Universidad Simón Bolívar
Departamento de computación
Inteligencia aritifical CI5437

Proyecto 3
Clustering

Profesora :
Carolina Martínez

Grupo:
Carlos Farinha - 09-10270
Javier López - 11-10552
Nabil Márquez - 11-10683

16 de marzo del 2017

Introducción

K-means es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano. Es un método muy utilizado en minería de datos debido a su capacidad de ubicar patrones dentro de un espacio definido de observaciones.

En el siguiente informe se presenta una aplicación del algoritmo k-means y su representacion grafica en el conjunto de datos "Iris Data Set" el cual cataloga 3 categorizaciones de una planta de Iris, el objetivo es que el algoritmo pueda generar la definición de a que conjunto pertenece una observacion en base a longuitud y ancho del sepalò y petalo de la flor, para categorizar a que especie pertenece.

Implementación

Librerias y Funciones Externas Usadas

- *PIL - Image* : Generador de Imagenes jpg
- *Math - random* : Generador de valores pseudo-aleatorios
- *os* : Creador de directorios para el guardado de los graficos generados
- *numpy* : Control de estructuras en el algoritmo k-means
- *matplotlib - pyplot* : Generador de Scatterplots de los conjuntos

Algoritmo K-means

Para la implementacion del algoritmo k-means se creo el modulo *k-means.py* el cual contiene las siguientes funciones definidas:

- *puntos_del_cluster*: Clasifica los puntos segun el centroide más cercano a la observacion
- *centrar*: Calcula el punto medio de las observaciones clasificadas
- *iteracion*: Realiza una iteracion de la clasificacion de los puntos siempre que el cambio en el centroide varie.

Programa principal

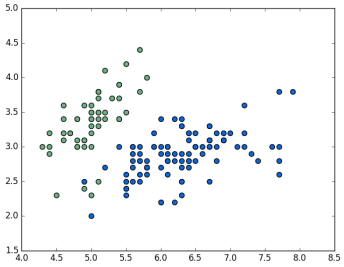
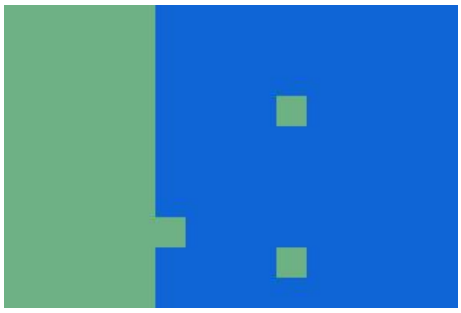
El archivo principal esta ubicado como *main.py* y el cual lee los datos del archivo principal *bezdekIris.data* realiza una iteracion continua centrando los datos y definiendo los nuevos centroides del conjunto en base a un k definido en la funcion, las iteraciones continuan formalmente hasta lograr una convergencia, una vez realizada la convergencia se comprime la informacion en un Scatter plot donde se toman 2 de las variables para presentar la información (longuitud y ancho del sepalo) ademas genera una imagen comprimida de los resultados de cada uno de las observaciones, ambos graficos son guardados en las carpetas “/Imagenes/Ejercicio_a” o “/Imagenes/Ejercicio_b” respectivamente dependiendo del enunciado.

Presentación de los resultados

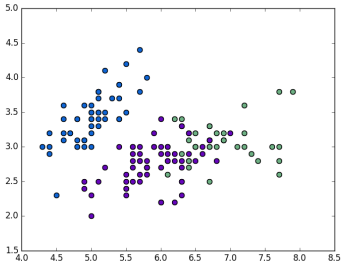
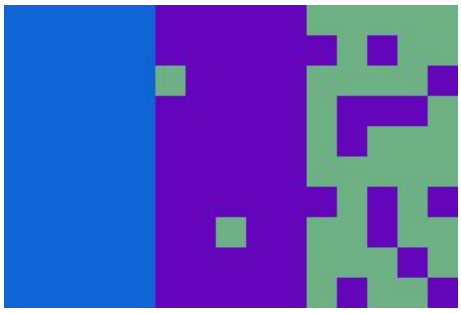
Para la presentacion de los resultados se escogieron las mejores corridas del algoritmo en base a k a fin de poder presentar la información mas veraz del mismo.

Ejercicio a

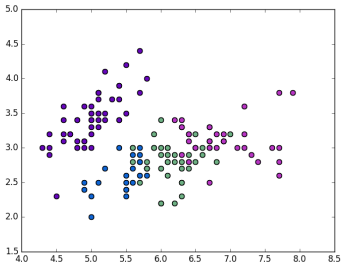
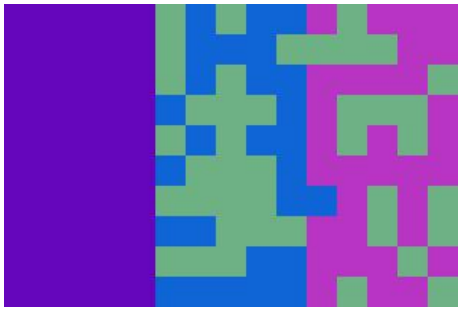
- k = 2

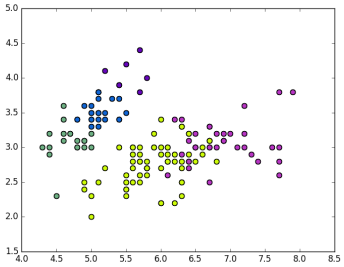
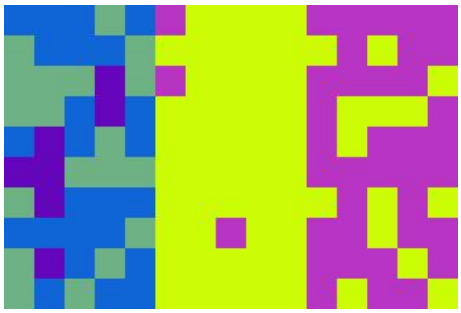
- k = 3

- k = 4

- k = 5

Se puede apreciar que para las corridas con k = {2,3} el algoritmo puede clasificar con un alto grado de eficiencia el cluster al cual pertenece cada observación.

- k = 2 Para el caso de k = 2 el algoritmo genera un solo cluster que contiene 2 clasificaciones en conjunto a pesar de ello usalmente entre 3 a 5 de las 150 observaciones es catalogado de manera incorrecta teniendo en cuenta que el conjunto "Iris versicolor" e "Iris virginia" pertenecen al mismo cluster. Generando asi un margen de error de 2% a 3% en la clasificaciòn
- k = 3 Para el caso de k = 3 el algoritmo genera un solo cluster que contiene 2 clasificaciones en conjunto a pesar de ello usalmente entre 10 a 16 de las 150 observaciones es catalogado de manera incorrecta teniendo en cuenta que el conjunto "Iris setosa" es catalogado perfectamente en el 100% de los casos pero debido a la similitud de las observaciones de "Iris versicolor" e "Iris virginia" el algoritmo tiende a categorizarlas de manera erronea, se deben realizar multiples corridas del algoritmo para poder tener en cuenta cual realmente es la clasificacion de los ultimos 2 conjuntos. Generando asi un margen de error de 6% a 10% en la clasificaciòn.

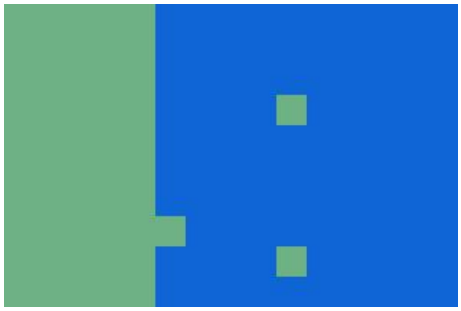
Ejercicio b

Originalmente el conjunto a estudiar se puede presentar en una imagen comprimida. En este caso cada pixel en la imagen original es un ejemplo, que será asignado en la imagen comprimida al cluster ki . K será el número de colores en la imagen comprimida.

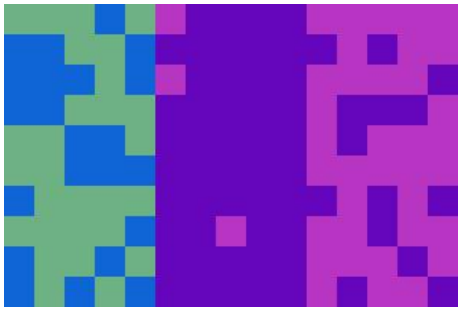
- Formalmente la imagen original deberia ser de la siguiente forma:

Se evalua el algoritmo para los siguientes conjuntos:

- k = 2

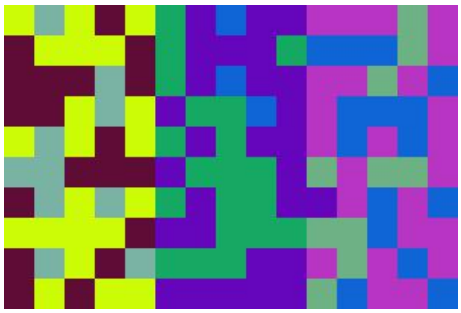


- k = 4

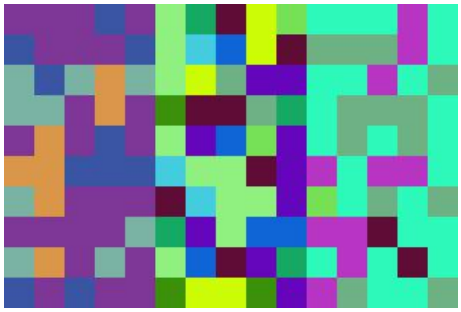


- k = 8

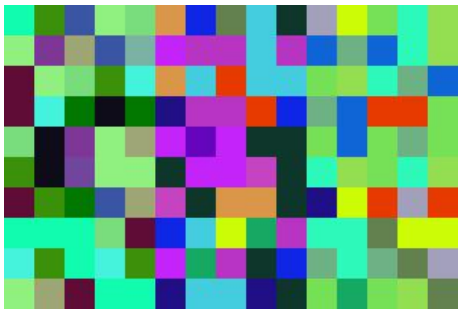
- k = 16



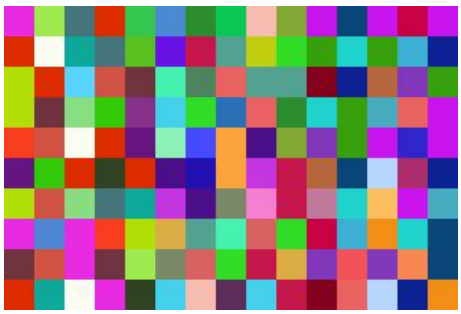
- k = 32



- k = 64



- k = 128



Conclusiones

Se pudo observar al realizar varias corridas del algoritmo que los resultados de k-means pueden variar drasticamente dependiendo de donde se inicialize los centroides de los clusters.

Mediante un Diagrama de Voronoi se podrian generar los espacios a los cuales cada conjunto se encuentra definido permitiendo predecir en que conjunto puede ser clasificado un punto.

En el caso de conjuntos que se encuentren muy proximos los casos bordes de dichos conjuntos pueden variar a la hora de categorizar los resultados.

Cuando se realiza una corrida con una gran cantidad de centroides los mismos convergiran a un solo valor en las observaciones creando overfitting d ela data.

En cualquiera de los casos superiores a $k=4$ los cluster que se generan son subconjuntos de los clusters cuando $k=3$ esto debido a que la tendencia predictiva del algoritmo converge hacia sonas similares definidas.