

# Trabajo Práctico 2

*Si nos organizamos aprobamos todos...*

Métodos Numéricos

Primer cuatrimestre - 2015

# Antes de pasar al TP2...

Donde estamos y qué vimos hasta ahora

- ▶ Errores numéricos.
- ▶ Diferencias finitas aplicado a sistemas complejos (TP1: EG, LU, sistemas banda)
- ▶ Refinamiento iterativo, número de condición.
- ▶ *Denoising* aplicado a imágenes (Taller 1: Matriz SDP, Cholesky).

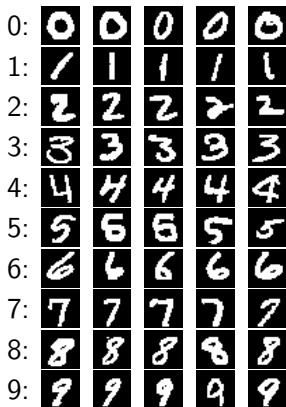
## Reconocimiento de dígitos - Aplicaciones



# Trabajo Práctico 2

## Reconocimiento de dígitos

- Datos: base de datos etiquetada de imágenes de dígitos manuscritos (0-9) tomadas de una forma particular.
- Objetivo: dada una nueva imagen de un dígito, ¿A cuál corresponde?



### Problema a resolver

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

# Reconocimiento de dígitos

## Contexto

### Objetivo

Desarrollar (no solo en términos de implementación) un *clasificador* que permita reconocer dígitos manuscritos.

### Contexto

- ▶ Disponemos de una base de datos etiquetada (train), y un conjunto de datos para los que no conocemos cual es su etiqueta (test). Este último nos permitirá evaluar como se comporta nuestro clasificador.
- ▶ Consideramos la base MNIST, en la versión utilizada en *Kaggle*. 42k dígitos en train, 18k dígitos en test.
- ▶ Cada dígito es una imagen en escala de grises de  $28 \times 28$ .

# 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, recorreremos 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*

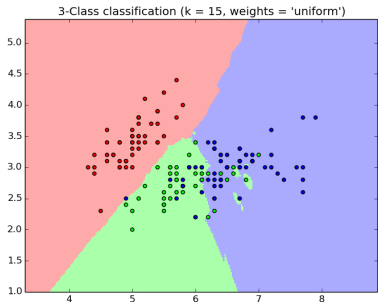
- ▶ Consideramos los  $k$  vecinos más cercanos.
- ▶ Entre ellos hacemos una votación, eligiendo como clase la *moda* del conjunto. En otras palabras, hacemos una votación y se elige aquella clase con más votos.



# 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)

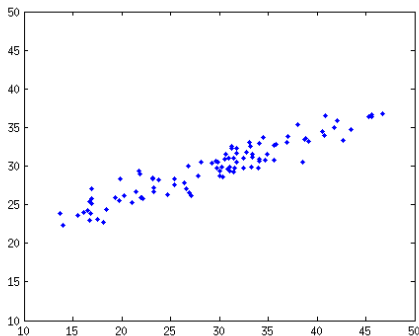


# Análisis de Componentes Principales

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

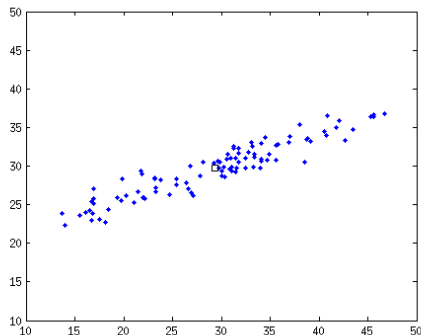
Ejemplo datos en  $\mathbb{R}^2$

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

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

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

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  $x_k$ ,  $x_j$ , y  $v = (1, \dots, 1)^t$ :

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

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,  $A = PDP^t$ .

## ¿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  $M_X = VDV^t$ .

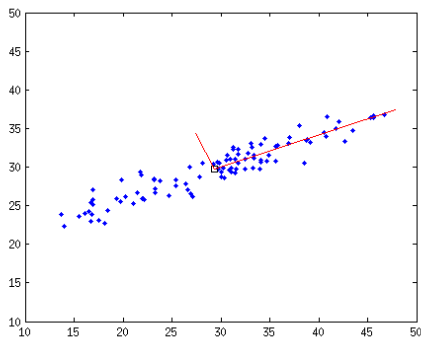
$$\begin{aligned}M_{\hat{X}} &= PM_X P^t \\&= P(VDV^t)P^t \quad \text{tomamos } P = V^t \\&= (V^t V)D(VV^t) = D\end{aligned}$$



# ¿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  $M_X$ .

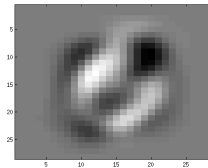
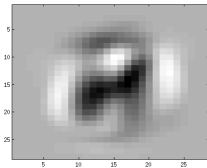
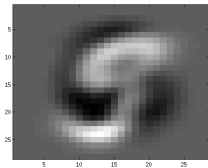
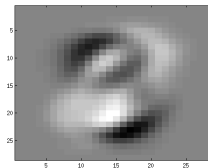
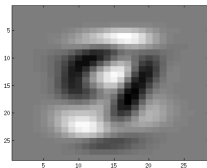
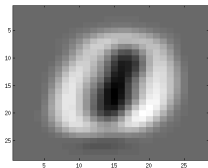
## Propiedades del cambio de base

- ▶ 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

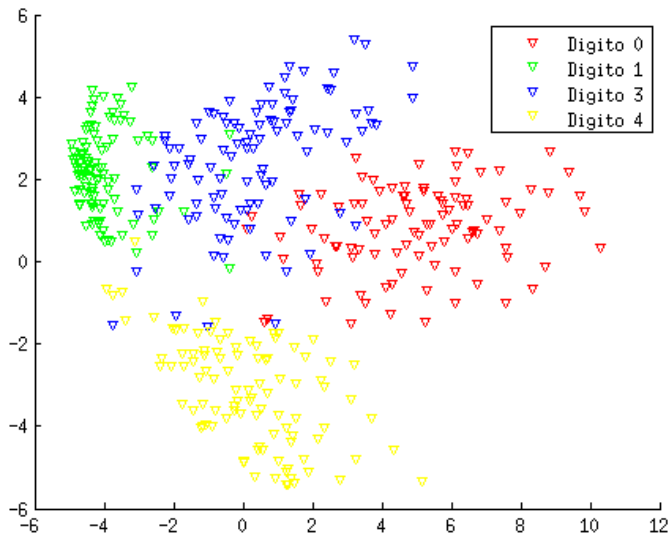
- ▶ Utilizar el cambio de base, transformando cada imagen convenientemente.
- ▶ Reducir la dimensión de los datos utilizando sólo algunas de las nuevas variables (eligiendo aquellas que capturan una fracción mayor de la varianza).

## 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  $tc(x^{(i)}) = \bar{V}^t x^{(i)} = (v_1^t x^{(i)}, \dots, v_\alpha^t x^{(i)})$ .

# 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 qué dígito pertenece.

Pregunta:

Sugerencias para buscar a qué dígito pertenece?

# Reconocimiento de dígitos

## Metodología de evaluación

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

- ▶ Como medimos la efectividad del método?

# Reconocimiento de dígitos

## Metodología de evaluación

Elegimos un numero de vecinos  $k$  (adicionalmente un número  $\alpha$  de componentes principales). 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 numero de vecinos  $k$  (adicionalmente un número  $\alpha$  de componentes principales). 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 numero de vecinos  $k$  (adicionalmente un número  $\alpha$  de componentes principales). 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.

# Reconocimiento de dígitos

## K-Fold Cross Validation

- ▶ Particionamos de forma aleatoria nuestra base de training en  $K$  conjuntos de igual tamaño.
- ▶ Se realizan  $K$  experimentos, cada uno de ellos reteniendo uno de los conjuntos para validación y utilizando los restantes  $K - 1$  para entrenamiento.
- ▶ Suelen realizarse varias corridas para un mismo valor de  $K$ .

Reportamos valores promedio de efectividad en el reconocimiento para cada combinación de parámetros.

## Sugerencia

Considerar el comando `CVPARTITION` de MATLAB.

# ¿Qué hay que hacer en el TP?

## Objetivos generales

- ▶ Implementar el método  $kNN$ .
- ▶ Implementar el método de Análisis de Componentes Principales, y combinarlo con  $kNN$ .
- ▶ Experimentar variando:  $k$ ,  $\alpha$ ,  $K$ , Analizar los resultados en términos de la tasa de efectividad aplicando *cross validation* sobre la base de training.
- ▶ Para encontrar la transformación  $V$ , utilizar el *Método de la Potencia + Deflación*. (Clase teórica Miercoles próximo)

# ¿Qué hay que hacer en el TP?

## Objetivos generales

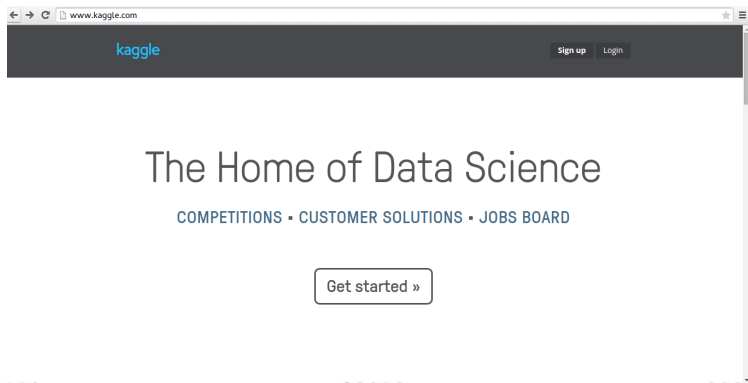
- ▶ Implementar el método  $kNN$ .
- ▶ Implementar el método de Análisis de Componentes Principales, y combinarlo con  $kNN$ .
- ▶ Experimentar variando:  $k$ ,  $\alpha$ ,  $K$ , Analizar los resultados en términos de la tasa de efectividad aplicando *cross validation* sobre la base de training.
- ▶ Para encontrar la transformación  $V$ , utilizar el *Método de la Potencia + Deflación*. (Clase teórica Miercoles próximo)

## Algunas (posibles) preguntas y dificultades

- ▶  $kNN$  y 42k imágenes de  $28 \times 28$ ?
- ▶ Tolerancia de corte Método de la Potencia? Se cumplen las condiciones para aplicar deflación?
- ▶ Cuántas componentes principales tomar?
- ▶ Que combinación de parámetros (modelo) da los mejores resultados?







# Por último...



Competencia activa en KAGGLE.COM



# Por último...

## Competencia activa en KAGGLE.COM

	<b>Bag of Words Meets Bags of Popcorn</b> Use Google's Word2Vec for movie reviews	Knowledge	383	2 months
	<b>Digit Recognizer</b> Classify handwritten digits using the famous MNIST data	Knowledge	571	8 months
	<b>Titanic: Machine Learning from Disaster</b> Predict survival on the Titanic (using Excel, Python, R, and Random Forests)	Knowledge	2562	8 months
	<b>Facial Keypoints Detection</b> Detect the location of keypoints on face images	Knowledge	79	8 months
	<b>First Steps With Julia</b> Identify characters from Google Street View Pictures + tutorial with Julia.	Knowledge	80	8 months
	<b>15.071x - The Analytics Edge (Spring 2015)</b> Test your analytics skills by predicting which New York Times blog articles will be the most popular	Private	1399	11 days

174	<span>-25</span>	max777alex	0.97486	2
175	<span>-25</span>	Ye Han	0.97471	14
176	<span>-25</span>	Subkhan - Denis 	0.97457	1
177	new	AIX1	0.97457	3
178	<span>-26</span>	Ohad Zadok	0.97414	3
179	<span>-25</span>	JMB	0.97400	2
180	<span>-25</span>	Qizhen	0.97400	8
181	<span>-25</span>	Rangudu Venkata Pavan Kumar 	0.97386	6
182	<span>-25</span>	Alexander Vasyuk	0.97371	4
183	<span>-25</span>	Muhammed Miah	0.97371	2
184	<span>-25</span>	Laurent Van Winckel	0.97357	4
185	<span>-25</span>	Chi-Ming Chang	0.97357	4
186	new	Vignesh Panneerselvam	0.97357	5
187	<span>-26</span>	Prashant Dheeraj	0.97343	9
188	<span>-26</span>	bowen	0.97329	2
189	<span>-18</span>	Constant	0.97314	3

# Cronograma sugerido

- ▶ Viernes 1 de Mayo: Lectura base de training, *kNN*, *cross validation*.
- ▶ Viernes 8 de Mayo: Método de la potencia, PCA, primeros experimentos.

## Fecha de entrega

- ▶ Formato electrónico: Jueves 14 de Mayo de 2015, hasta las 23:59 hs., enviando el trabajo (informe+código) a `metnum.lab@gmail.com`.
- ▶ Formato físico: Viernes 15 de Mayo de 2015, de 17:30 a 18:00 hs.