Team #3

2/OCT/2018

K-means

desarrollo colaborativo

# Introducción

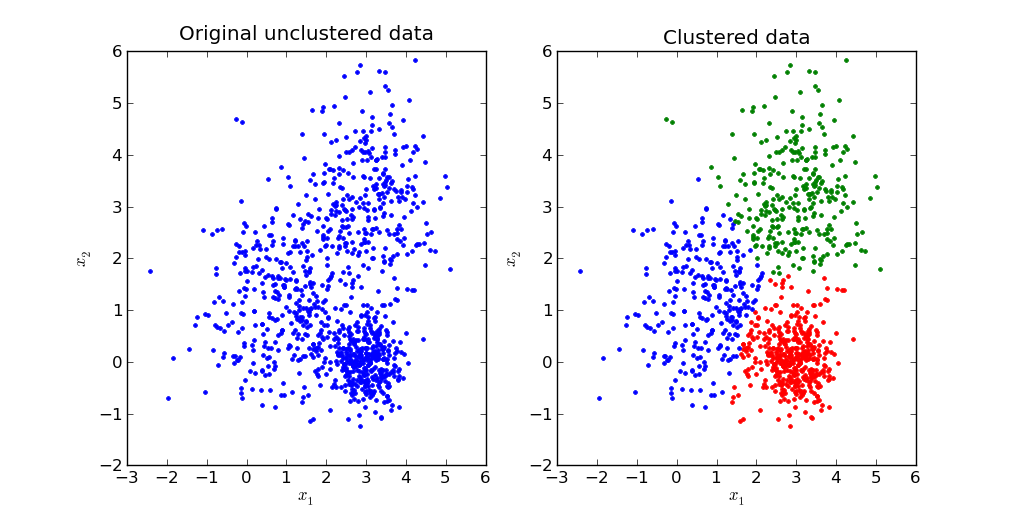
La clasificación automática de objetos o datos es uno de los objetivos del aprendizaje de la maquina para poder asignar grupos por medio discriminaciones de diferentes tipos, la página unioviedo.es considera que existen 3 tipos de clasificación para el aprendizaje automático.

* Clasificación supervisada: disponemos de un conjunto de datos, ya sea alguna imagen, un conjunto de datos del mismo rango que tengan en común una etiqueta. Utilizando esa etiquetase construye un modelo y en base a ese modelo es como se va a clasificando.
* Clasificación no supervisada: clasificación en la que los datos no tienen etiquetas y estos se clasifican a partir de su estructura interna.
* Clasificación semi-supervisada: algunos datos tienen etiquetas, pero no todos. Son muy comunes en la clasificación de imágenes.

# K-means

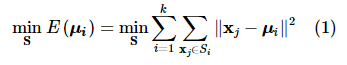
K-means es un algoritmo de clasificación no supervisada (clasificación de los datos sin etiquetas, pero con características iguales o parecidas) que agrupa objetos en *k* grupos. El algoritmo se resuelve de la siguiente manera:

1. Se inicializan los grupos, es decir se escoge el numero de grupos, k, se establecen k centroides para cada grupo en el espacio de los datos.
2. Se asignan los objetos a su grupo (Al centroide al que más se acerque).
3. Se mueve la posición del centroide, conforme a las características del nuevo centriolo.
4. Se repiten los dos pasaos anteriores con todos los demás objetos o hasta que haya un umbral en el que no cambie mucho las posiciones.



Como se puede observar en esta imagen tenemos la “clusterización” de los puntos en 3 grupos *k*.

Los objetos se representan con vectores reales de dd dimensiones (x1,x2,…,xn)(x1,x2,…,xn) y el algoritmo k-means construye kk grupos donde se minimiza la suma de distancias de los objetos, dentro de cada grupo **S={S1,S2,…,Sk}S={S1,S2,…,Sk}**, a su centroide. El problema se puede formular de la siguiente forma:





El término “k-means” fue utilizado por primera ves por James McQueen 3n 1967, pero principalmente fue propuesta por Hugo Steinhaus e 1957.

**Algoritmo estándar**

Este utiliza la técnica de refinamiento iterativo, también se le conoce como el algoritmo de Lloyd. El algoritmo se considera que ha convergido cuando las asignaciones ya no cambian.

# Font End

**Technologies:**

* **React.js**
* **Redux**
* **Axios**

**Components:**

* MainView
* DataBase
* Graphic Component

# Back End

In backend we have the resolution of the different algorithms to serve the resolution to React.js client.

**Technologies:**

* Node.js
* Express

**Routes:**

/kmeans

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Route | Parameters | Success |
| POST | /api/kmeans/iris/  Returns the k means resolve from iris data set | Number of centroids as int inside a json object.  { k: <int> } | Array of objects where each object contains x and y coordinate and group id.  Example:  [ { x:<float>, y: <float>, group: <int> }] |

/competitivelearning

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Route | Parameters | Success |
| POST | /api/competitivelearning/iris/  Returns the competitive learning resolve from iris data set | Number of centroids as int inside a json object.  { k: <int> } | Array of objects where each object contains x and y coordinate and group id.  Example:  [ { x:<float>, y: <float>, group: <int> }] |

# **Helpers:**

* Document.helper.js: this helper gets the documents from our data sets located in resources directory.
* Kmeans.helper.js: This helper resolves the k means algorithm for now it just resolves iris data set.
* Competitive.helper.js: This helper resolves competitive learning algorithm for now it just resolves iris data set.

**Components**

