

# PCA, Método de la potencia y Machine Learning 101

Métodos Numéricos

Nicolás Roulet

Primer cuatrimestre - 2018

# Dónde estamos y qué vimos hasta ahora

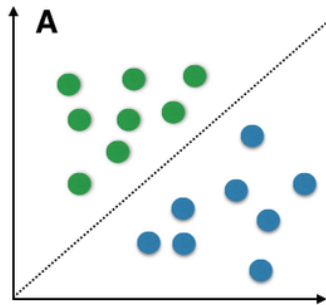
- ▶ Errores numéricos.
- ▶ Resolución de sistema lineales. (TP1: EG, EDD, etc.)
- ▶ Aplicación de resolución de sistemas (PageRank).
- ▶ LU, Cholesky, SDP, Matrices ortogonales.

# Lo que se viene: ML

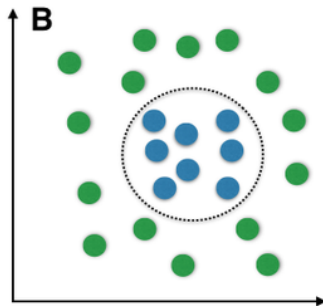
¿Tan distinto es?



# Problemas de clasificación



$$f : \mathbb{R}^2 \longrightarrow \{\text{verde}, \text{azul}\}$$





















































# En el menú de hoy

Reconocimiento de dígitos - Aplicaciones



# Ejercicio

## Reconocimiento de dígitos

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 0: |  |  |  |  |  |
| 1: |  |  |  |  |  |
| 2: |  |  |  |  |  |
| 3: |  |  |  |  |  |
| 4: |  |  |  |  |  |
| 5: |  |  |  |  |  |
| 6: |  |  |  |  |  |
| 7: |  |  |  |  |  |
| 8: |  |  |  |  |  |
| 9: |  |  |  |  |  |

### Problema a resolver

Recibimos un nuevo dígito manuscrito, ¿Podemos determinar automáticamente cuál es?

# Reconocimiento de dígitos

## Contexto

### Objetivo

Implementar un *clasificador* que permita reconocer dígitos manuscritos.

*Piensen en programar esto con un algoritmo convencional...*

### Contexto

- ▶ Disponemos de una base de datos etiquetada de dígitos manuscritos (sabemos qué dígito representa cada uno).
- ▶ Utilizaremos el dataset MNIST, donde tenemos 70k dígitos.
- ▶ Cada dígito es una imagen en escala de grises de  $28 \times 28$ .

# Aprendizaje supervisado

Vamos a idear un algoritmo que utilice estos datos para obtener la respuesta.

La idea es que si nos dan un dígito que no está en nuestros datos, podamos inferirlo igual. Podemos decir que nuestro algoritmo aprenderá de los datos.

Esto se llama **aprendizaje supervisado**, porque los datos que tenemos están anotados con la respuesta correcta.



# Reconocimiento de dígitos

Vecino más cercano

## Idea general (caso particular reconocimiento dígitos)

- ▶ Consideramos cada imagen como un vector  $x_i \in \mathbb{R}^m$ ,  $m = 28 \times 28$ ,  $i = 1, \dots, n$ . Para las imágenes en la base de datos, sabemos además a que clase pertenece.
- ▶ Cuando llega una nueva imagen de un dígito  $z$ , con el mismo formato, recorremos toda la base y buscamos aquella que minimice

$$\arg \min_{i=1, \dots, n} \|z - x_i\|_2$$

Luego, le asignamos la clase del representante seleccionado.

## Generalización

Considerar más de un vecino.

# Reconocimiento de dígitos

Vecinos más cercanos:  $kNN$

- ▶ Consideramos los  $k$  vecinos más cercanos.
- ▶ Entre ellos hacemos una votación, eligiendo como clase la *moda* del conjunto. En otras palabras, hacemos una votación y se elige aquella clase con más votos.

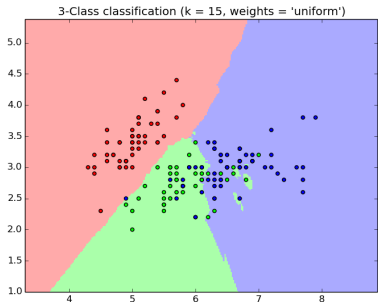


Veamos el código

# Reconocimiento de dígitos

*k*NN: Ejemplo de clasificación y definición de fronteras

## Algunos pros & cons



- + Es conceptualmente simple.
- + Funciona bien en general para dimensiones bajas, y puede ser utilizado con pocos ejemplos.
- Sufre de *La maldición de la dimensionalidad*.
- La clasificación puede ser lenta dependiendo del contexto.

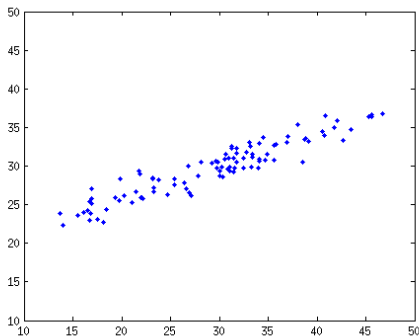
Imagen tomada de [SCIKIT-LEARN.ORG](http://SCIKIT-LEARN.ORG)

# Análisis de Componentes Principales

Ejemplo datos en  $\mathbb{R}^2$

Sean  $x^{(1)}, x^{(2)}, \dots, x^{(n)}$  una secuencia de  $n$  datos, con  $x^{(i)} \in \mathbb{R}^2$ .

$$X = \begin{bmatrix} x^{(1)t} \\ x^{(2)t} \\ x^{(3)t} \\ x^{(4)t} \\ x^{(5)t} \\ x^{(6)t} \\ \vdots \\ x^{(n)t} \end{bmatrix} = \begin{bmatrix} 26.4320 & 27.7740 \\ 26.8846 & 26.5631 \\ 23.3309 & 26.6983 \\ 30.6387 & 31.5619 \\ 30.5171 & 30.8993 \\ 45.6364 & 36.6035 \\ \vdots & \vdots \\ 16.0650 & 24.0210 \end{bmatrix}$$



# Análisis de Componentes Principales

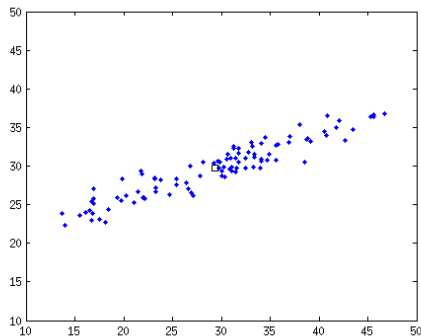
Ejemplo datos en  $\mathbb{R}^2$

$$X = \begin{bmatrix} 26.4320 & 27.7740 \\ 26.8846 & 26.5631 \\ 23.3309 & 26.6983 \\ 30.6387 & 31.5619 \\ 30.5171 & 30.8993 \\ 45.6364 & 36.6035 \\ \vdots & \vdots \\ 16.0650 & 24.0210 \end{bmatrix}$$

Media:

$$\mu = \frac{1}{n}(x^{(1)} + \dots + x^{(n)})$$

$$\mu = (29.3623, 29.7148)$$



Varianza de una variable  $x_k$ : Medida para la dispersión de los datos.

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_k^{(i)} - \mu_k)^2$$

$$\sigma_{x_1}^2 = 66.2134, \quad \sigma_{x_2}^2 = 12.5491$$

# Análisis de Componentes Principales

Ejemplo datos en  $\mathbb{R}^2$  - Covarianza

Covarianza: Medida de cuánto dos variables varían de forma similar. Variables con mayor covarianza inducen la presencia de cierta dependencia o relación.

$$\sigma_{x_j x_k} = \frac{1}{n-1} \sum_{i=1}^n (x_j^{(i)} - \mu_j)(x_k^{(i)} - \mu_k)$$

# Análisis de Componentes Principales

## Ejemplo datos en $\mathbb{R}^2$ - Covarianza

Dadas  $n$  observaciones de dos variables  $x_1$ ,  $x_2$ , y  $v = (1, \dots, 1)^t$ :

$$\sigma_{x_1 x_2} = \frac{1}{n-1} \sum_{i=1}^n (x_1^{(i)} - \mu_1)(x_2^{(i)} - \mu_2) = \frac{1}{n-1} (x_2 - \mu_2 v)^t (x_1 - \mu_1 v)$$

Matriz de Covarianza:

$$X = \begin{bmatrix} 26.4320 - \mu_1 & 27.7740 - \mu_2 \\ 26.8846 - \mu_1 & 26.5631 - \mu_2 \\ 23.3309 - \mu_1 & 26.6983 - \mu_2 \\ 30.6387 - \mu_1 & 31.5619 - \mu_2 \\ 30.5171 - \mu_1 & 30.8993 - \mu_2 \\ 45.6364 - \mu_1 & 36.6035 - \mu_2 \\ \vdots & \vdots \\ 16.0650 - \mu_1 & 24.0210 - \mu_2 \end{bmatrix}$$
$$M_X = \frac{1}{n-1} X^t X = \begin{bmatrix} \sigma_{x_1 x_1} & \sigma_{x_1 x_2} \\ \sigma_{x_1 x_2} & \sigma_{x_2 x_2} \end{bmatrix}$$
$$= \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} \\ \sigma_{x_1 x_2} & \sigma_{x_2}^2 \end{bmatrix}$$
$$M_X = \begin{bmatrix} 66.2134 & 27.1263 \\ 27.1263 & 12.5491 \end{bmatrix}$$

# ¿Cómo expresar mejor nuestros datos?

## Objetivo

Buscamos una transformación de los datos que disminuya la redundancia (es decir, disminuir la covarianza).

- ▶ Cambio de base:  $\hat{X}^t = PX^t$ .
- ▶ Cómo podemos hacerlo? Diagonalizar la matriz de covarianza. Esta matriz tiene la varianza de cada variable en la diagonal, y la covarianza en las restantes posiciones. Luego, al diagonalizar buscamos variables que tengan covarianza cero entre sí y la mayor varianza posible.



# Autovalores y Autovectores

## Definición

Sea  $A \in \mathbb{R}^{n \times n}$ . Un *autovector* de  $A$  es un vector no nulo tal que  $Ax = \lambda x$ , para algún escalar  $\lambda$ . Un escalar  $\lambda$  es denominado *autovalor* de  $A$  si existe una solución no trivial  $x$  del sistema  $Ax = \lambda x$ . En este caso,  $x$  es llamado *autovector asociado a  $\lambda$* .

Consideramos:

$$A = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}, u = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, v = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$Au = \begin{bmatrix} -5 \\ -1 \end{bmatrix}, Av = \begin{bmatrix} 4 \\ 2 \end{bmatrix} = 2 \begin{bmatrix} 2 \\ 1 \end{bmatrix} = 2v$$

Gráficamente.... $A$  sólo estira (o encoge) el vector  $v$ .

# Diagonalización

En muchos casos, la presencia de autovectores-autovalores puede ser utilizada para encontrar una factorización  $A = PDP^{-1}$ , donde  $D$  es una matriz diagonal.

## Intuición

Podemos encontrar una base donde la transformación lineal  $A$  se comporta como si fuese diagonal.

## Observación

No toda matriz  $A \in \mathbb{R}^{n \times n}$  es diagonalizable.

## Teorema

Una matriz  $A \in \mathbb{R}^{n \times n}$  es diagonalizable sí y solo sí  $A$  tiene  $n$  autovectores linealmente independientes (las columnas de  $P$ ).

## Teorema

Si  $A \in \mathbb{R}^{n \times n}$  es simétrica, entonces existe una base ortonormal de autovectores  $\{v_1, \dots, v_n\}$  asociados a  $\lambda_1, \dots, \lambda_n$ .

Consecuencia: Existe  $P$ , y  $P^{-1} = P^t$ . Luego,  $A = PDP^t$ .

# Cálculo de autovalores/autovectores

- ▶ La matriz de covarianza  $M_X = \frac{1}{n-1}X^tX$  es simétrica y semidefinida positiva.
- ▶ Vamos a querer diagonalizar  $M_X$  para obtener la transformación que queremos. Para eso vamos a calcular sus autovectores.
- ▶ Podemos considerar el Método de la Potencia para calcular  $\lambda_1$  y  $v_1$ .

1. MetodoPotencia( $B, x_0, niter$ )
2.  $v \leftarrow x_0$ .
3. Para  $i = 1, \dots, niter$
4.  $v \leftarrow \frac{Bv}{||Bv||}$
5. Fin Para
6.  $\lambda \leftarrow \frac{v^t B v}{v^t v}$
7. Devolver  $\lambda, v$ .

# Cálculo de autovalores/autovectores

Una vez que tenemos  $\lambda_1$  y  $v_1$ , como seguimos?

## Deflación

Sea  $B \in \mathbb{R}^{n \times n}$  una matriz con autovalores distintos

$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$  y una base ortonormal de autovectores.

Entonces, la matriz  $B - \lambda_1 v_1 v_1^t$  tiene autovalores  $0, \lambda_2, \dots, \lambda_n$  con autovectores asociados  $v_1, \dots, v_n$ .

- ▶  $(B - \lambda_1 v_1 v_1^t)v_1 = Bv_1 - \lambda_1 v_1(v_1^t v_1) = \lambda_1 v_1 - \lambda_1 v_1 = 0v_1$ .
- ▶  $(B - \lambda_1 v_1 v_1^t)v_i = Bv_i - \lambda_1 v_1(v_1^t v_i) = \lambda_i v_i$ .

## Observación

En nuestro caso, no hace falta que todos los autovalores tengan magnitudes distintas.

## ¿Cómo expresar mejor nuestros datos?

- Cambio de base:  $\hat{X}^t = PX^t$ .

Sea  $P$  ortogonal y  $M_{\hat{X}}$  la matriz de covarianza de  $\hat{X}$ .

$$\begin{aligned}M_{\hat{X}} &= \frac{1}{n-1} \hat{X}^t \hat{X} \\&= \frac{1}{n-1} (PX^t)(XP^t) \\&= P \frac{X^t X}{n-1} P^t \\&= PM_X P^t\end{aligned}$$

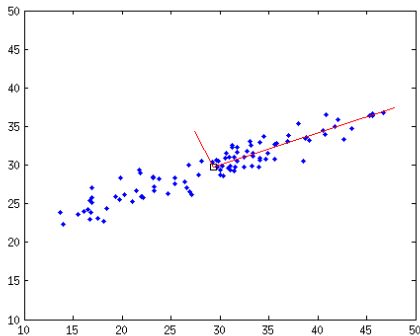
- $M_X$  es simétrica, entonces existe  $V$  ortogonal tal que  $M_X = VDV^t$ .

$$\begin{aligned}M_{\hat{X}} &= PM_X P^t \\&= P(VDV^t)P^t \quad \text{tomamos } P = V^t \\&= (V^t V)D(V^t V) = D\end{aligned}$$

# ¿Cómo expresar mejor nuestros datos?

Volvemos al ejemplo

$$\begin{aligned} M_X &= \begin{bmatrix} 66.2134 & 27.1263 \\ 27.1263 & 12.5491 \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} 0.9228 & -0.3852 \\ 0.3852 & 0.9228 \end{bmatrix}}_V \underbrace{\begin{bmatrix} 77.5362 & 0 \\ 0 & 1.2263 \end{bmatrix}}_{D=M_{\hat{X}}} \underbrace{\begin{bmatrix} 0.9228 & 0.3852 \\ -0.3852 & 0.9228 \end{bmatrix}}_{V^t} \end{aligned}$$



# Análisis de Componentes Principales

## Resumen hasta acá

- ▶ Tenemos  $n$  muestras de  $m$  variables.
- ▶ Calculamos el vector  $\mu$  que contiene la media de cada una de las variables.
- ▶ Construimos la matriz  $X \in \mathbb{R}^{n \times m}$  donde cada muestra corresponde a una fila de  $X$  y tienen media cero (i.e.,  $X_i := (x^{(i)} - \mu) / \sqrt{n-1}$ ).
- ▶ Diagonalizamos la matriz de covarianzas  $M_X$ . La matriz  $V$  (ortogonal) contiene los autovectores de  $M_X$ .

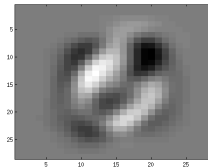
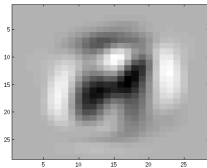
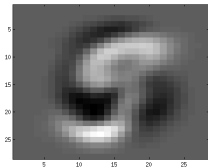
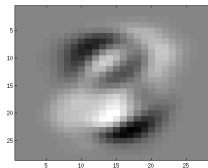
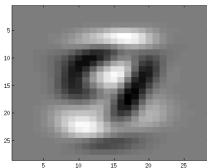
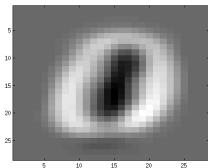
## Propiedades del cambio de base

- ▶ Disminuye redundancias.
- ▶ El cambio de base  $\hat{X}^t = PX^t = V^t X^t$  asigna a cada muestra un nuevo *nombre* mediante un cambio de coordenadas.
- ▶ Las columnas de  $V$  (autovectores de  $M_X$ ) son las componentes principales de los datos.
- ▶ En caso de  $m$  grande, es posible tomar sólo un subconjunto de las componentes principales para estudiar (i.e., aquellas que capturen mayor proporción de la varianza de los datos).

# Reconocimiento de dígitos

## Autodígitos (Eigendigits)

Los primeros 6 autovectores en  $V$ .





# Reconocimiento de dígitos

¿Cómo reconocemos un dígito?

## Idea

- ▶ Utilizar el cambio de base, transformando cada imagen convenientemente.
- ▶ Reducir la dimensión de los datos utilizando sólo algunas de las nuevas variables (eligiendo aquellas que capturan una fracción mayor de la varianza).

## Procedimiento

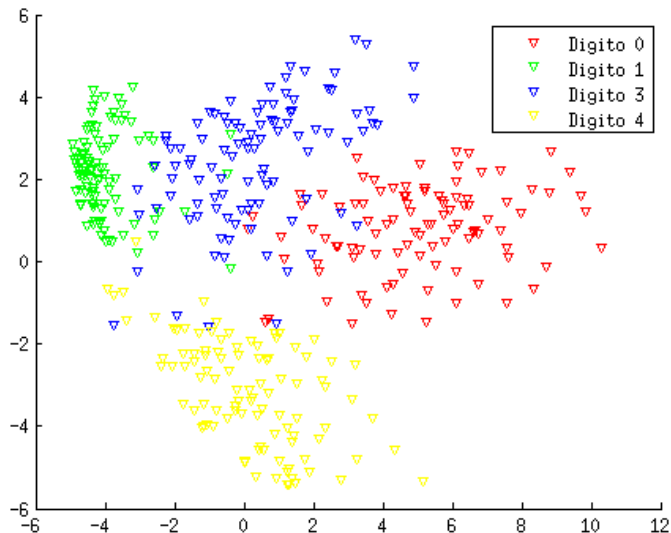
- ▶ Reducción de la dimensión: parámetro de entrada que indica cuántas componentes principales considerar,  $\alpha$ . Es decir, tomaremos  $\bar{V} = [v_1 \ v_2 \ \dots \ v_\alpha]$ .
- ▶ Transformación característica: Aplicamos el cambio de base a cada muestra  $x^{(i)}$ , definimos  $tc(x^{(i)}) = \bar{V}^t x^{(i)} = (v_1^t x^{(i)}, \dots, v_\alpha^t x^{(i)})$ . Matricialmente,  $tc(X) = (V^t X^t)^t = XV$ .

# Taller

jA codear!

# Reconocimiento de dígitos

Transformación + Reducción ( $k = 2$ )



# Reconocimiento de dígitos

¿Cómo reconocemos un dígito?

Finalmente, dada una imagen de un dígito que no se encuentra en la base:

- ▶ Vectorizamos la imagen en  $x^* \in \mathbb{R}^m$ .
- ▶ Definimos  $\bar{x}^* = (x^* - \mu) / \sqrt{n-1}$ .
- ▶ Aplicamos la transformación característica,  $tc(\bar{x}^*)$  y buscamos (de alguna manera) a que dígito pertenece.

# Matriz de confusión

Matriz de confusión o matriz de errores: es una forma de visualizar el desempeño del algoritmo. Es una matriz  $C \in \mathbb{R}^{p \times p}$  ( $p$  es la cantidad de clases), donde  $C_{ij}$  indica la cantidad de elementos para los que el algoritmo predijo la clase  $i$ , cuando en realidad la respuesta correcta era  $j$ .

Veamos un ejemplo en el código.

# Reconocimiento de dígitos

## Metodología de evaluación

Elegimos un número de vecinos  $k$  (adicionalmente un número  $\alpha$  de componentes). Como evaluamos si el método funciona?

- ▶ Como medimos la efectividad del método?

# Reconocimiento de dígitos

## Metodología de evaluación

Elegimos un número de vecinos  $k$  (adicionalmente un número  $\alpha$  de componentes). Como evaluamos si el método funciona?

- ▶ Como medimos la efectividad del método?
- ▶ Tiene sentido probarlo sobre la base de training?

# Reconocimiento de dígitos

## Metodología de evaluación

Elegimos un número de vecinos  $k$  (adicionalmente un número  $\alpha$  de componentes). Como evaluamos si el método funciona?

- ▶ Como medimos la efectividad del método?
- ▶ Tiene sentido probarlo sobre la base de training?
- ▶ De alguna forma defino una instancia, pruebo todas las combinaciones de parámetros sobre la misma. Es correcto? Puede surgir algún problema?



# Reconocimiento de dígitos

## Metodología de evaluación

Elegimos un número de vecinos  $k$  (adicionalmente un número  $\alpha$  de componentes). Como evaluamos si el método funciona?

- ▶ Como medimos la efectividad del método?
- ▶ Tiene sentido probarlo sobre la base de training?
- ▶ De alguna forma defino una instancia, pruebo todas las combinaciones de parámetros sobre la misma. Es correcto? Puede surgir algún problema?

## Idea

Utilizar la base de entrenamiento convenientemente para estimar y proveer suficiente evidencia respecto a la efectividad del método.