}} = \frac{1}{n} A A^t$
- PCA consiste en obtener  $W$  y  $\Lambda$  que diagonalicen  $\Sigma_{\mathcal{X}}$
- Problema: cuando  $n \ll D$ .
  - $\Sigma_{\mathcal{X}}$  no tendrá más de  $n$  vectores propios
  - Almacenar  $\Sigma_{\mathcal{X}} \in \mathbb{R}^{D \times D}$  puede ser problemático con  $D$  grande
- Solución: diagonalizar  $\Sigma'_{\mathcal{X}} = \frac{1}{D} A^t A$ , con  $\Sigma'_{\mathcal{X}} \in \mathbb{R}^{n \times n}$
- Pero hay que obtener obtener  $W$  y  $\Lambda$  que diagonalicen  $\Sigma_{\mathcal{X}}$

# Consideraciones prácticas

- Método:

- Obtener  $W'$  y  $\Lambda'$  de la diagonalización de  $\Sigma'_{\mathcal{X}}$
- Expresarlo en términos de la diagonalización de  $\Sigma_{\mathcal{X}}$

$$\Sigma'_{\mathcal{X}} W' = \frac{1}{D} A^t A W' = W' \Lambda'$$

multiplicamos a ambas partes de la igualdad por  $\frac{D}{n} A$

$$\boxed{\frac{1}{n} A A^t} A W' = \frac{D}{n} A W' \Lambda' \longrightarrow \Sigma_{\mathcal{X}} \quad \boxed{A W'} = \boxed{A W'} \quad \boxed{\frac{D}{n} \Lambda'}$$

$\downarrow$   
 $W$

$\downarrow$   
 $\Lambda$

- Por tanto,  $W = A W'$ ,  $\Lambda = \frac{D}{n} \Lambda'$
- Los vectores propios obtenidos son ortogonales pero no ortonormales
- Se debe dividir cada vector por su módulo para obtener vectores ortonormales

# Índice

- 1 Introducción ▷ 3
- 2 Proyecciones lineales ▷ 9
- 3 Vectores propios ▷ 17
- 4 Principal Component Analysis (PCA) ▷ 25
- 5 *Aplicación de PCA: EigenFaces* ▷ 39

# Aplicación de PCA: EigenFaces

- Muestras  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  que representan imágenes de caras
- Representación global:  $\mathbf{x}_i \in \mathbb{R}^D$ , con  $D$  número de píxeles de la imagen
- Los vectores propios de la matriz de covarianza tienen esta apariencia:



<sup>2</sup>Imágenes de <http://www.chrisdecoro.com/eigenfaces/>



# Aplicación de PCA: EigenFaces

- Toda imagen  $\mathbf{x}_i$  puede ser reconstruida como una suma ponderada de una selección de los  $k$  primeros vectores propios (imágenes)  $\hat{\mathbf{x}}_i = \bar{\mathbf{x}} + WW^t(\mathbf{x}_i - \bar{\mathbf{x}})$
- A mayor valor de  $k$  mejor reconstrucción:



Resultado de la reconstrucción usando  $k = 1 \dots 25$  vectores propios

# Aplicación de PCA: EigenFaces

- En general, se puede reconstruir cualquier imagen del mismo tamaño que las imágenes empleadas en la obtención de los vectores propios
- La reconstrucción siempre tenderá a ser una cara:

