# Práctica 3: Agrupamiento

Tratamiento de Datos

8 de enero de 2015

## Introducción

En esta práctica analizaremos el comportamiento de algoritmos de agrupamiento "duro" y "blando", utilizando para ello una base de datos sintéticos en dos dimensiones. Concretamente, trabajaremos con el algoritmo $k$-medias, así como con versiones basadas en mezclas de Gaussianas, optimizadas mediante el algoritmo EM.

## 1. Agrupamiento "duro" con el algoritmo $k$-medias

Cargue los datos contenidos en el fichero `'datosP3.mat'`. El fichero contiene tres variables:

- `X`: Matriz $(2 \times 4000)$ de datos de entrenamiento.
- `true_model`: Struct de Matlab, que contiene el modelo utilizado para la generación de los datos sintéticos. Contiene en sus diversos campos la siguiente información: número de grupos, media y matriz de covarianzas de las densidades normales multidimensionales condicionadas a cada clase, y probabilidad a priori de cada grupo.
- `true_labels`: Vector $(1 \times 4000)$ que contiene la información de qué grupo generó cada uno de los datos de entrenamiento.

Nótese, que siendo éste un problema de agrupamiento, la información proporcionada en las variables `true_model` y `true_labels` no está habitualmente disponible (al menos no al completo), siendo precisamente el objetivo del problema obtener dicha información.

### 1.1. Visualización de los datos

Represente el diagrama de dispersión de los datos de entrenamiento. Represente los datos asociados a cada grupo utilizando un color diferente, utilizando para ello la información proporcionada en la variable `true_labels`. Represente asimismo el vector de medias para cada uno de los grupos.

### 1.2. Agrupamiento basado en "k-medias"

1. Aplique la función `kmeans` de Matlab para 6 grupos. Visualice los grupos generados como resultado de dicho agrupamiento, y calcule la distorsión media total:

$$J = \sum_{i=1}^{c} \sum_{\mathbf{x}_k \in \mathcal{C}_i} \|\mathbf{x}_k - \mu_i\|^2$$

2. Repita el procedimiento anterior para un total de 10 inicializaciones diferentes. Compruebe el carácter no determinista del algoritmo $k$-medias.

3. Aplique el algoritmo $k$-medias para un número variable de grupos entre 1 y 12, realizando un total de 10 inicializaciones diferentes para cada valor. Analice el comportamiento de la distorsión media promedio, así como el valor mínimo alcanzado en función del número de grupos.

## 2. Agrupamiento basado en modelo GMM con optimización EM

En esta parte de la práctica, los alumnos implementarán el 'paso M' de una optimización EM para un modelo de mezcla de Gaussianas. La implementación del 'paso E' se proporciona en la función `E_step.m` proporcionado junto con este enunciado. Dicho script recibe como entrada el conjunto de datos de entrenamiento y un modelo GMM, y proporciona dos salidas: el conjunto de probabilidades a posteriori para cada grupo y dato de entrada, y la log-verosimilitud del modelo medida sobre el conjunto de datos de entrada.

### 2.1. Optimización EM conocidas las matrices de covarianza

1. Inicialice el modelo a utilizar, utilizando un struct similar al `true_model` proporcionado. Utilice el siguiente criterio para inicializar el modelo: se considerarán 4 grupos equiprobables, matrices de covarianza conocidas e iguales a las del modelo verdadero, y vectores de medias tomados al azar entre todas las muestras de entrenamiento (le puede ser útil utilizar la función `randperm` de Matlab).

2. Implemente la optimización EM basada en la aplicación iterativa de los pasos E y M del algoritmo. Utilice para el paso E la función `E_step`, e implemente la actualización de los vectores de medias del modelo mediante maximización de la verosimilitud, que como se sabe está dada por la expresión siguiente:

$$\hat{\mu}_i(l) = \frac{\sum_k P(\omega_i|\mathbf{x}_k; \hat{\mu}(l-1))\ \mathbf{x}_k}{\sum_k P(\omega_i|\mathbf{x}_k; \hat{\mu}(l-1))}$$

3. Visualice la evolución de la localización de los vectores de medias según se realizan iteraciones del algoritmo EM, así como la evolución de la log-verosimilitud del modelo. Repita el experimento varias veces, de manera que pueda comprobar la influencia de la inicialización en los resultados obtenidos.

### 2.2. Optimización EM con inicialización basada en $k$-means

Con el objetivo de disponer de una inicialización más robusta para el EM, en esta sección se estudia una inicialización del modelo basada en el algoritmo $k$-medias. Implemente una función `init_model` que calcule un modelo inicial siguiendo los siguientes pasos:

- Para el número de grupos indicado, se debe ejecutar el algoritmo $k$-medias 10 veces, conservando la salida correspondiente a la ejecución que proporcione una menor distorsión media global.
- El modelo para el algoritmo EM debe inicializarse con los vectores de medias proporcionados por dicha mejor ejecución. Además, deben calcularse estimaciones de las matrices de covarianzas utilizando las muestras asignadas a cada grupo (puede utilizar para ello las funciones `cov` y `find` de Matlab).

Estudie el comportamiento del algoritmo EM utilizando la inicialización proporcionada por el algoritmo $k$-medias.

1. Considerando conocido el número de grupos $c = 4$, repita el experimento de la sección anterior, i.e., visualice la convergencia del algoritmo EM que actualiza únicamente los vectores de medias. Analice la evolución de la log-verosimilitud, y la velocidad de convergencia del algoritmo.
2. Repita el análisis anterior, implementando en esta ocasión la actualización de las matrices de covarianzas dentro del paso M del algoritmo:

$$\hat{\Sigma}_i(l) = \frac{\sum_k P(\omega_i|\mathbf{x}_k;\hat{\theta}(l-1))\ (\mathbf{x}_k - \hat{\mu}_i(l-1))(\mathbf{x}_k - \hat{\mu}_i(l-1))^T}{\sum_k P(\omega_i|\mathbf{x}_k;\hat{\theta}(l-1))}$$

   donde $\hat{\theta}$ representa un vector que contiene el conjunto de parámetros del modelo.

## 2.3. Determinación del número de grupos

Analice la evolución de la log-verosimilitud que se obtiene al aplicar la optimización EM a un modelo de mezcla de gaussianas, en función del número de grupos. Considere en su análisis inicialización mediante $k$-medias, y un número variable de grupos entre 1 y 12.

Analice también el comportamiento del BIC (Bayesian Information Criterion), que está dado por

$$\text{BIC} = -2L + n_p \log(n)$$

donde $L$ es la log-verosimilitud del modelo obtenido, $n_p$ es el número de parámetros del modelo, y $n$ es el número de muestras de entrenamiento.