INF391 Reconocimiento de Patrones en Minería de Datos
Tarea 1

Universidad Técnica Federico Santa María, Campus San Joaquín
Departamento de Informática

11 de mayo de 2015

Profesor Marcelo Mendoza

Juan Pablo Escalona
juan.escalona@alumnos.usm.cl
201073515-k

Rafik Masad
mailrafik@alumnos.usm.cl
201073519-2

Gianfranco Valentino
mailgina@alumnos.usm.cl
2860574-9

Introducción

En el presente informe se analizan los resultados obtenidos comparando diferentes técnicas de clustering sobre el dataset iris incluido en la librería *sklearn*.

Análisis de los resultados obtenidos

1.1. k-means

Algoritmo: k-means

Parámetros utilizados:

- n_clusters: 3
- tol: 0.1
- max_iter: 300
- n_jobs: 1

Resultados

- Errores: 14
- Tiempo de ejecución: 0.0123851 [s]

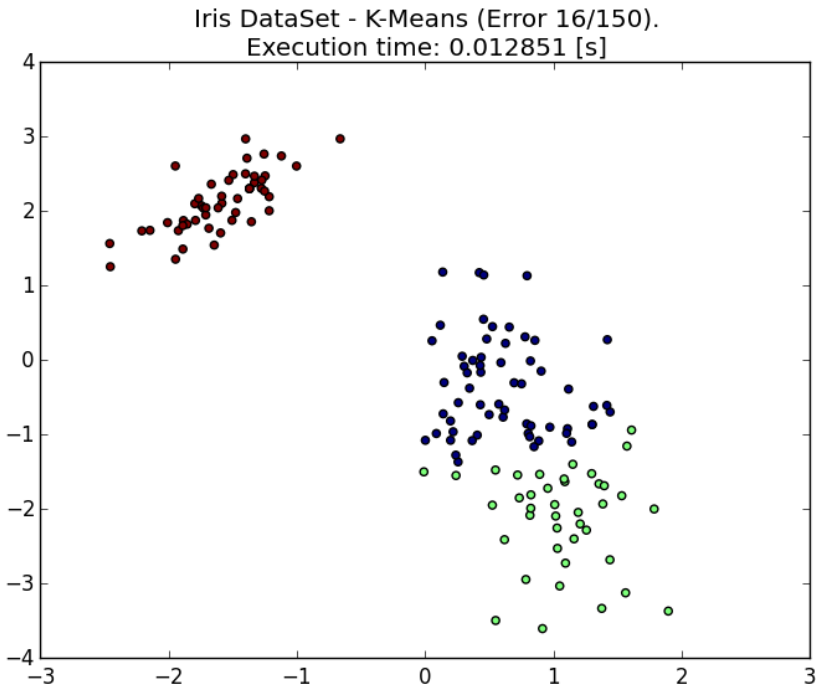

Figura 1: K-means++

1.2. Minibatch k-means

Algoritmo: Minibatch k-means

Parámetros utilizados:

- n_clusters: 3
- reassignment_ratio: 0.01
- max_iter: 100
- batch_size: 5
- init: 'k-means++'
- n_init: 3
- tol: 0.5

Resultados

- Errores: 15
- Tiempo de ejecución: 0.016015 [s]

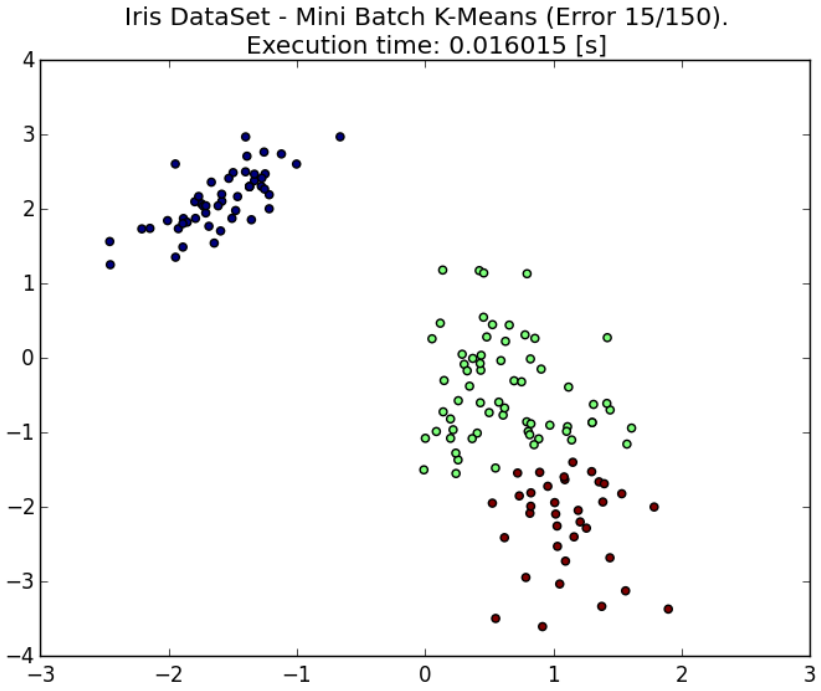

Figura 2: Minibatch k-means

En general k-means, al converger a un máximo local, es extremadamente dependiente de las semillas, explicando así las diferencias en la calidad de los resultados por ejecución.

1.3. HAC

Algoritmo: HAC

Parámetros utilizados:

- n_clusters: 3
- affinity: euclidean
- n_components: None
- linkage: average

Resultados

- Errores: 16
- Tiempo de ejecución: 0.002657 [s]

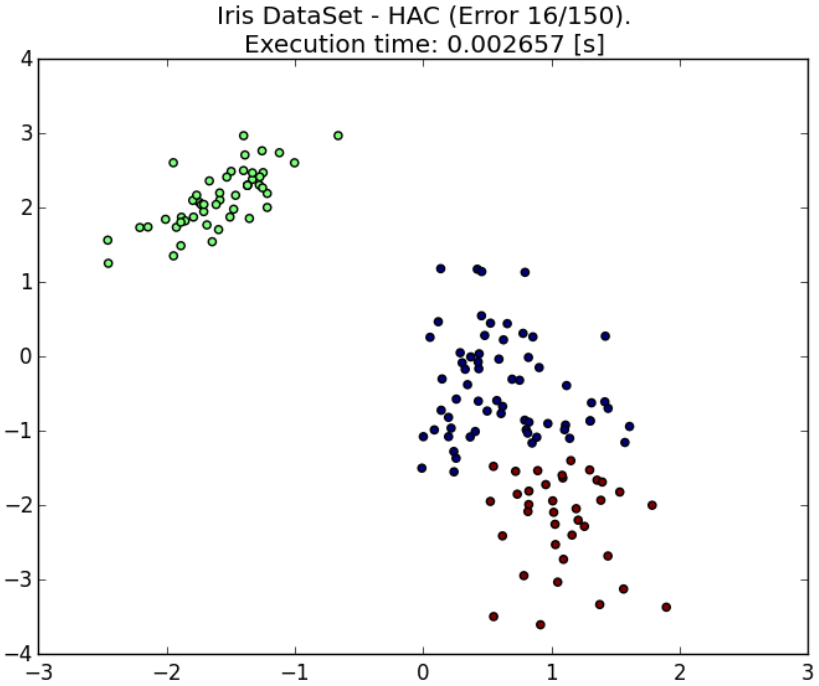

Figura 3: HAC

1.4. Ward

Algoritmo: Ward

Parámetros utilizados:

- n_clusters: 3

Resultados

- Errores: 16
- Tiempo de ejecución: 0.011104 [s]

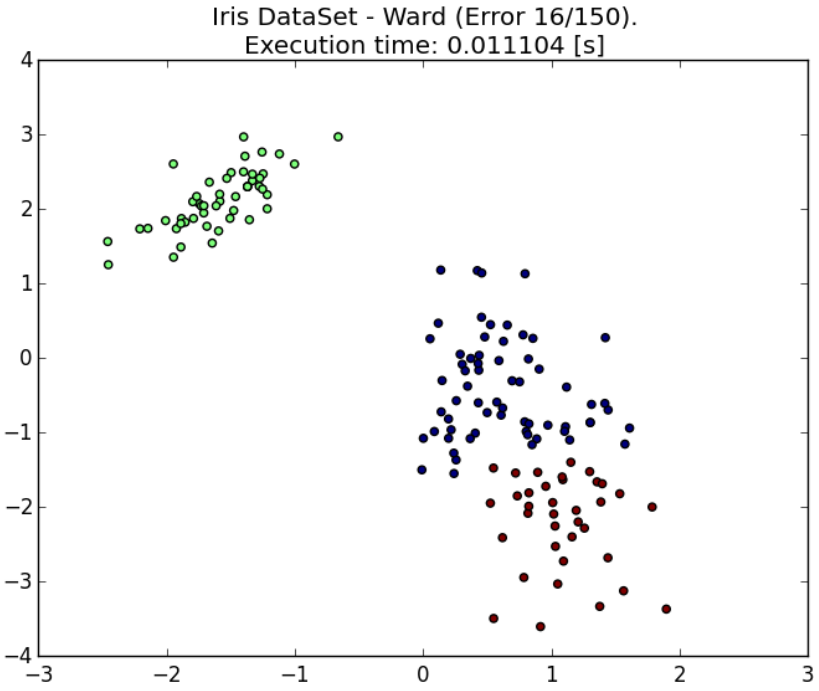

Figura 4: Ward

1.5. DBScan

Algoritmo: DBScan

Parámetros utilizados:

- min_samples: 14
- eps: 0.5

Resultados

- Errores: 22
- Tiempo de ejecución: 0.031344 [s]

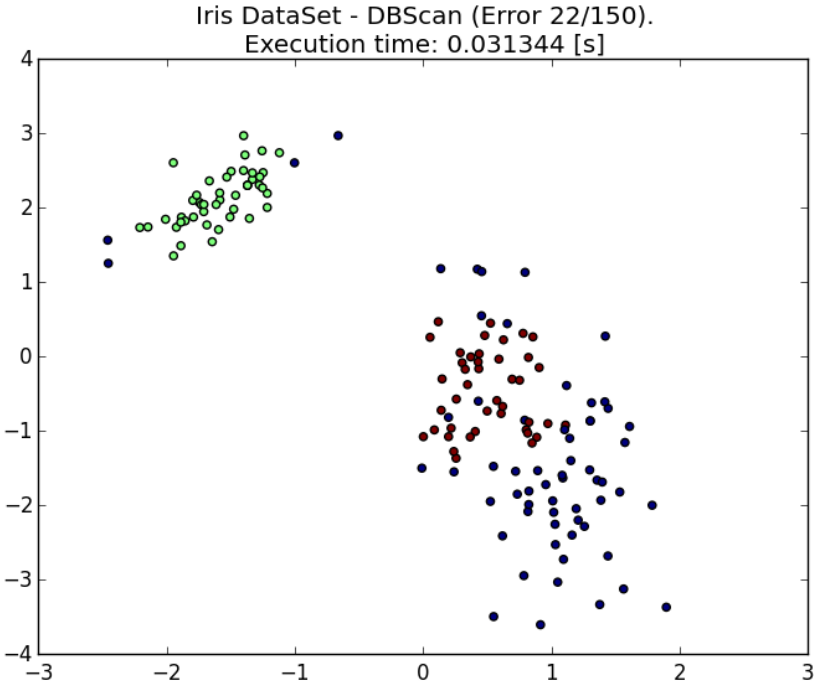

Figura 5: DBScan

1.6. C-Means

Algoritmo: C-Means

Parámetros utilizados:

- c: 3
- m: 0.01
- error: 0.3
- maxiter: 20
- seed: None

Resultados

- Errores: 10-50[1]
- Tiempo de ejecución: 0.017491 [s]

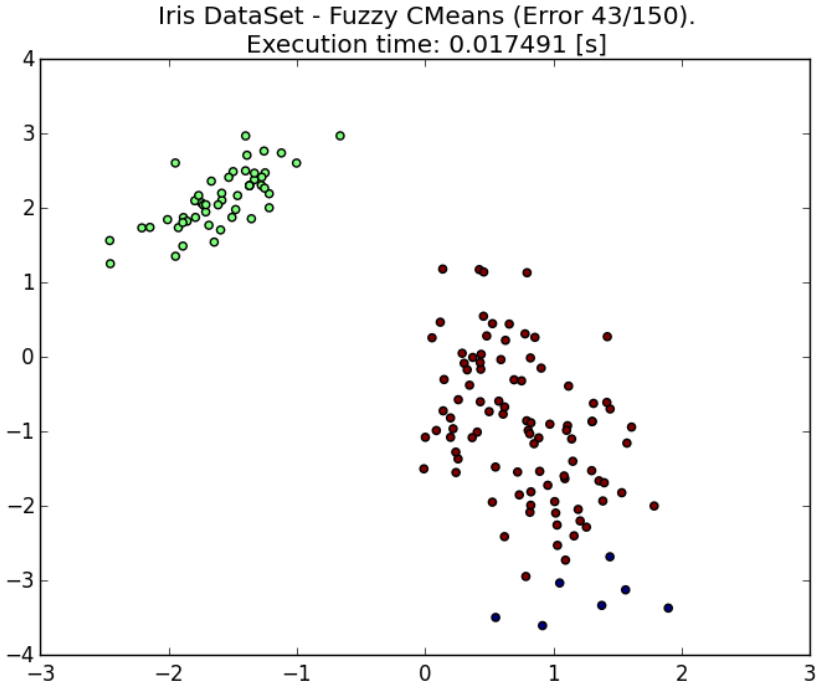

Figura 6: C-Means

[1] Muy variable

1.7. Mean shift

Algoritmo: Mean shift

Parámetros utilizados:

- bandwidth: 0.9
- bin_seeding: True
- min_bin_freq: 1
- cluster_all: False

Resultados

- Errores: 33
- Tiempo de ejecución: 0.080781 [s]

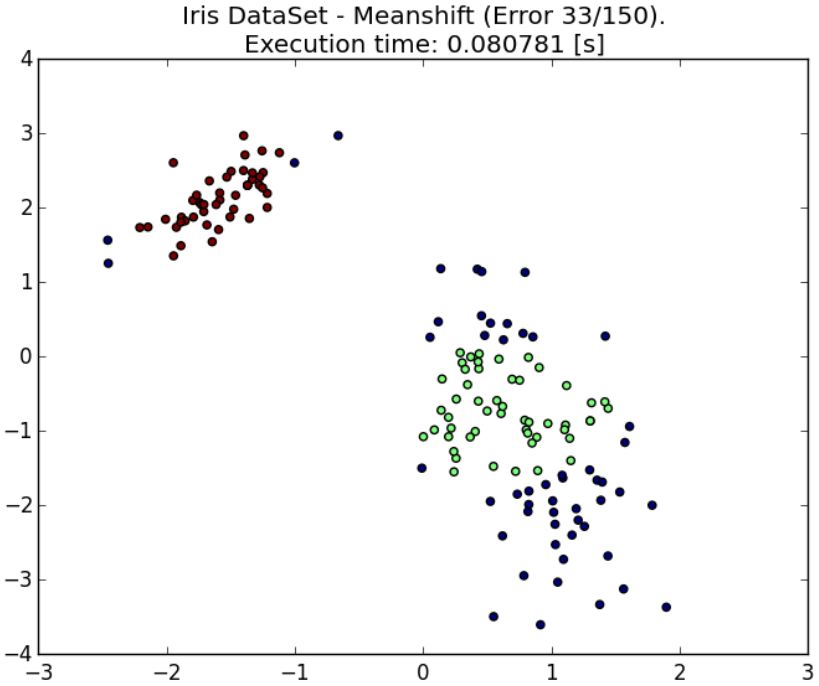

Figura 7: Mean shift

1.8. Spectral Clustering

Algoritmo: Spectral Clustering

Parámetros utilizados:

- n_clusters: 3
- n_components: 3
- eigen_solver: 'arpack'
- assign_labels: 'discretize'
- n_init: 1
- weight: 9.5

Resultados

- Errores: 6
- Tiempo de ejecución: 0.064943 [s]

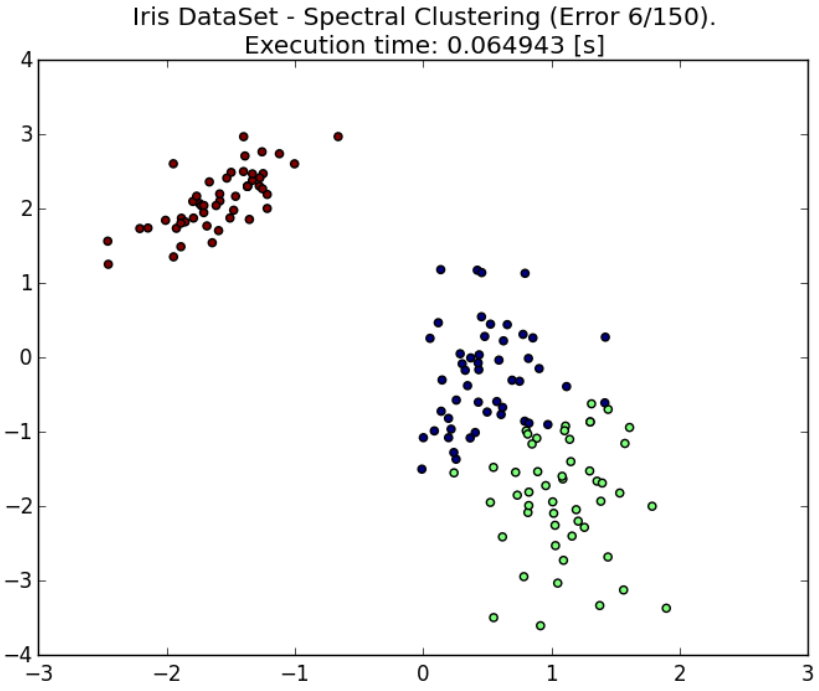

Figura 8: Spectral Clustering

2 Tabla comparativa

En la siguiente tabla se resumen los errores y tiempos de cada algoritmo

Algoritmo	Errores	Tiempo [s]
Spectral Clustering	6	0.064943
k-means	14	0.012385
Minibatch k-means	15	0.016015
HAC	16	0.002657
Ward	16	0.011104
DBScan	22	0.031344
Mean shift	33	0.080781
C-Means	10-50	0.017491

Conclusiones

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Referencias

- —

Anexo