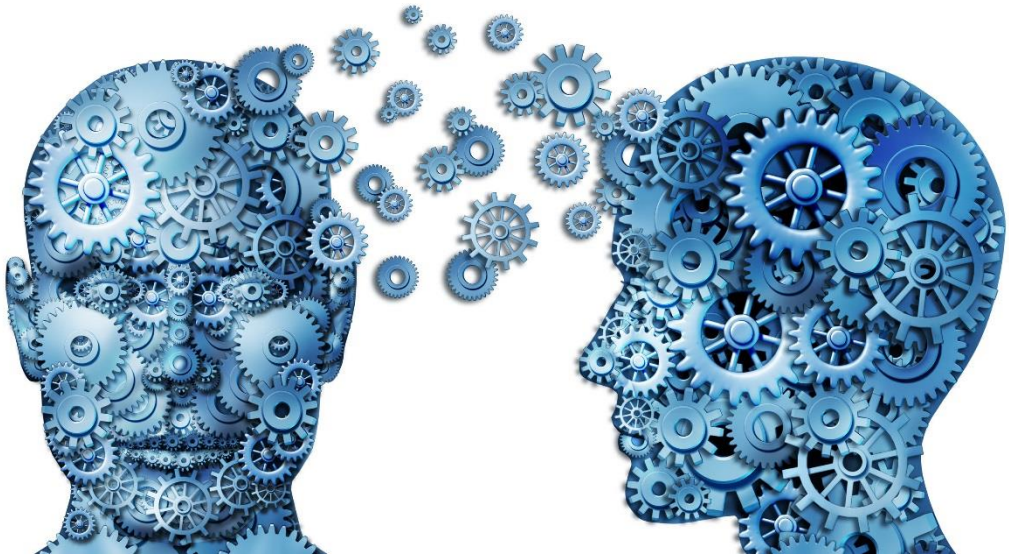
Práctica 3: Clustering



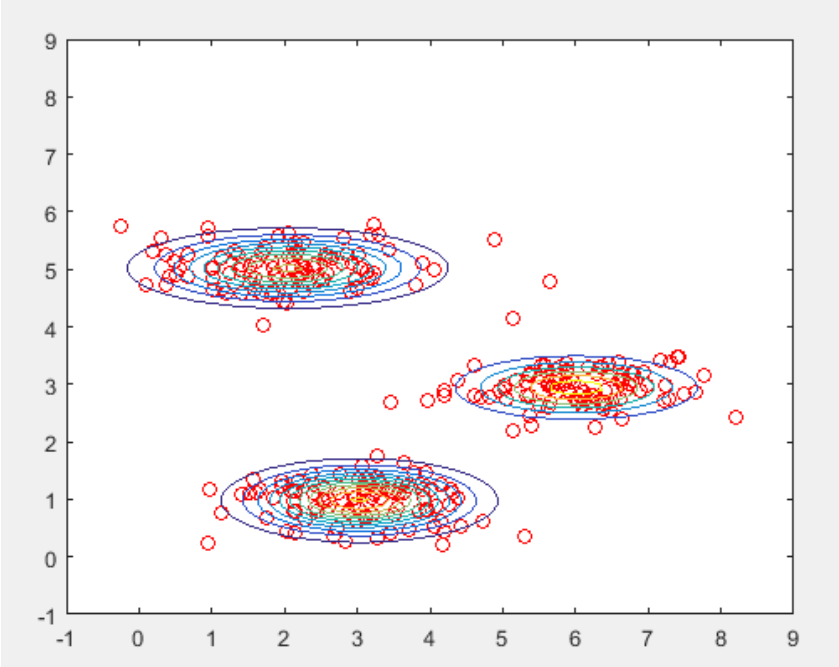
Uxue Ayechu

Aprendizaje Formal

Algoritmo Maximización-Expectación

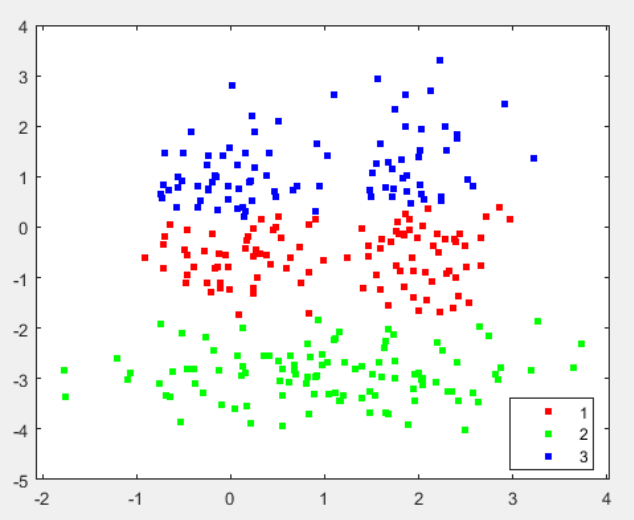
Lo primero que hemos hecho en esta práctica ha sido programar el algoritmo Maximización-Expectación. Para ello hemos generado la función EM, a la cual le pasamos los datos, las matrices de covarianza, las medias y las probabilidades de cada cluster. Dichos valores se inicializan aleatoriamente. La función EM estima las probabilidades de cada ejemplo para cada clase, las probabilidades de pertenencia y a partir de estos datos recalculamos las medias y las matrices de covarianza hasta obtener la resultante. Dichos cálculos se divide en dos partes, **estimación** que es la parte en la que se calcula la verosimilitud utilizando la estimación de los parámetros en ese momento, y la parte de **maximización** donde se calculan los parámetros que maximizan la verosimilitud esperada del paso anterior. Habremos obtenido la mejor clasificación, o nos habremos estancado, cuando de una iteración a otra apenas obtenemos cambios.

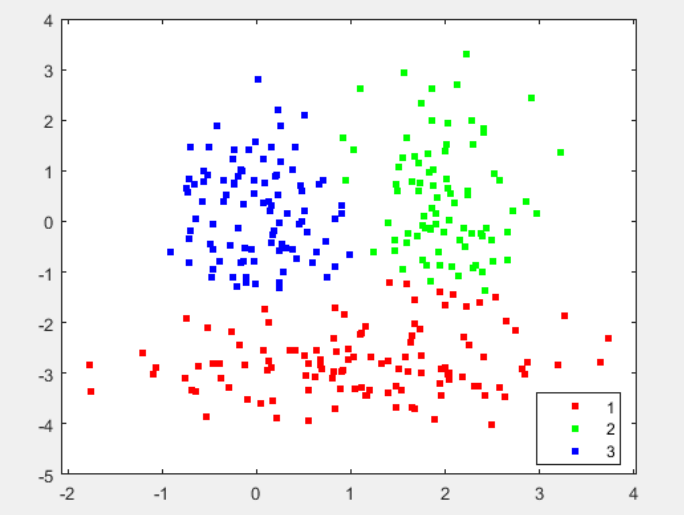
Gráfico EM para los datos “ex7data”



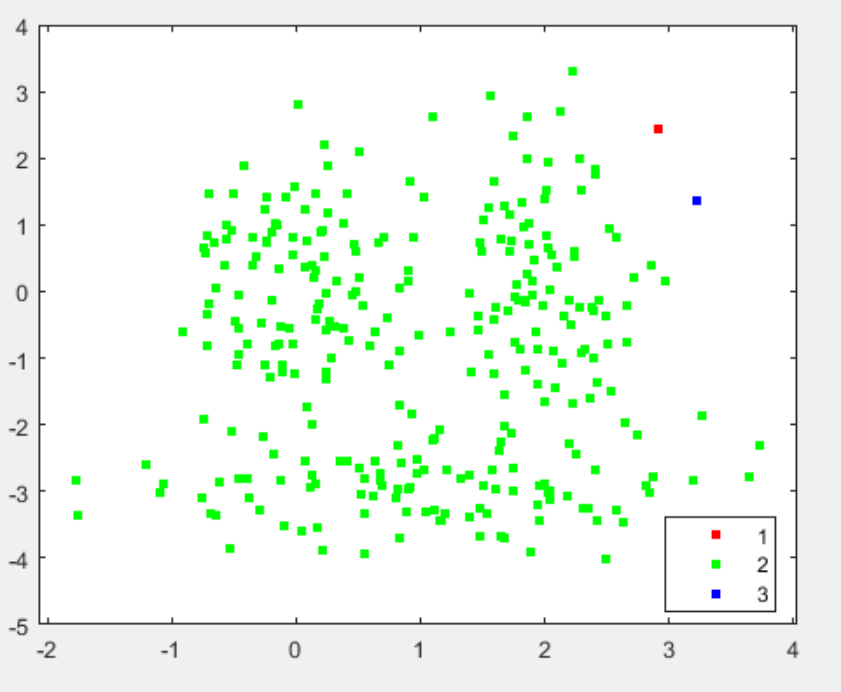
Comparar diferentes técnicas de clustering para los datos “datos3.mat”

k-means

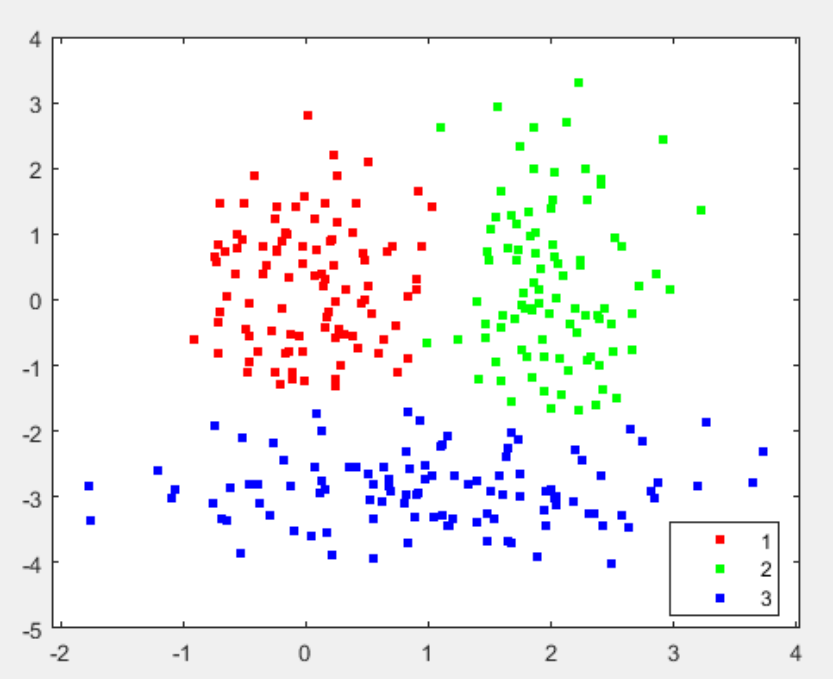




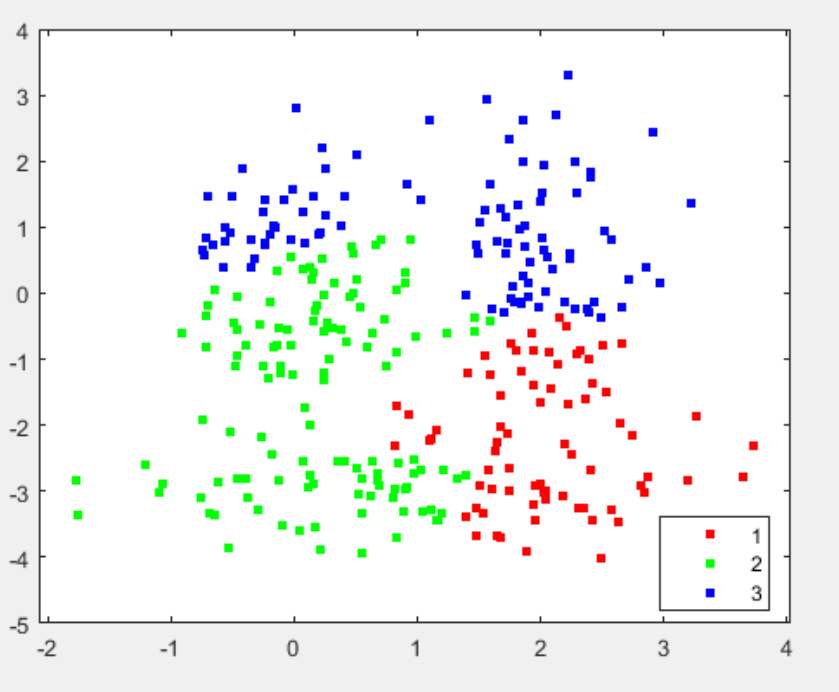
Clustering jerárquico: single-link



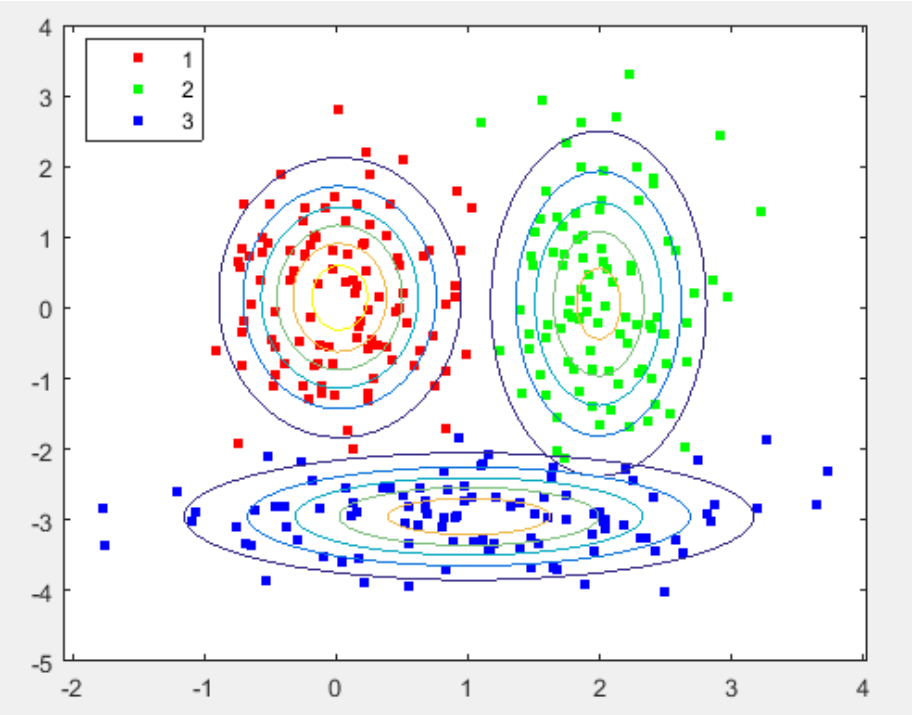
Clustering jerárquico: average-link



Clustering jerárquico: complete-link



EM



Comparación del porcentaje de aciertos, calculado con RandIndex

RI (k-means) = 0.761962095875139 (figura izquierda) || 0.915183946488294 (figura derecha)
RI (single-link) = 0.335629877369008
RI (complete-link) = 0.657814938684504
RI (average-link) = 0.943656633221851
RI (EM) = 0.960824972129320

Como podíamos observar, las gráficas single-link y complete-link, nos daban muy malos resultados y así lo corrobora el RI. Hemos obtenido estos resultados para single-link debido a que este método clasifica cada ejemplo a la clase del ejemplo más cercano y en estos datos podemos ver que todos los ejemplos están próximos luego se confunde y asigna casi todos a la misma clase, lo cual se denomina “encadenamiento”. El método de clustering jerárquico complete-link lo que hace es asignar las clases a partir de la máxima distancia, lo que sucede en este caso es que hay puntos del mismo cluster bastante lejos, lo cual confunde al clasificador que los asigna a distintas clases en vez de a la misma.

Los demás métodos funcionan correctamente, average-link es una mezcla de complete-link y single-link y cómo podemos ver gracias a esta mezcla evita el problema de tener ejemplos de la misma clase lejanos y el de tener ejemplos de distintas clases cercanos.

El clustering de k-means funciona regular algunas veces (figura1), debido a que los datos están muy próximos lo cual le genera dudas a la hora de dividir los ejemplos en clusters. Además en este caso el número de elementos de cada cluster no es similar, pues el cluster inferior es el doble que los otros. Sin embargo, en otras ocasiones el k-means sí que resuelve el problema de manera sobresaliente, esta variación se debe a la inicialización aleatoria de algunos parámetros. Como podemos ver en la figura 2 hemos obtenidos buenos resultados con un 91% de aciertos que en comparación con la figura 1 que obteníamos un 76% de aciertos obtenemos una mejoría de hasta un 15%. En definitiva, podemos decir que el algoritmo k-means varía mucho de una iteración a otra, pero podemos obtener buenos resultados.

Por último, podemos observar en los datos de RI que **el mejor clasificador es el de EM** que acierta hasta un 96% de los ejemplos, esto se debe a que este método encuentra la estimación de máxima verosimilitud de los parámetros en modelos estadísticos y no se basa en la distancia como los otros algoritmos de clustering estudiados.