



No supervisado

Aprendizaje Automático

Víctor de la Cueva

vcueva@itesm.mx

Aprendizaje no supervisado

- El aprendizaje no supervisado es un tipo de aprendizaje **inductivo**, es decir, el sistema se entrena mediante ejemplos.
- Sin embargo, los ejemplos **no tienen etiquetas**, esto es, no tienen **salida** deseada.
- Por esta razón, la actividad más importante de un sistema de aprendizaje no supervisado es la **descripción** de los datos.

Clustering

- Una de las formas principales para hacer descripción de los datos es tratar de **formar grupos** de ellos que tengan características comunes.
- A estas técnicas se les conoce como **agrupamiento** o **clustering**.
- Los métodos se pueden agrupar en dos grandes grupos:
 - **Jerárquicos**: establecen una jerarquía entre *clusters* dando una sucesión de particiones donde cada partición se obtiene uniendo o dividiendo *clusters*.
 - **No jerárquicos (particionales)**: obtienen una única partición de los datos mediante la **optimización de alguna función** adecuada.

Aplicaciones del agrupamiento

- Segmentación de mercado
- Análisis de redes sociales
- Organización de *clusters* computacionales
- Análisis de datos astronómicos
- Big Data
- Data Mining

No supervisado



K-MEANS

Métodos particionales

- Los métodos particionales dan lugar a una partición mediante la **minimización** o **maximización** de algún **criterio**.
- Estos métodos requieren que el usuario fije el **número de clusters k** que tendrá la partición.
 - Conocer el verdadero número de *clusters* que existen es un problema particular no trivial.
 - Hay diferentes **técnicas** que pueden ayudar a **estimar** este número.
- **K-means** es uno de los métodos particionales más utilizados.

Algoritmo General

1. De entre los m casos elegir k que llamaremos semillas y denotaremos $c_j, j = 1, \dots, k$. Cada semilla c_j representará el *cluster* C_j .
2. Asignar el caso i al cluster C_j cuando

$$d(x_i, c_j) = \min_{l=1, \dots, k} d(x_i, c_l)$$
 es decir, a la semilla que tiene más cerca.
3. Calcular la mejora que se produciría en el criterio elegido al asignar un caso a otro *cluster* en el que no está actualmente.
4. Hacer el cambio que mayor mejora produce en el criterio.
5. Repetir los pasos 3 y 4 hasta que ningún cambio mejore el criterio elegido.

Comentarios

- Estos pasos forman la idea básica del algoritmo *k-means* pero existen muchas versiones del mismo.
- El algoritmo dependen totalmente del criterio de optimización utilizado.
- El algoritmo a veces no converge, lo cual se puede deber:
 - Se seleccionó mal el número k
 - No hay estructura de *clusters* en los datos
- El algoritmo, cuando converge sólo garantiza un *óptimo local*.

Una variante

- Una de las variantes más usadas del algoritmo k-means consiste en **cambiar** en cada paso los **centroides** c_j de los clusters (a los que llamamos semillas), y hacer la reasignación de los datos.
- La forma en la que se cambian los centroides es calcularlos nuevamente como la **media** de los datos actuales que pertenecen al cluster.

Algoritmo K-means

- Seleccionar aleatoriamente K de los m datos como centroides $c_1, c_2, \dots, c_K \in \mathbb{R}^n$ de los K clusters, donde n es el número de dimensiones de los datos y m es el número de datos.
- Repetir hasta que no haya más cambios
 - Asignar cada dato a un *cluster*
 - for $i=1$ to m
 $c^{(i)}$ = índice (de 1 a K) del centroide más cercano a $x^{(i)}$
 - for $k = 1$ to K
 c_k = media de los puntos asignados al cluster k

Función objetivo

- La **función objetivo** (criterio de optimización) para *K-means* es la distancia promedio de cada dato a su centroide:

$c^{(i)}$ = índice del *cluster* (1,...,K) al que el ejemplo $x^{(i)}$ fue asignado.

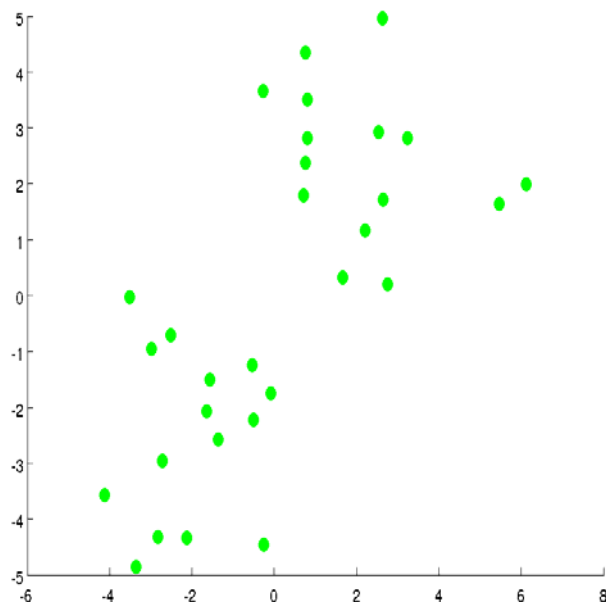
c_k = centroide del *cluster* k ($c_k \in \mathbb{R}^n$)

$c_{c(i)}$ = centroide del *cluster* al que fue asignado el ejemplo $x^{(i)}$

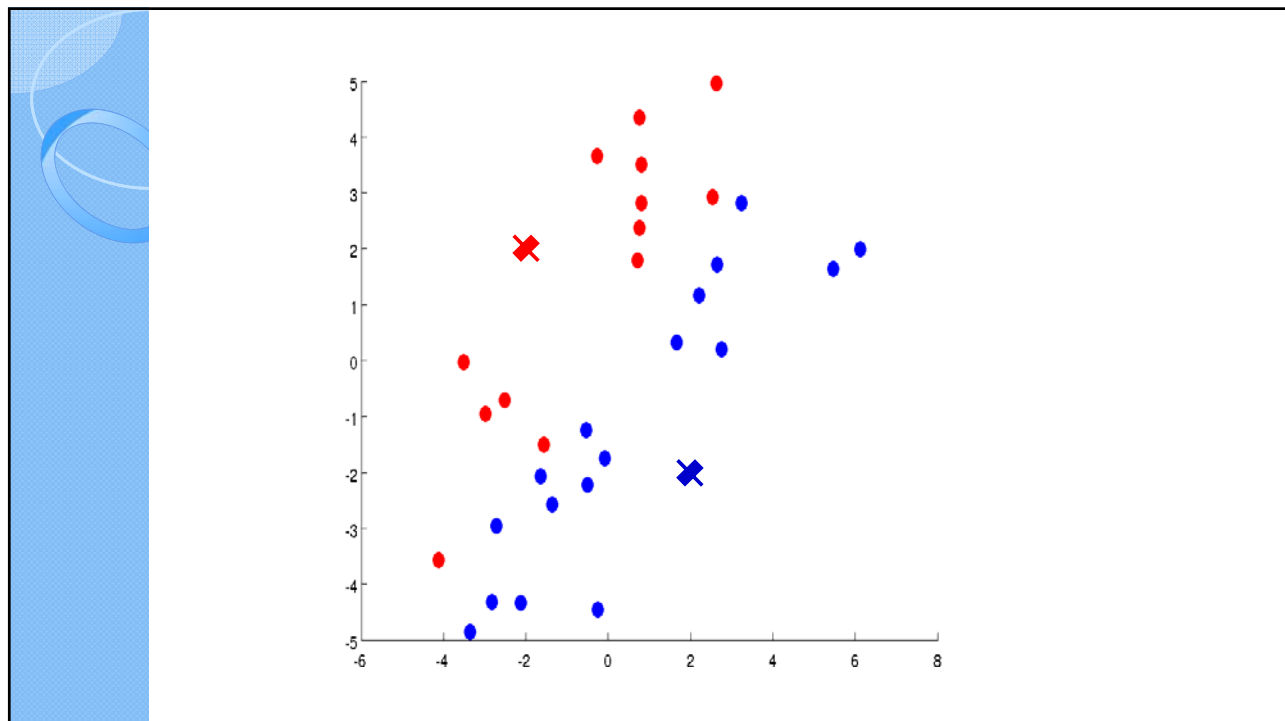
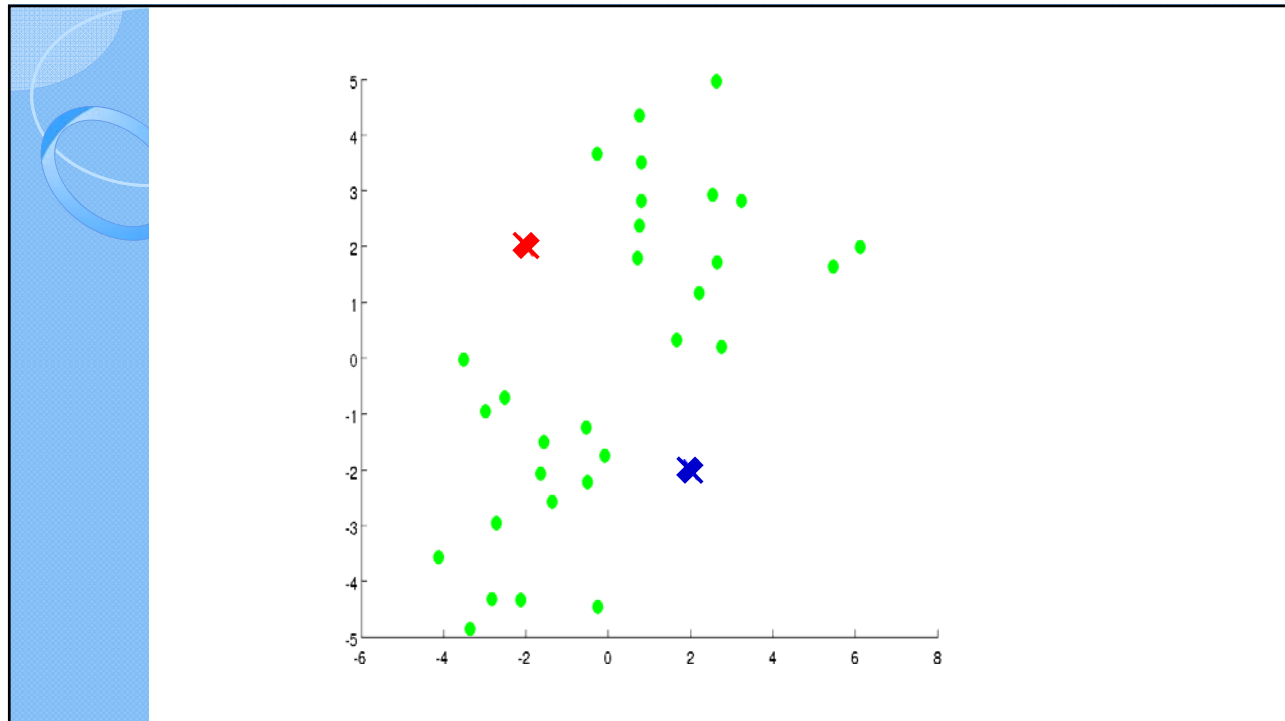
Función objetivo:

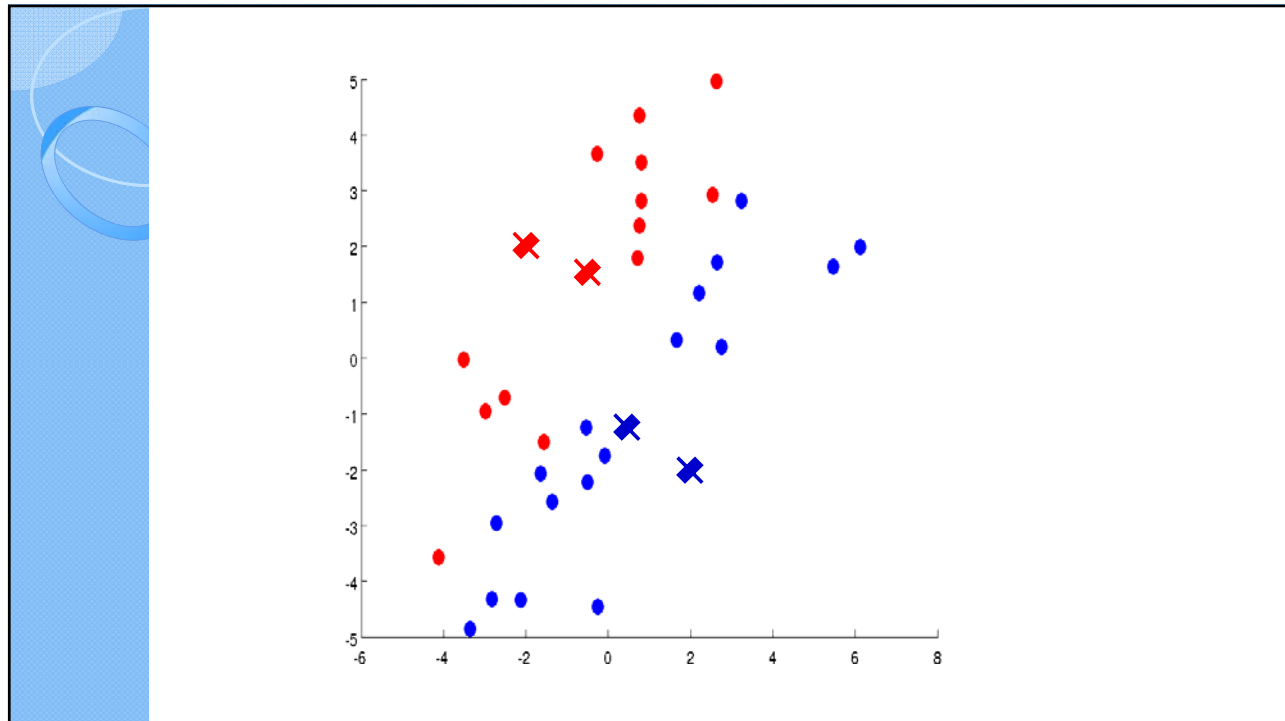
$$J(c^{(1)}, \dots, c^{(m)}, c_1, \dots, c_K) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - c_{c(i)}\|^2$$

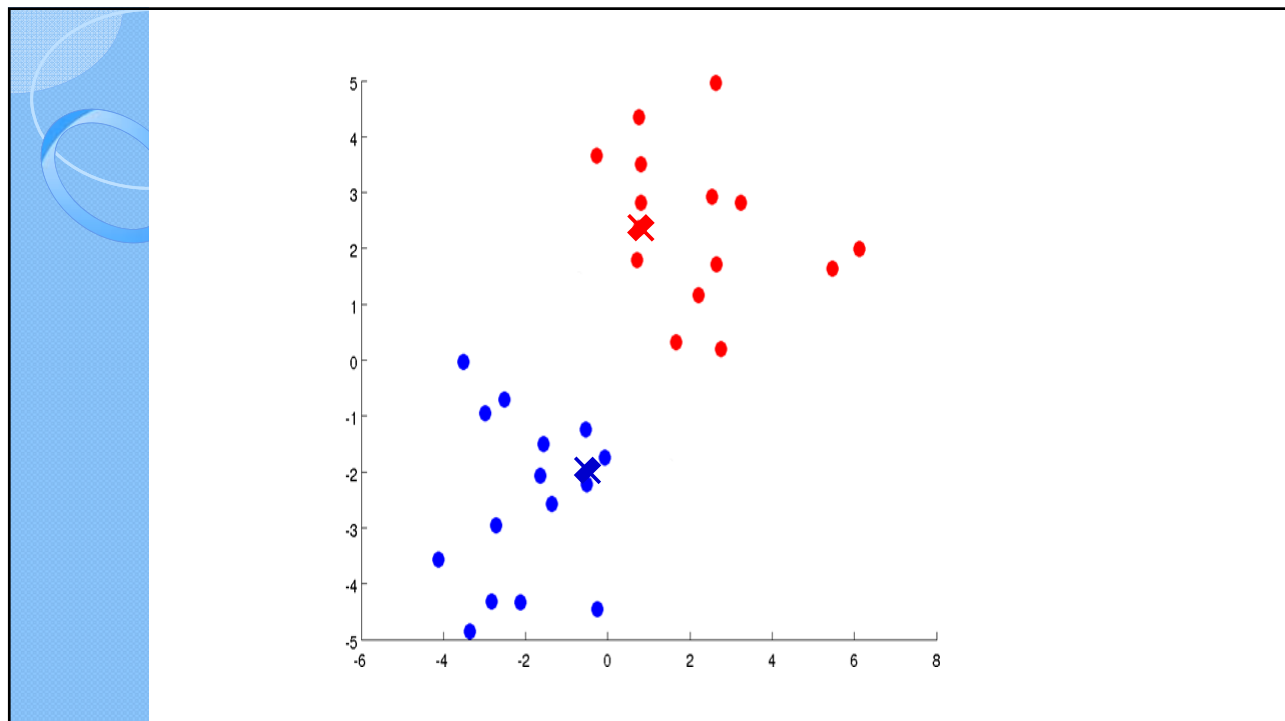
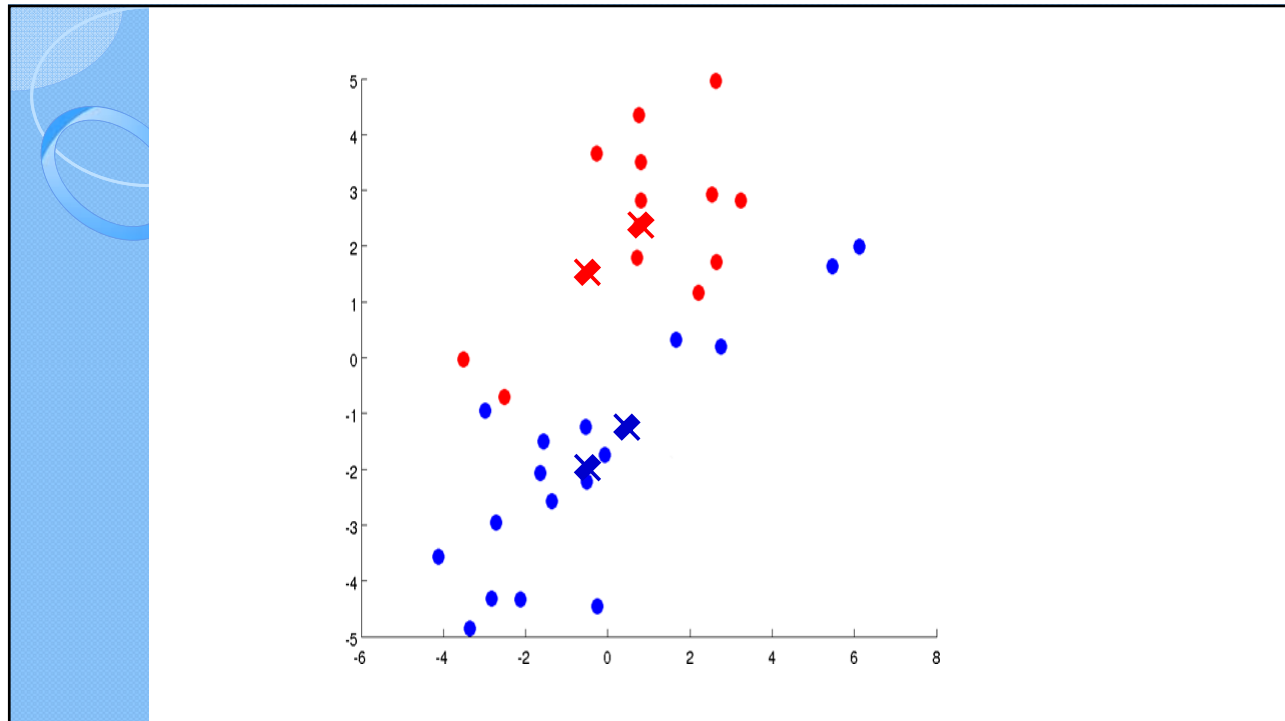
$$\min_{\substack{c^{(1)}, \dots, c^{(m)} \\ c_1, \dots, c_K}} J(c^{(1)}, \dots, c^{(m)}, c_1, \dots, c_K)$$

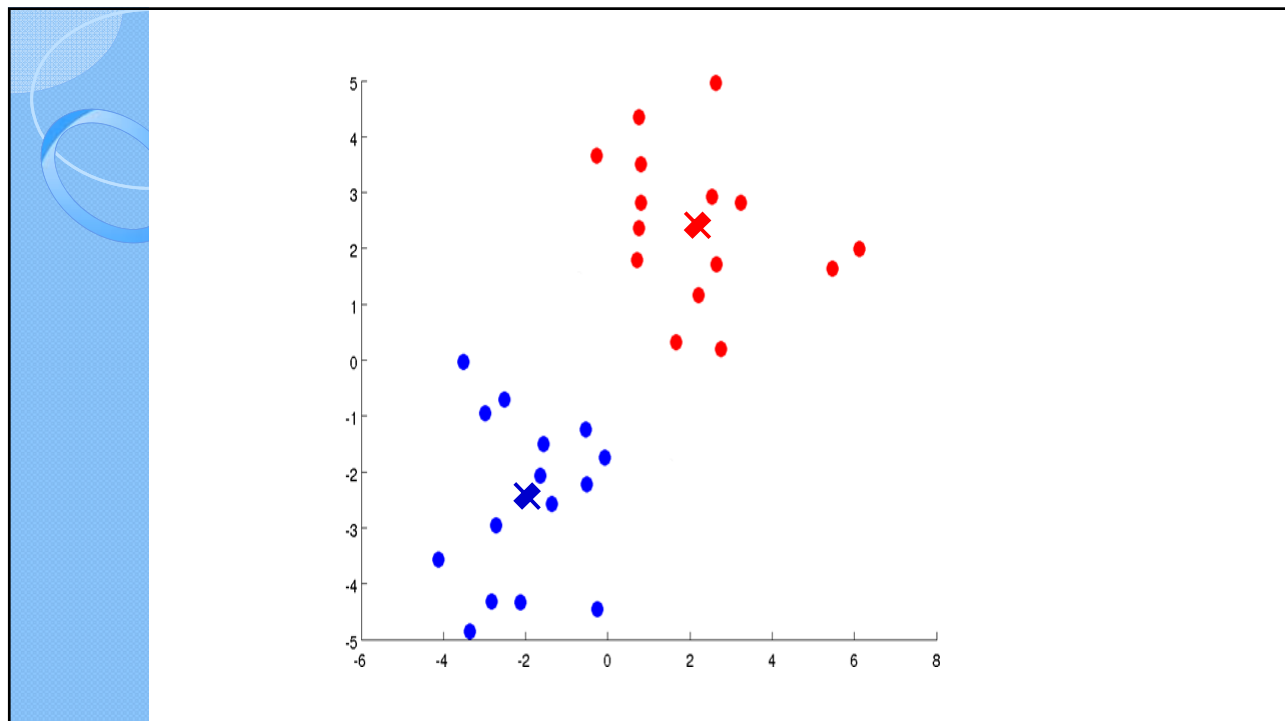
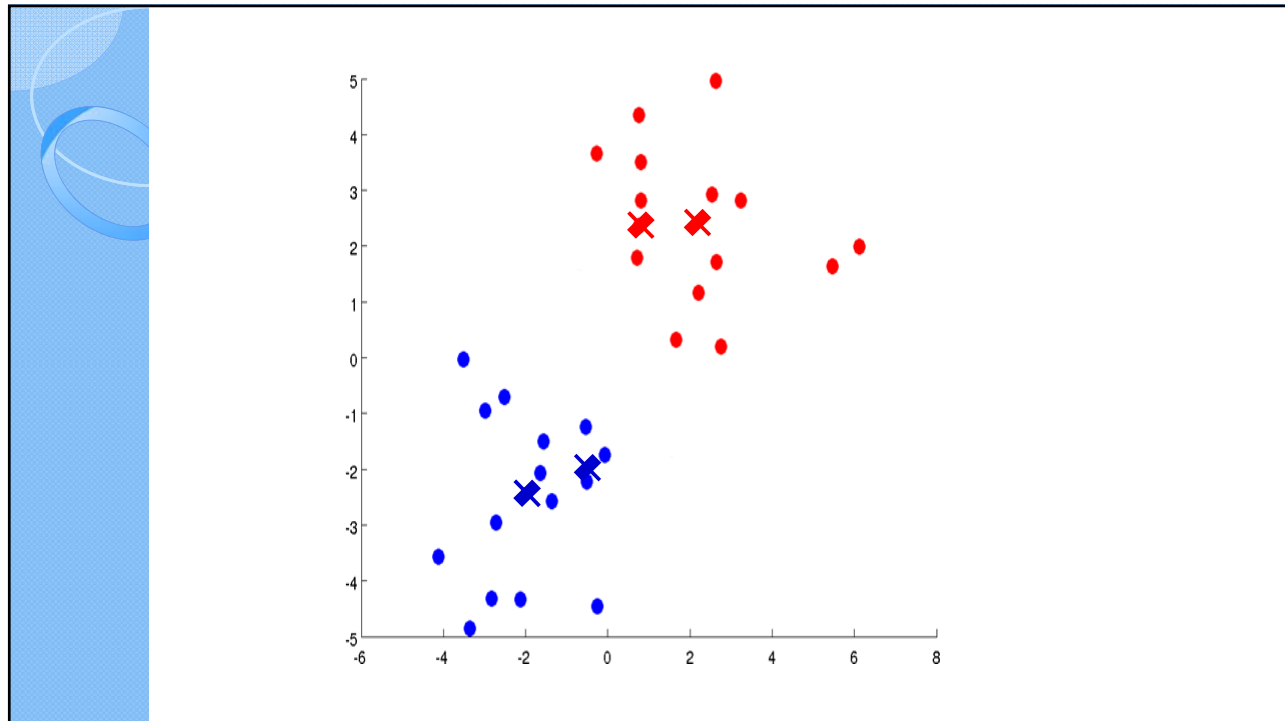


Tomado Coursera de ML Prof. Ng



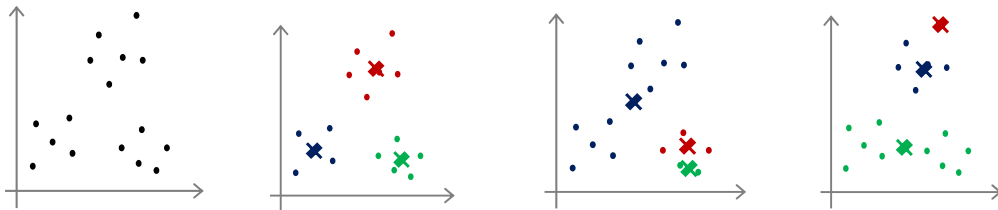






Óptimos Locales

- Debemos recordar al K-means cuando converge sólo garantiza llegar a óptimos locales.



Evitando óptimos locales

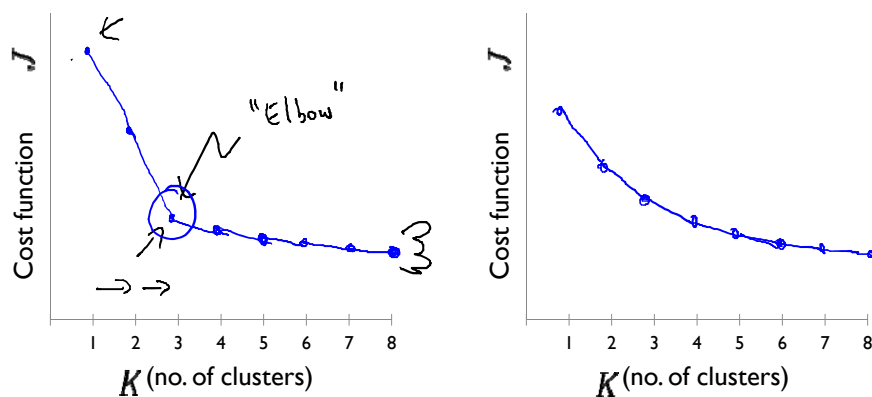
- Debido a que K-means llega a un óptimo local una forma de tratar de llegar al óptimo global es **correrlo varias veces**.
- Algoritmo:
 - For $i=1$ to 100 {
 - Correr *K-means*
 - Calcular la función de costo J (distorsión)
 - }
 - Seleccionar el *clustering* que tenga menor costo

Número de clusters

- Una de las cuestiones más complicadas en el método de K-means es la selección del **número óptimo de clusters**.
- Hay varias opciones pero una de las más usadas en el “método del codo” (*elbow method*):
 - Correr K-means para diferentes valores de k
 - Calcular J
 - Graficar los valores obtenidos y seleccionar el del “codo”

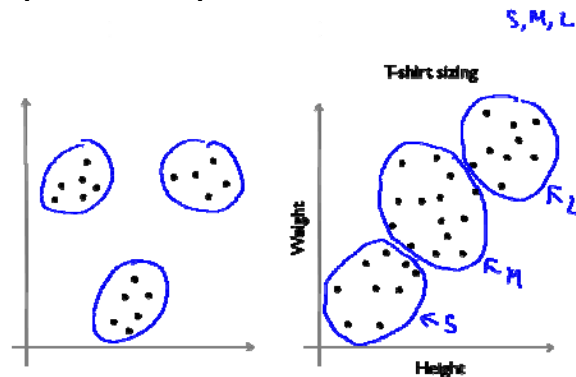
Seleccionando el valor de K

Método del Codo (Elbow method):



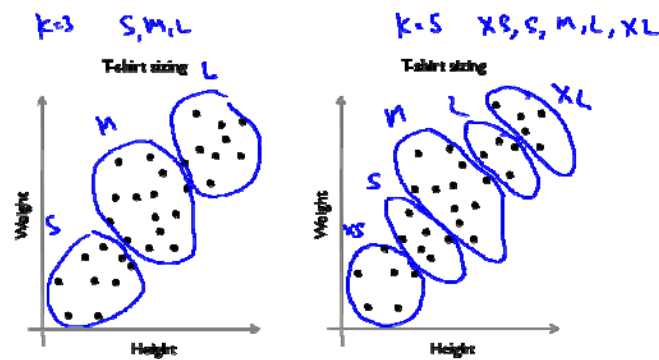
K-means para clusters no separados

- En algunas ocasiones los *clusters* no están completamente separados por lo que será necesario una métrica adecuada para su separación.



K-means para un propósito específico

- En algunas ocasiones se corre K-means para un propósito específico posterior.
- En ese caso, la métrica tiene que ser para verificar qué tanto los clusters cumplen ese propósito específico.



Referencias

- B. Sierra. Aprendizaje Automático conceptos básicos y avanzados. Pearson (2006).
- Coursera (<https://www.coursera.org/>)
 - Curso de Machine Learning: Prof. Andrew Ng (2011)
<https://www.coursera.org/course/ml>