Tecnologías para Inteligencia Artificial (247101009)
Tema 3. Clustering y detección de anomalías

Javier Vales Alonso
Máster Universitario en Ingeniería Telemática

2020

Universidad Politécnica de Cartagena

Tabla de contenido

¿Cómo estudiar esta unidad?

1. Haga una primera lectura de las diapositivas de la unidad. Concéntrese en ver las ideas generales y hacer una primera revisión de las matemáticas.
2. Haga una revisión a fondo de las matemáticas con las diapositivas y resuelva en el notebook los ejercicios indicados. Intente comprender todos los desarrollos involucrados. En caso de dudas, lea las referencias sugeridas (ver referencias en Tema 0) o contacte con el profesor.
3. Finalmente, envíe el notebook a través de AV.

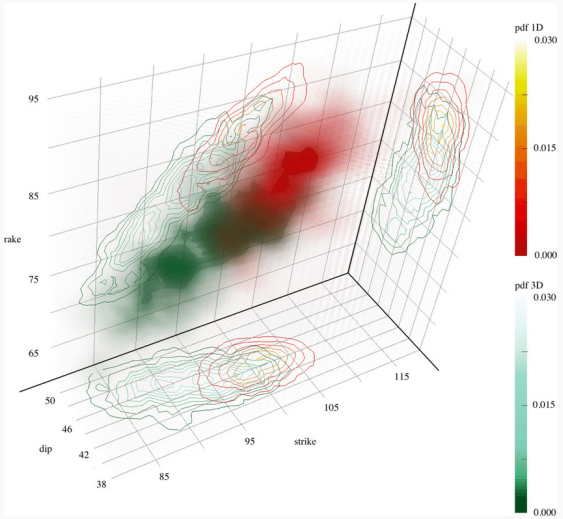
Introducción

¿Por qué el aprendizaje no supervisado?

Etiquetar los datos es un proceso errorso, largo, difícil, y altamente subestimado. Por ejemplo, etiquetar un conjunto de datos con millones de caras (ALEGRÍA/SORPRESA/TRISTEZA/etc.) es, cuando menos, engorroso, y propenso a errores si no se hace con cuidado. El resultado probable es obtener un conjunto de datos con un etiquetado incorrecto o usar muchos recursos (tiempo, dinero) para poder hacerlo.

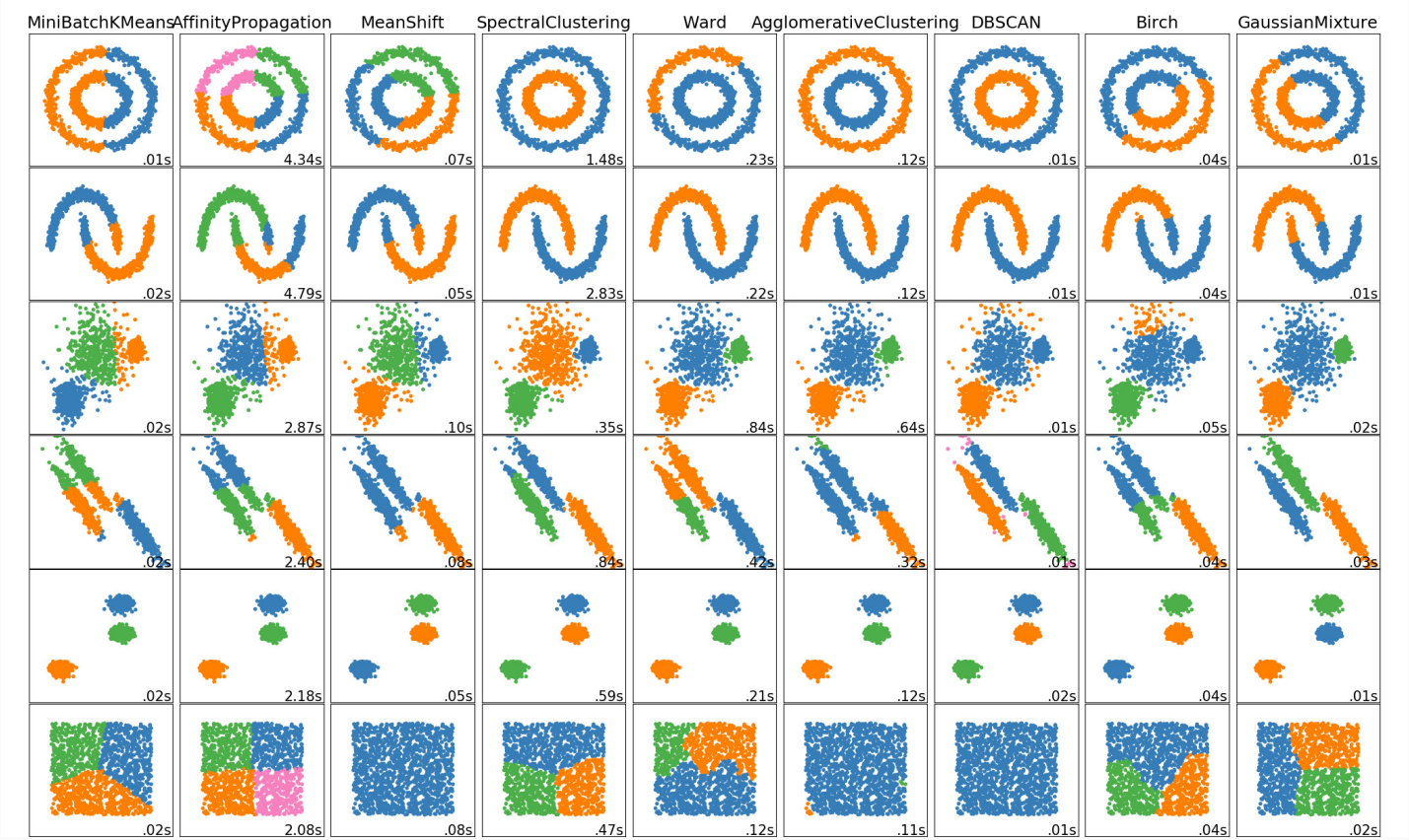
El aprendizaje no supervisado es una alternativa cuando hay datos disponibles, pero no etiquetados. Sus principales objetivos son descubrir similitudes (patrones), detección de anomalías, y preprocesamiento de datos (incluido el etiquetado automático).

Ejemplos de aprendizaje no supervisado



- Estimación de densidad.
- En aprendizaje supervisado se busca la distribución a posteriori $p(\boldsymbol{x}|t)$ (la distribución de puntos según su clase). Los métodos de estimación de densidad, tienen como objetivo inferir la distribución a priori $p(\boldsymbol{x})$.
- Conocer $p(\boldsymbol{x})$ es útil para visualización de datos, detección de anomalías, etc.

Ejemplos de aprendizaje no supervisado (II)

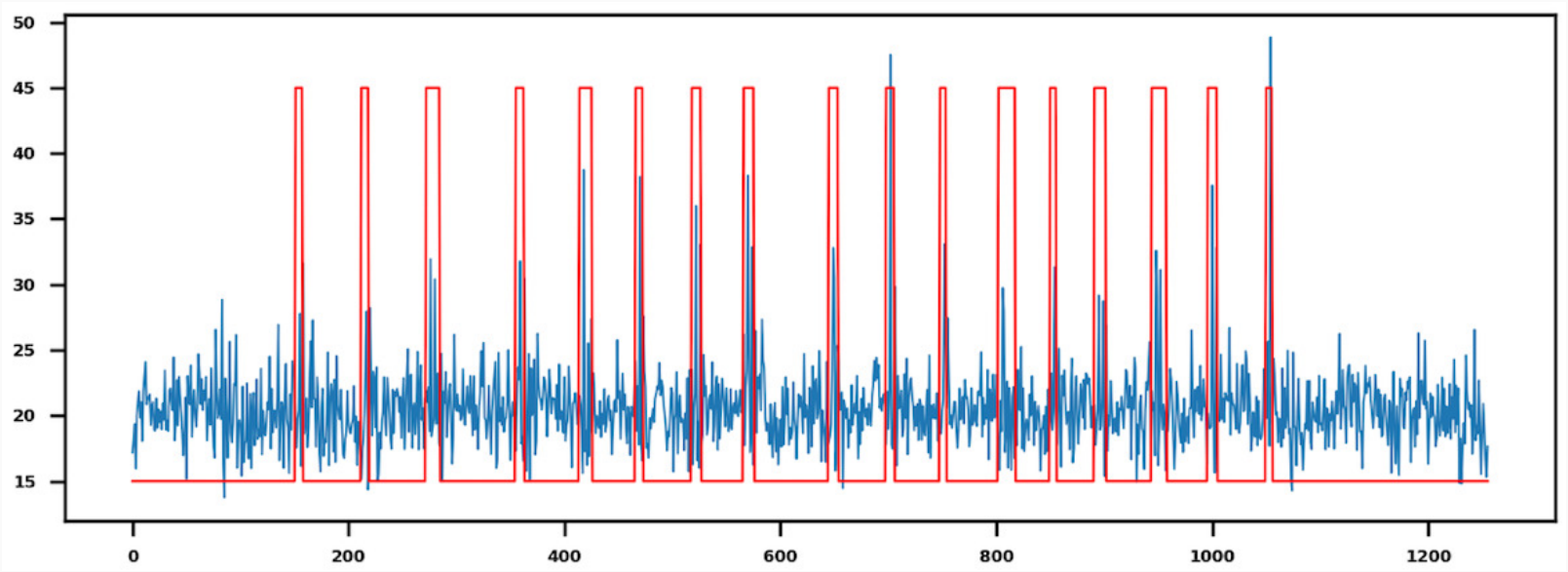


- Data clustering.
- Agrupar los datos por similitud.
- Esencial para data Mining en muchos campos científicos, e.g., finanzas, astronomía, biología, etc.

Ejemplos de aprendizaje no supervisado (III)

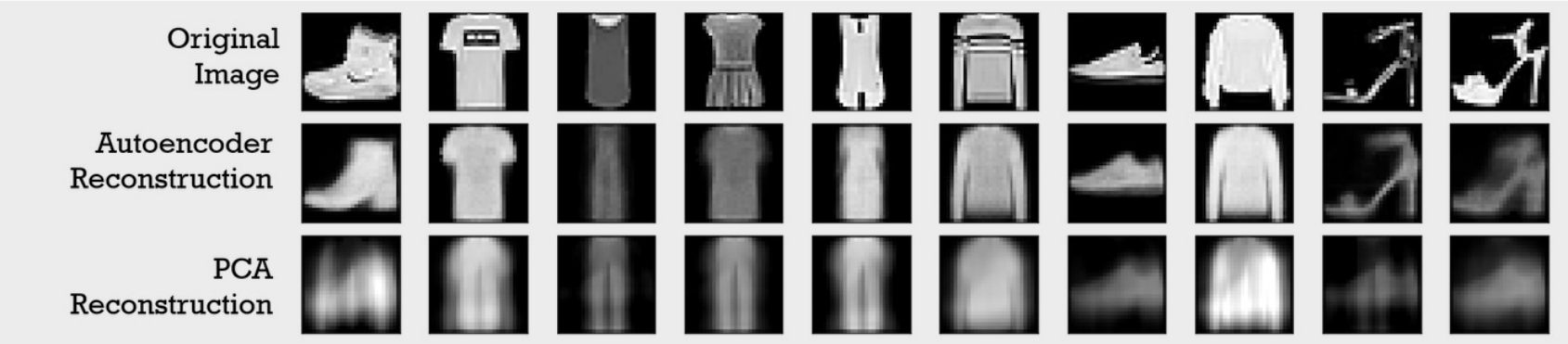
- Segmentación de imagen y "cuantización" de datos.
- Los píxeles similares se agrupan juntos.
- Útil para preprocesamiento de imágenes y compresión de datos.

Ejemplos de aprendizaje no supervisado (IV)



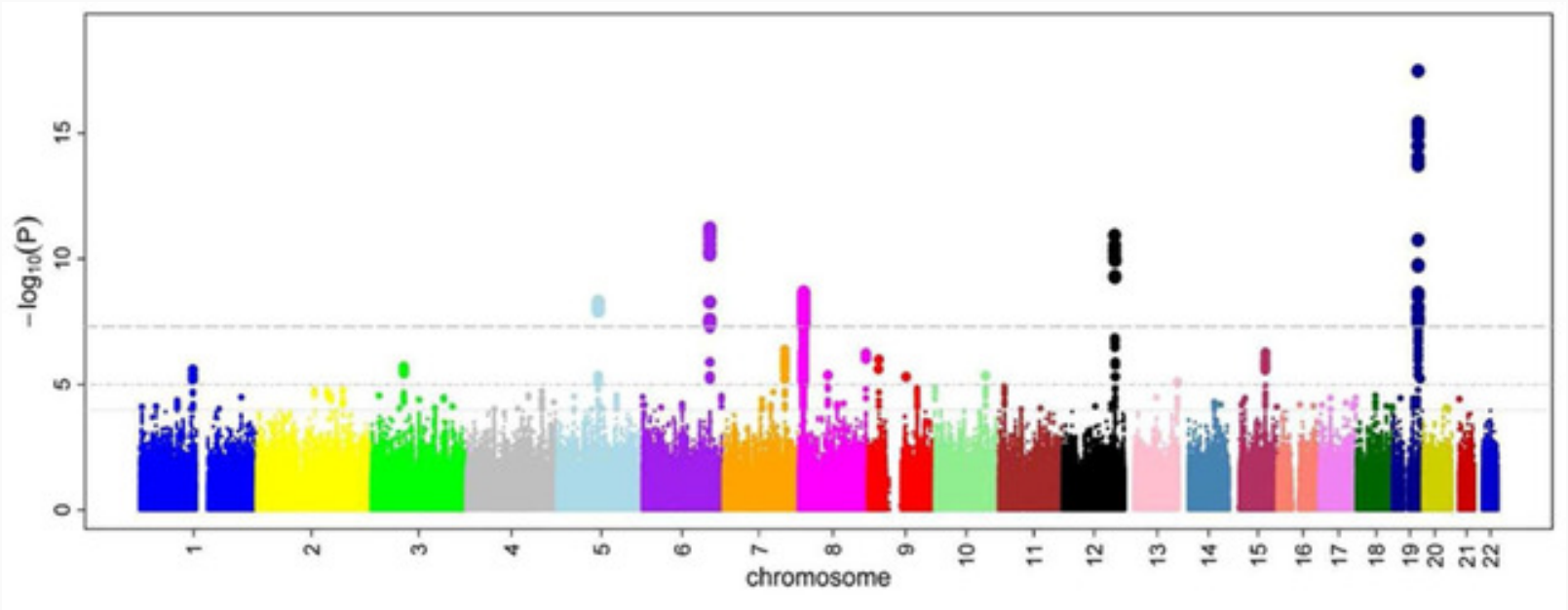
- Detección de anomalías y novedades.
- Encuentra puntos en los datos (también series temporales) que parecen diferentes al resto.
- Útil en muchos campos, e.g., mantenimiento predictivo, *forecasting*.

Ejemplos de aprendizaje no supervisado (V)



- Autoencoder.
- Encuentra codificación eficiente de datos.
- Generar datos artificiales.
- Útil para reducción de la dimensionalidad, búsqueda de características, modelos generativos y detección de valores atípicos entre otros usos.

Ejemplos de aprendizaje no supervisado (VI)



- Aprendizaje de reglas de asociación.
- Descubrir relaciones no aparentes entre variables.
- Relacionado con sequence mining, aunque éste tiene en cuenta el orden temporal, mientras que el aprendizaje de reglas de asociación.

Clustering

K-means

Supondremos un conjunto de datos que consiste en N instancias/observaciones $\{x_1, x_2, \ldots, x_N\}$ de dimensionalidad D.

Los algoritmos de *clustering* tienen como objetivo identificar grupos (*clusters*) de puntos de datos "similares". K-means es un algoritmo de agrupación cuya idea intuitiva es agrupar puntos de datos "cercanos", mientras se separan los puntos "lejanos". El algoritmo asume un número conocido de *clusters* K (una forma común de seleccionar K es el método de codo que se trata en la diapositiva 17).

Denotaremos $\boldsymbol{\mu}_k$ (D-dimensional) al punto promedio del *cluster* k-ésimo (*centroid*), para $k{=}1, 2, \ldots, K$.

K-means (II)

Sean $\boldsymbol{r}_{nk}$ variables binarias con valor 1 si el punto de datos $\boldsymbol{x}_n$ se asigna al *cluster* k, o 0, de lo contrario, para $n{=}1, \ldots, N$ y $k{=}1, \ldots, K$.

Para caracterizar cómo de buena es una asignación de *clusters*, se define una *medida de distorsión*:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} \boldsymbol{r}_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \tag{1}$$

El objetivo es encontrar los valores $\{\boldsymbol{r}_{nk}\}$ y $\{\boldsymbol{\mu}_k\}$ minimizando el error J.

K-means (III)

J se puede minimizar realizando repetidamente dos fases:

1. Seleccionar la asignación óptima a los *clusters* $\{\boldsymbol{r}_{nk}\}$ asumiendo centroides $\{\boldsymbol{\mu}_k\}$ fijos. Es decir, para cada punto $\boldsymbol{x}_n$ se elige el *cluster* con el centroide más cercano:

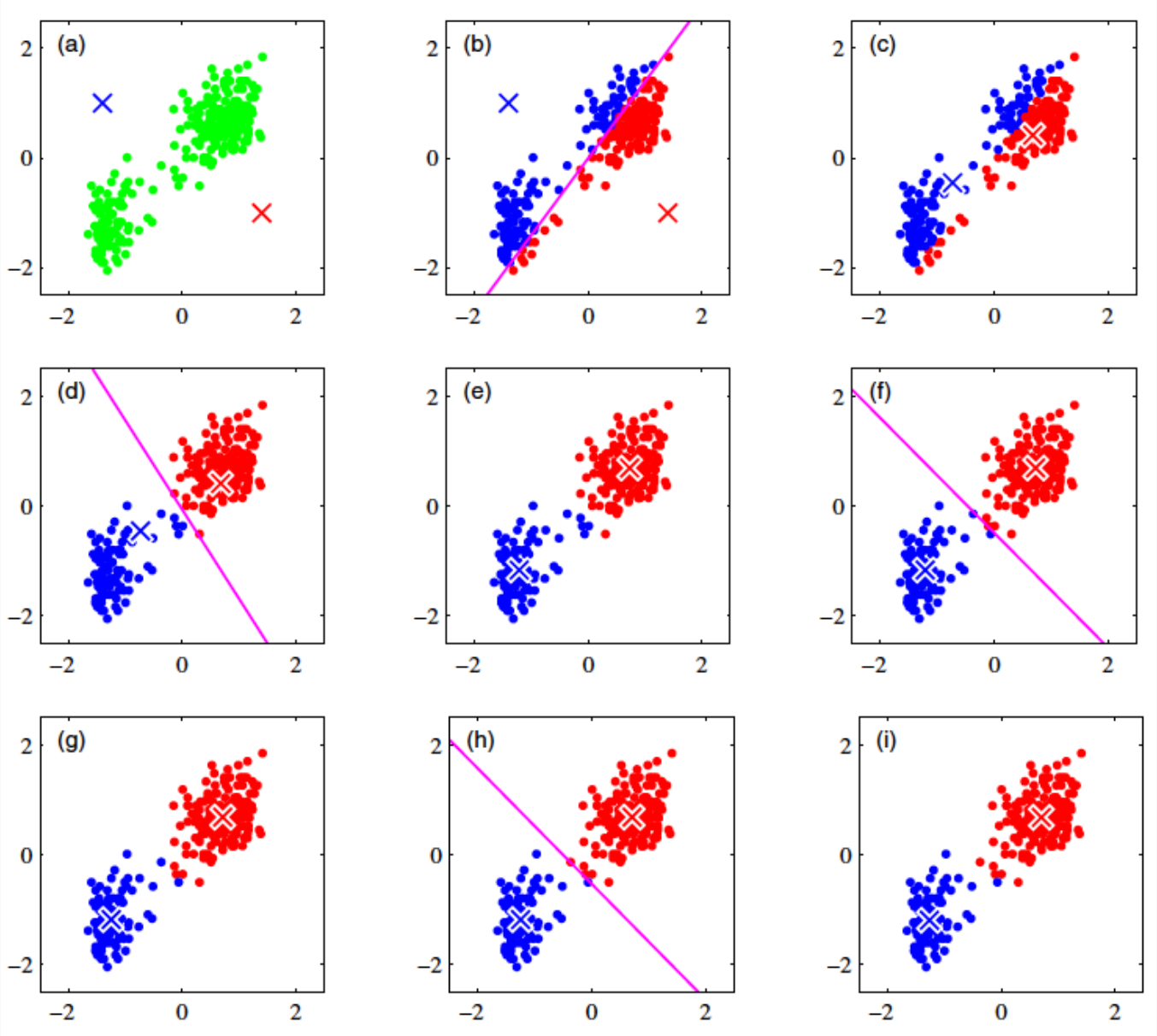
$$\boldsymbol{r}_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

2. Seleccionar los centroides óptimos $\{\boldsymbol{\mu}_k\}$ asumiendo asignaciones $\{\boldsymbol{r}_{nk}\}$ fijas. Dicha asignación se puede calcular igualando la derivada de J con respecto a $\boldsymbol{\mu}_k$ 0:

$$2\sum_{n=1}^{N} \boldsymbol{r}_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0 \implies \boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \boldsymbol{r}_{nk}\boldsymbol{x}_n}{\sum_{n=1}^{N} \boldsymbol{r}_{nk}} \tag{3}$$

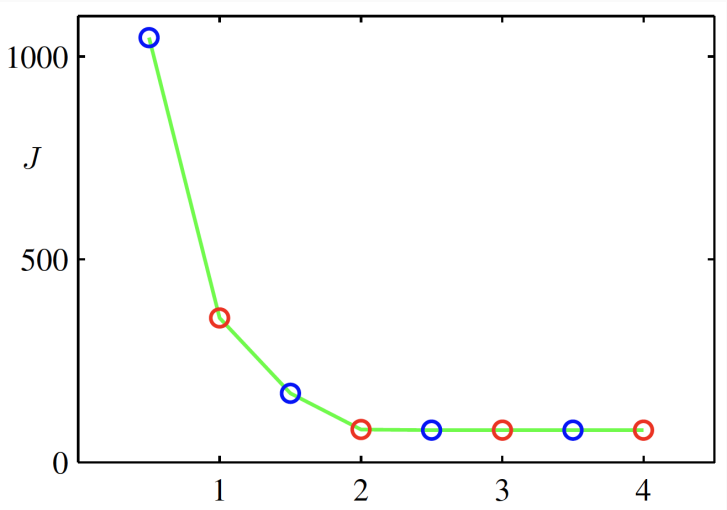
   Eso es, $\boldsymbol{\mu}_k$ es el *promedio* de los puntos asignados al *cluster* k-ésimo.

K-means ejemplo para $K=2$



K-means ejemplo para K=2 (II)

El primer paso se llama **expectation** (E) y el segundo **maximization** (M). Esta terminología se justificará en la siguiente sección. Con este algoritmo J converge ya que en cada paso el error disminuye, como se puede ver en el ejemplo a continuación.



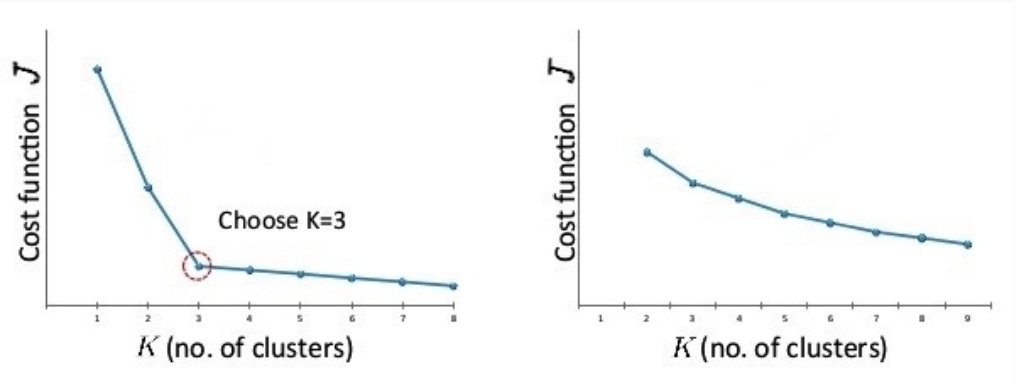
- Los pasos E de están marcados en azul y los pasos M en rojo
- El algoritmo converge después del 3er paso M

Iobservaciones de implementación

- Los centroides iniciales $\{\boldsymbol{\mu}_k\}$ se eligen al azar.
- El algoritmo se ejecuta hasta que no ocurran más cambios, y no se logre ninguna mejora en J.
- La solución es un óptimo (posiblemente *local*).
- Usualmente, el algoritmo se ejecuta varias veces con diferentes centroides iniciales, y la mejor solución es seleccionada.
- Dado que se utiliza la distancia euclídea debe hacerse una **estandarización** de los datos, (aplicando una transformación lineal para obtener media 0 y varianza 1) para evitar asimetrías entre las diferentes variables de entrada.

Notas de implementación (II)

- Puede emplearse con otras distancias. En este caso, el algoritmo se llama *K-methoids*.
- Un método frecuente para determinar K es el criterio del codo (*elbow*), que selecciona el valor de K a partir del que el error tiene sólo mejoras marginales:



Modelo de mezcla de Gaussianas

K-means se limita a grupos de "forma esférica". Para superar esta restricción, el modelo de mezcla de Gaussianas (Gaussian Mixture Model - GMM) permite que los *clusters* se conformen como Gaussianas D-dimensionales arbitrarias $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Una GMM es una superposición de K-Gaussianas D-dimensionales, y tiene por distribución:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4}$$

donde los $\{\pi_k\}$ son no negativos y satisfacen $\sum_{k=1}^{K} \pi_k = 1$.

Modelo de mezcla de Gaussianas (II)

El modelo GMM también se puede entender asumiendo una variable latente (oculta) z, cuyo valor $k \in \{1, \ldots, K\}$ selecciona la Gaussiana k-ésima, de la cual se muestrea la variable $\boldsymbol{x}$.

Dado un conjunto de N datos de entrenamiento $\{\boldsymbol{x}_n\}$, el objetivo es calcular los parámetros $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ que maximicen su (log)-verosimilitud:

$$\sum_{n=1}^{N} ln(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) \tag{5}$$

Para encontrar soluciones máximo-verosímiles para el GMM usaremos el algoritmo *expectation-maximization (EM)*.

Algoritmo EM

Como K-means, EM asume un valor de K fijo, y procede secuencialmente realizando los siguientes pasos:

1. Establecer parámetros iniciales $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$.
2. (**Expectation**). Con los parámetros actuales fijos, evaluar las **responsabilidades** $\{\gamma_{nk}\}$ (la probabilidad de que el *cluster* k haya generado el punto de datos n):

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \Sigma_j)} \tag{6}$$

   El nombre de este paso proviene del hecho de que las responsabilidades pueden ser interpretadas como las *esperanzas* de variables aleatorias $\{z_{nk}\}$, que asignan 1 al punto n si pertenece al *cluster* k, o 0 de lo contrario.

Algoritmo EM (II)

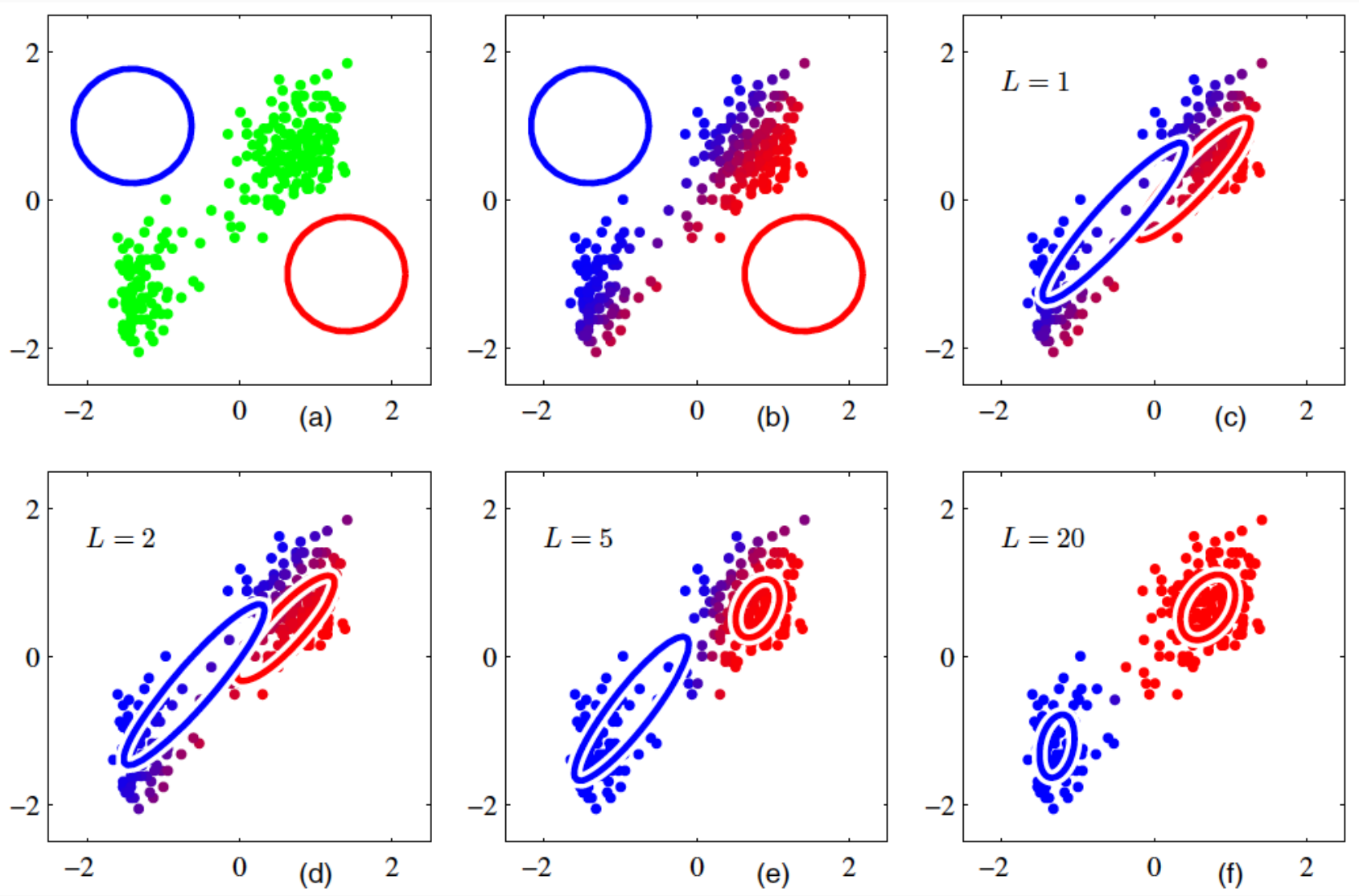
3. (**Maximization**). Fijando $N_k = \sum_{n=1}^{N} \gamma_{nk}$ se estiman los parámetros:

$$\begin{aligned} \boldsymbol{\mu}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x}_n \\ \boldsymbol{\Sigma}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{\text{new}})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{\text{new}})^T \\ \pi_k^{\text{new}} &= \frac{N_k}{N} \end{aligned} \tag{7}$$

   Este paso elige los parámetros que *maximizan la probabilidad de observación del conjunto de datos*, de ahí su nombre.

4. Calcular la log-verosimilitud con la ec. (5). Si ésta o los parámetros han cambiado ir al paso 2.

GMM ejemplo para $K=2$



Notas de implementación (III)

- Se pueden utilizar las mismas ideas de implementación que en K-means.
- Debido a similitudes de operación, los pasos en K-means se llaman también expectation-maximization.
- El algoritmo EM se puede aplicar a cualquier otro modelo basado en variables latentes.
- En contraste con un modelo de una sola Gaussiana, EM puede encontrar singularidades en el cálculo. Se emplean heurísticos para solucionar tales situaciones.

Detección de anomalías

Outlier detection

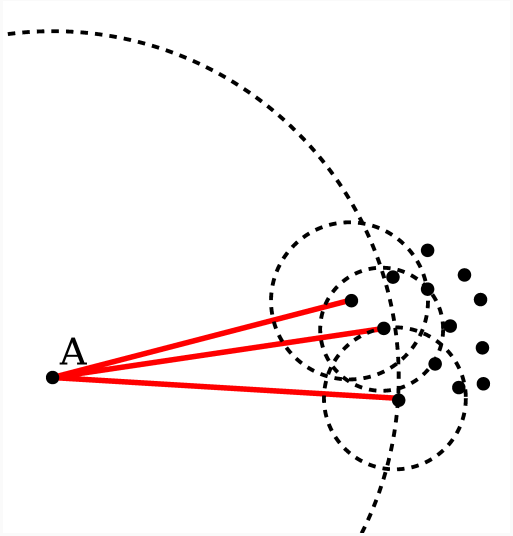
Muchas veces es posible recopilar datos provenientes de fuentes naturales o artificiales. Por ejemplo, de sensores instalados en motores, de gráficos financieros, secuencias de alelos en el ADN, imágenes de personas o su voz, etc.

En tales situaciones, la **detección de anomalías** se preocupa por determinar qué datos tienen diferencias significativas con el resto.

Existen métodos *locales* (que consideran la densidad de vecinos) o *globales* (que forman un modelo de los datos y luego analizan qué datos encajan menos en ese modelo). En las siguientes secciones, se presenta un ejemplo de cada tipo.

Local Outlier Factor

El *local outlier factor* (LOF) es un método *de densidad local*, parecido al K-NN, pero trabajando con datos no etiquetados. Cuando se detectan zonas con una densidad de puntos menor que la de sus vecinos, ésta se considera *anómala*.

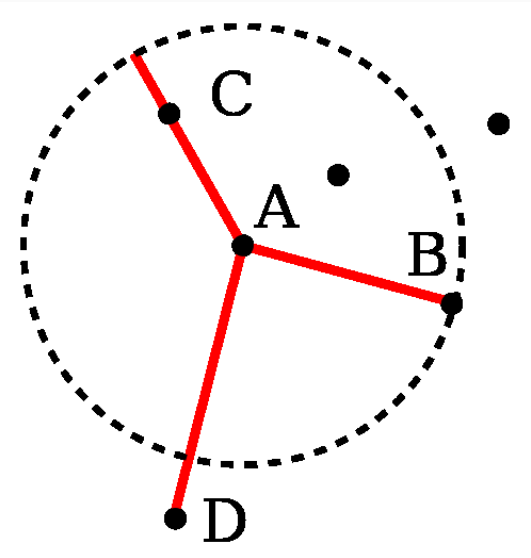


El punto A se considera un valor atípico ya que tiene una densidad mucho menor que sus vecinos.

Local outlier factor (II)

Se define la k-distance($\boldsymbol{x}$) como la distancia de un punto $\boldsymbol{x}$ a su k-ésimo vecino más cercano, y $N_K(\boldsymbol{x})$ como el conjunto de vecinos a una distancia menor o igual que la K-distance($\boldsymbol{x}$):

Por otro lado, se llama *reachability* entre dos puntos $\boldsymbol{x}, \boldsymbol{x}'$ a r_K=máx$\{K$-distance$(\boldsymbol{x}), d(\boldsymbol{x}, \boldsymbol{x}')\}$. La *reachability* fuerza todos los puntos de un "cluster local" a la misma "distancia", tal como sucede con los puntos B y C en la figura.



Local outlier factor (III)

La densidad local de $\boldsymbol{x}$ se define como:

$$\rho_K(\boldsymbol{x}) = \left(\frac{\sum_{\boldsymbol{x}' \in N_K(\boldsymbol{x})} r_K(\boldsymbol{x}, \boldsymbol{x}')}{|N_K(\boldsymbol{x})|} \right)^{-1} \tag{8}$$

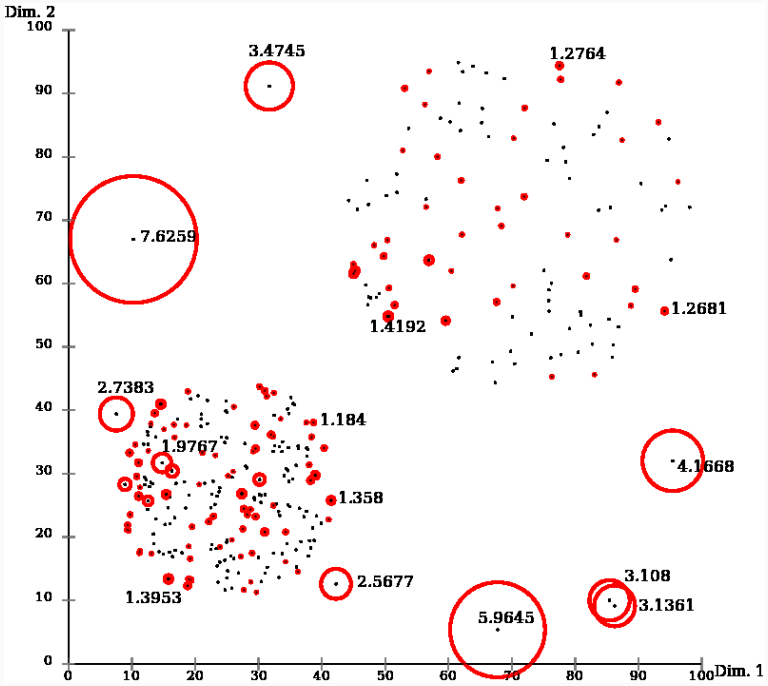
y representa el inverso de la distancia a $\boldsymbol{x}$ **desde** sus vecinos.

Finalmente, la densidad del K-vecindario de $\boldsymbol{x}$ es:

$$\mathrm{LOF}_K(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{x}' \in N_K(\boldsymbol{x})} \rho_K(\boldsymbol{x}')}{|N_K(\boldsymbol{x})| \rho_K(\boldsymbol{x})} \tag{9}$$

Un **LOF mayor que un $\mathrm{LOF}_{\mathrm{critical}}$ identifica un valor atípico**.

Local outlier factor (IV)



Este ejemplo muestra puntos con LOF > 1. Como ventaja sobre los modelos globales, LOF puede identificar puntos, que, en su vecindad, constituyen valores atípicos. La relación crítica puede ser difícil de determinar. En algunos contextos puede estar cerca de 1, mientras que en otros ser incluso mayor que 2.

Métodos globales

La idea central de estos métodos es determinar cómo de inesperado es el punto $\boldsymbol{x}$ dado un modelo entrenado con un conjunto de datos $\{\boldsymbol{x}_n\}$.

Un enfoque natural es utilizar el conjunto de datos para **ajustar una variable gaussiana**, y luego usar su distribución $p(\boldsymbol{x})=\mathcal{N}(x|\boldsymbol{\mu},\boldsymbol{\Sigma})$ como medida de la "atipicidad". Los estimadores insesgados para dicho ajuste son:

$$\begin{aligned} \widehat{\boldsymbol{\mu}} &= \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n \\ \widehat{\boldsymbol{\Sigma}} &= \frac{1}{N-1}\sum_{n=1}^{N}(\boldsymbol{x}_n-\widehat{\boldsymbol{\mu}})(\boldsymbol{x}_n-\widehat{\boldsymbol{\mu}})^T \end{aligned} \tag{10}$$

Métodos globales (II)

Para una gaussiana multi-variante, su densidad es:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = (2\pi)^{-D/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}$$

Por lo tanto, el logaritmo de la distribución es proporcional a $\ln p(\boldsymbol{x}) \propto -\frac{1}{2}d^2_{\mathcal{M}}$, donde:

$$d_{\mathcal{M}} = [(\boldsymbol{x}-\widehat{\boldsymbol{\mu}})^T\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}})]^{1/2} \tag{11}$$

$d_{\mathcal{M}}$ se conoce como la *distancia de Mahalanobis*, y se usa comúnmente como una medida de la atipicidad de un punto en comparación con una distribución de probabilidad. Su nivel crítico se puede establecer mediante métodos heurísticos.