Minería de datos Práctica 1:Clustering knn-means

Jose Ignacio Sánchez
Josu Rodríguez

26 de octubre de 2014

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

1. Introducción

El objetivo principal de esta práctica es obtener la capacidad de formular un algoritmo de aprendizaje automático de clasificación ***No-Supervisada***. Por otra parte, se trabajarán la capacidad de sintetizar uns técnica de aprendizaje automático no-supervisado, conocer su coste computacional así como sus limitaciones de representación y de inteligibilidad

2. Recursos

- PC con aplicación Weka.
- Bibliografía.
- Librerías de Weka.
- Manual de Weka.
- Guía de la práctica.
- Ficheros para los datos de la práctica: food.arff, colon.arff.
- Otros ficheros que no están en formato *.arff*:
  - En formato *.txt*: ClusterData.atributos.txt (este fichero si tiene la clase asociada para evaluar la calidad del *clustering* en ClusterData.clase.txt).
  - En formato *.csv* bank-data.csvclustering

3. Clasificación NO-supervisada o *Clustering*

(*Definición) 3.1 Técnicas de aprendizaje donde no hay un conocimiento a priori, donde agrupa las instancias sin atributos dependientes pre-especificados.Los algoritmos de "clustering" son un método común de aprendizaje no supervisado.*

3.1. Clustering *k-means*

4. Diseño

Estructuramos la ejecución del algoritmo en fases como se puede ver en la figura 4 , las cuales se detallan a continuación.

Primera fase: carga de datos y configuración

Inicialmente se carga la configuración establecida por el usuario en el fichero **kmeans.conf**, es decir: path del fichero y su formato, tipo de inicialización para el *codebook*, número de clusters, distancia a utilizar, número de clústers deseados, elección manual o automática sobre la normalización y diversas opciones más, especificando datos sobre el fichero que se utilizará para las instancias.

Segunda fase: Preproceso de datos

En el preproceso de datos se normalizará o no, dependiendo del parámetro indicado por el usuario. Si el parámetro es 0 no se normalizará, si es 1 se normaliza y si es 2 se hará uso del método experimental para decidir la idoneidad de la normalización.

Tercera Fase: Algoritmo K-means

En esta fase se implementa el algoritmo **K-means**.

1. En primer lugar inicia los *centroides* con el criterio establecido por el usuario, o la matriz de bits de pertenencias.
2. Recorre las instancias del conjunto y calcula la distancia a cada uno de los *codeword* actualizando la matriz de bits de pertenencia,el valor del bit es uno si es el centroide más cercano a la instancia.
3. Se calcula de nuevo el vector promedio para cada cluster.
4. Iterar los pasos dos y tres hasta converger.

Cuarta Fase: Evaluación

Existen distintos métodos propuestos de evaluación de Clustering, los que conocemos actualmente pueden dividirse en dos grupos:

- *Extrinsic methods:* Se aplican cuando disponemos de la etiqueta de las instancias, asignando una puntuación en función de las instancias correctamente agrupadas.
- *Intrinsic methods:* Se aplican cuando no disponemos de la clase. Los métodos que conocemos intentan medir la cohesión **intra-cluster** y la separación **inter-cluster**

Inicialmente pensamos en utilizar un método de los que utilizan datos de los que se conoce la clase previamente. Pero siguiendo el criterio de que se trata de implementar un sistema capaz de explorar patrones comunes en conjuntos de datos, de los que **NO** se conoce la clase a priori, evaluar con un problema que no se ajusta a este hecho se nos antoja que no es una medida muy realista, dado que se conocen el número de clusters "óptimo" a priori y que atributos correlan mejor con la clase.

Además en un contexto real en el que se intenta explorar un conjunto de datos para encontrar patrones similares, no se dispone de las predicciones o de la etiqueta de la clase, por lo que nos decidimos a analizar e implementar un método de evaluación.

Otro de los motivos para implementar nuestro propio método nace de la base para nosotros necesaria de independencia, es decir queremos un sistema capaz por si sólo de decidir cuanto de bien agrupa las instancias, para éste fin como es de esperar lo que buscamos es un índice indicador de la cohesión de las instancias agrupadas en un mismo cluster.

El método escogido ha sido el coeficiente **Silhouette**, esto es así porque como se explicará mas adelante, para calcular el indicador se utilizan medidas de cohesión y además de separación, por lo que nos parece que se ajusta a lo que buscamos para nuestro sistema, teniendo en cuenta que tratamos con heurísticos relativamente novedosos y de base experimental.

4.1. Algoritmo en pseudocódigo

```
Let k be the number of clusters to partition the data set
Let X = x_1, x_2, ..., x_n be the data set to be analyzed
Let M = m_1, m_2, ..., m_k be the code-book associated to the clusters
Let dist(a, b) be the desired distance metric
Let B = B_11, B_12, ..., B_nk be the temporary pertenece bit matrix

Ensure: C = C_1, C_2, ..., C_k set of clusterized instances

Begin:

  //randomly initialize the first centroids
  for each m_j
    m_j = randomsample(X)
  end

  //assign dataset instances to each cluster generated by the centroids
  for each x_n
    B_nj = 1 if argmin dist(x_n, m_j) = m_j \foreach m_j else B_nj = 0
  end

  for each B_nj
    if B_nj == 1
      C_j.add(x_i)
    end
  end

  //iterate the algorithm generatin new centroids based on previously clusterized instances until
      there are no changes between iterations
  while changes in M do
    for each m_j
      m_jnew = calculatecentroid(C_j)
      if m_jnew == m_j
        changes = false
      else
        changes = true
      end
      m_j = m_jnew
    end

    for each x_n
      B_nj = 1 if argmindist(x_n, m_j) = m_j \foreach m_j else B_nj = 0
    end

    for each B_nj
      if B_nj == 1
        C_j.add(x_i)
      end
    end
  end

  return C = C_1, C_2, ..., C_k
end
```

5. Implementación

5.1. Formato de entrada de datos

Este particular es el que menos tiempo y esfuerzo nos ha llevado en la implementación, ya que el manejo de archivos se encuentra resuelto con clases disponibles en el API de java.

Gracias al alto nivel de configuración del sistema, que se pasará a explicar a continuación, éste es capaz de tratar ficheros de entrada de los tres tipos propuestos: ARFF, TXT, CSV.

Tanto para los ficheros con extensión arff como txt hemos realizado nuestra propia implementación, para los ficheros de entrada hemos decidido hacer uso de la librería OpenCSV disponible de manera libre en Internet, aunque podrían ser tratados como los ficheros txt de manera interna en nuestro sistema, decidimos hacer uso de esta librería ya que desconocíamos de su existencia y nos pareció útil a la vez que práctico hacer uso de ella.

5.2. Configuración del sistema

Cómo hemos nombrado anteriormente en este documento, el sistema es altamente configurable, por lo que los argumentos de configuración son numerosos y sabemos, dada nuestra propia experiencia, que no es eficiente tratar un número alto de argumentos de entrada a través de la línea de comandos.

La decisión a este respecto ha sido crear un fichero de configuración a través del cual el usuario tiene la posibilidad de ajustar la ejecución a sus datos o incluso realizar diferentes ejecuciones con distintos parámetros en la búsqueda de una ejecución óptima.

Parámetros:

- file : donde indicaremos el path del fichero que contiene el conjunto de datos.
- k: donde indicaremos el número de clusters para la ejecución.
- iterations: si este es 0 la parada ejecución se decidirá por disimilitud de los codebook.
- difference: valor para ponderar la diferencia de las distancias entre las instancias y los centroides. Es decir el cambio de pertenencia.
- distance: el exponente de la distancia Minkowski.
- initialize: para inicializar con una matriz de pertenencia aleatoria(0) o con instancias del conjunto escogidas aleatoriamente como codewords(1).
- file_ extension: indica la extensión del archivo.
- data_ line_ start: permite al usuario indicar la línea en la que comienzan los datos. Este parámetro nos pareció interesante por dos motivos. El primero es que permite manejar archivos con cualquier información antes de comenzar a extraer los datos y el segundo es que se pueden obtener datos conjuntos de datos de diferentes tamaños extraídos de un mismo fichero.
- delimiter: para indicar el delimitador entre los distintos atributos.
- normalize: para dejar que el usuario decida si normalizar(1) o no(0).Por otra parte cabe la posibilidad de dejar que el sistema decida si normalizar o no(2).
- ratio_ max: para indicar la disimilitud entre codebooks(0.0 distintos, 1.0 iguales).

5.3. Análisis del conjunto de datos para decidir la conveniencia de la normalización

Debido a las dudas surgidas en torno a la normalización de los atributos y su conveniencia, consideramos adecuado para la tarea tratar de buscar algún método que fuese de alguna manera indicador de la utilidad de la normalización.

Inicialmente nuestro planteamiento se basaba en utilizar la media de la desviación típica de los valores de cada atributo con el objeto de poder analizar los rangos de las diferencias entre valores de cada atributo. Sin embargo la media por su cuenta no nos es útil, ya que por ejemplo, si la media es alta pero la desviación es baja, en realidad, puede no haber mucha variación en los rangos. Esto nos llevó a hallar lo que denominamos Coeficiente de Variación:

$$C_V = \frac{\sigma}{\bar{x}} \tag{1}$$

De esta forma logramos un indicador más preciso sobre el rango que buscamos ya que nos indica la proporción de la variabilidad de las desviaciones, en lugar de la mera cantidad de desviación.

La motivación de este análisis viene dada más por el interés de hallar una forma de conocer la utilidad que pueda tener la normalización en un conjunto de datos, ya que objetivamente, el beneficio principal que puede tener es que si no afecta demasiado al resultado final, nos puede interesar más tener los atributos en su rango numérico inicial (p.ej.: es más visual ver un gráfico con edades entre 0 y 100 que entre 0 y 1).

Por otra parte, hemos tratado de hallar una cifra del Coeficiente de Variación que esté comúnmente aceptada como baja, pero no hemos encontrado ninguna fuente fidedigna para ello. Por lo tanto, hemos decidido establecer una cifra pequeña (0,1) teniendo en cuenta que ante la posibilidad de que haya rangos muy variados, puede ser más conveniente normalizar.

Huelga decir que este método y sus resultados son experimentales, pese a tener cierta base empírica carecemos de la certeza sobre su eficacia, siendo nuestro objetivo principal investigar respecto a la normalización en lugar de simplemente aplicarla por estar aceptada como conveniente.

5.4. Evaluación: Silhouette Coefficient

Es una combinación de las medidas de separación y cohesión:

Figura 1: Medidas de cohesión y separación

El coeficiente Silhouette s puede ser calculado para puntos independientes y para clusters. Para un punto individual,a=la distancia promedio de i a los puntos del mismo cluster;b=la distancia promedio de i a los puntos de los otros clusters 2.

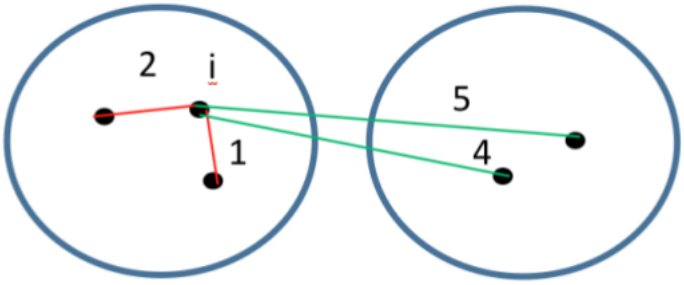

Figura 2: Coheficiente silhouette para un punto

Fuente e imágenes extraídas de (1)[pags 3,4]

Dependencias

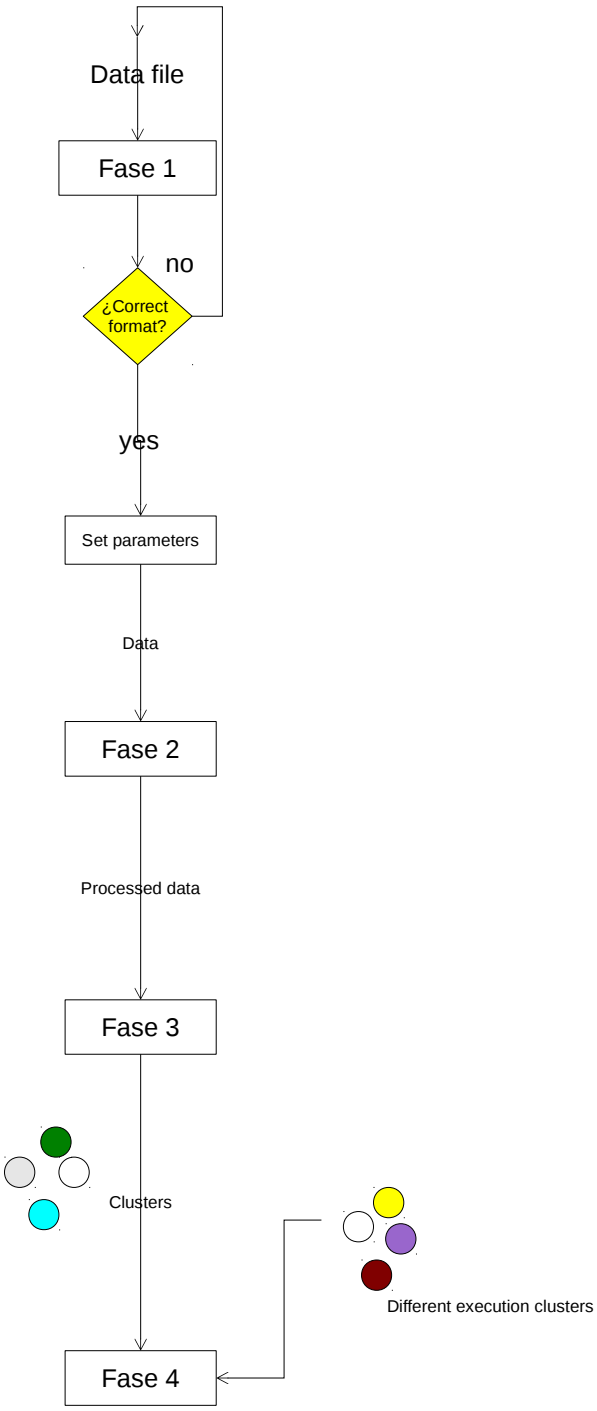

Figura 3: Dependencias del sistema

Para calcular el coeficiente del conjunto total de los clusters inicialmente se calcula el coeficiente para cada miembro de un mismo cluster y se calcula la media de todos los coeficientes hallados,se calcula ésto para cada cluster y se devuelve la media de todos los coeficientes hallados para todos los clusters.

5.5. Visualización

Con el fin de analizar los resultados de una forma visual, se presenta la representación gráfica de la matriz de pertenencias, el resumen de la ejecución por consola, y un informe detallado de la ejecución en formato pdf. Para este fin y dado que nos parece que no aporta a las competencias que se espera adquirir con la práctica hacemos uso de la librería **JFreeChart** para realizar los gráficos y de **iTex** para generar el informe en PDF.
Al finalizar el sistema muestra una gráfica de la matriz de pertenencias, permitiendo analizar de una forma visual a que cluster pertenece cada instancia:

Figura 4: Pertenencias

5.6. Problemas encontrados y soluciones adoptadas

5.6.1. Problema con la normalización de los atributos

En un comienzo, para normalizar, nuestra intención era aplicar la función Z-Score sobre los atributos de las diferentes instancias. Sin embargo, de esta forma se obtenían resultados fuera del rango de [-1, 1], que es incorrecto para esta normalización. No se consiguió localizar el error, desconociendo si se trataba de una formulación incorrecta o un bug de programación. En lugar de ello se ha realizado una proyección lineal al intervalo [0,1] y de esta forma se ha conseguido unificar los atributos en el mismo intervalo.

5.6.2. Problema de generación de excesivas instancias

Una vez estructurado todo el código y con todos los módulos programados, nos encontramos con el problema de que durante la ejecución, se generaban demasiadas instancias. Esto se debía a que en cada iteración del algoritmo, los clusters no eran reseteados correctamente, y cada vez que se añadía una nueva instancia en lugar de eliminar la que estaba presente en su lugar, desplazaba esta última y se insertaba en su posición. Para corregir esto sencillamente se modificó la función de añadir instancias para que esta sobreescribiese la presente, y además, se vacían de instancias los clusters en cada iteración.

5.6.3. Instancias repetidas en los gráficos

Al igual que en el anterior problema, nos encontramos con que a la hora de visualizar los resultados, se marcaban más instancias de las que realmente había. La solución también pasó por realizar un reinicio de la

matriz de bits de pertenencia en cada iteración del algoritmo.

5.6.4. Ausencia de representación para el cluster 0

Una vez incorporado el código necesario para la generación de gráficos su funcionamiento era correcto, pero no mostraba los datos del cluster 0. El problema radicaba en que a la hora de tratar la matriz de bits de pertenencia, el primer caso se trataba fuera del bucle principal para obtener la distancia inicial con la que comparar,. Una vez corregido esto el gráfico se crea correctamente.

5.7. Métrica o distancia utilizada

La métrica utilizada para medir la distancia entre instancias del conjunto es la **Distancia de Minkowski**:

Siendo:

$$P = (x_1, x_2, \ldots, x_n) \text{ y } Q = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n$$

La distancia Minkowski entre ambas instancias está definida por:

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Esta distancia es una generalización de la distancia Euclídea. En ella, variando el parámetro p, se pueden obtener distintas distancias. Siendo p=1 la distancia es la denominada Manhattan, con p=2 es la distancia Euclídea.(3)

Se ha programado como una implementación directa de su aplicación matemática, con una función que dadas dos instancias y el parámetro p devuelve la distancia entre ambas.

La ventaja de utilizar esta distancia es la flexibilidad que aporta variando su exponente para obtener diferentes distancias. De esta forma el algoritmo se puede adaptar a diferentes conjuntos de datos en los que la distancia entre instancias no tenga por qué ser estrictamente lineal.

6. Validación del *software*

6.1. Diseño del banco de pruebas

File	k	iterations	difference	distance	initialize	normalize	disimilitud
bank-data.csv	1	0	0.0	2	0	0	0.5
bank-data.csv	2	0	0.0	2	1	1	0.8
bank-data.csv	3	10	0.3	1	0	2	x
bank-data.csv	25	100	0	3.5	1	0	x
bank-data.csv	25	0	0	7.5	1	2	1.0
bank-data.csv	25	0	0	7.5	1	0	0.6
colon.arff	1	0	0.0	2	0	0	0.5
colon.arff	2	0	0.0	2	1	1	0.8
colon.arff	3	10	0.3	1	0	2	x
colon.arff	10	30	0	3.5	1	0	x
colon.arff	10	0	0	7.5	1	2	1.0
colon.arff	10	0	0	7.5	1	0	0.6
ClusterData.atributos.txt	1	0	0.0	2	0	0	0.5
ClusterData.atributos.txt	2	0	0.0	2	1	1	0.8
ClusterData.atributos.txt	3	10	0.3	1	0	2	x
ClusterData.atributos.txt	30	40	0	3.5	1	0	x
ClusterData.atributos.txt	30	0	0	7.5	1	0	1.0
ClusterData.atributos.txt	30	0	0	7.5	1	2	0.6

Tras resolver los problemas de implementación expuestos en puntos anteriores, el sistema pasa el banco de pruebas completo con éxito.

7. Análisis de resultados

Analizar los resultados no es tarea fácil, no existe un método de evaluación estandarizado para el clustering, por lo que nos basamos únicamente en el coeficiente ya que nos acerca a una medida de lo bien que el algoritmo agrupa las instancias.

Dicho esto y a la vista de los resultados aportados como anexo, se puede ver que el algoritmo es bastante eficiente. Podemos observar que para agrupar en un único cluster, la evaluación siempre nos dará el mismo resultado 1.0, que es el mejor que podemos obtener.

Si nos centramos la atención en los conjuntos de datos de los que disponemos la clase, con distancia euclidea y con disimilitud 1.0 aunque no obtenemos más de 0.3 de coeficiente, podemos analizar los datos con respecto a la ejecución y ver que el número de instancias de cada clase es similar al número de instancias en cada cluster. Esto nos índica que quizás merezca analizar los datos a partir de un índice no muy alto.

7.1. Modificando inicializaciones

Debido a que las inicializaciones posibles son matriz de pertenencia aleatoria o codebook inicial escogiendo instancias del conjunto de datos, los resultados no varían demasiado con el cambio de este método, pero el resultado si depende de como se inicializa el codebook.

7.2. Criterios de convergencia

7.2.1. Número fijo de iteraciones

Por lo general el aumento de iteraciones es proporcional al resultado, aunque llegado a un número de iteraciones el resultado no varía. Para los datos manejados generalmente a partir de la iteración diez el coeficiente silhouette no varía.

7.2.2. Disimilitud entre *codebooks*

Este punto se podría analizar igual que el punto anterior, dado que la disimilitud escogida lo que permite es un mayor número de iteraciones. Es decir que aunque pongamos que la disimilitud entre codebooks sea 0.2 y no 1.0 que es la máxima similitud, es decir que no para hasta que sean iguales, cuando alcanza el número de iteraciones a partir del cual el coeficiente no varía, se alcanza el mejor resultado posible.

7.3. Distintas métricas

Tras hacer diferentes ejecuciones, tanto con distancias mahattan y euclidea como con diversos exponentes para la distancia Minkowski, se observa que los mejores resultados se obtienen con las dos primeras, para los tipos de problemas de los que disponemos. Esto podría justificar porque algunas librerías de minería de datos ya existentes como Weka no implementen Minkowski para este tipo de algoritmo ya que a partir de un exponente de tres, los resultados del coeficiente disminuyen casi hasta cero.

8. Conclusiones

8.1. Técnicas de clustering: motivación

Tal y como se ha descrito anteriormente en este documento, el propósito de la clasificación no supervisada es, opuestamente a la clasificación supervisada, **descubrir** información, en lugar de predecirla.Explorar los datos en busca de patrones de comportamiento. Los campos y casos a los que es aplicable son múltiples y diversos: en sociología, análisis genético, reconocimiento facial, etcétera. Todo área de conocimiento capaz de recopilar gran cantidad de datos es susceptible y puede beneficiarse de metodologías de clustering. (2, Capítulo 7)

8.2. Conclusiones generales

Debido a la modularidad del software realizado, nos ha resultado relativamente sencillo encontrar los orígenes de los diversos errores y poder solucionarlos así como añadir varios módulos (p. ej.: generación de gráficos e informe, cálculos estadísticos) a lo largo del desarrollo, teniendo la estructura principal del código ya hecha. Además de esto, varias de las clases generadas son genéricas y pueden ser utilizadas en otros proyectos, como las funciones estadísticas de Statistics.java o la evaluación de clusters de Evaluation.java.

Por otra parte, la realización de esta práctica nos ha llevado a conocer en mayor profundidad el funcionamiento de un algoritmo de cierta complejidad, haciendo un uso mínimo de librerías externas. De no haber sido así, no se habría descubierto el algoritmo del coeficiente Silhouette, y aún menos haberlo implementado sin acudir a código ajeno, de hecho se podría haber prescindido de estas librerías. También cabe destacar la capacidad para manejar diversas que se ha obtenido, así como el uso de varias funciones estadísticas para calcular si es apto normalizar o no (experimental).

8.3. Puntos débiles y propuestas de mejora

Dadas la complejidad y la diversidad de funciones implementadas en el software, ha habido ciertos aspectos en los que no ha sido posible centrarse tanto como cabía. Podría destacarse el rendimiento del mismo, que pese a ser correcto y no tomar demasiado tiempo con volúmenes grandes de datos, no ha sido puesto a revisión exhaustiva por lo que podría ser mejorado.

Además de esto, la cantidad de métodos de evaluación que se han podido implementar son limitados. Sin embargo no lo consideramos un punto débil destacable, ya que el algoritmo implementado, tras su análisis y diversas pruebas, nos ha parecido consistente y efectivo.

9. Valoración subjetiva

1. ¿Has alcanzado los objetivos que se plantean?
2. ¿Te ha resultado de utilidad la tarea planteada?
3. ¿Qué dificultades has encontrado?Valora el grado de dificultad de la tarea.
4. ¿Cuánto tiempo has trabajado en esta tarea? Desglosado:

Coste temporal	
Diseño de software	5
Implementación de software	40
Tiempo trabajando con Weka	1
Búsqueda bibliográfica	1
Informe	2.5

5. Sugerencias para mejorar la tarea. Sugerencias para que se consiga despertar mayor interés y motivación en los alumnos.
6. Críticas(constructivas).

ANEXOS

Datos de agrupamiento de la instancias obtenidas del archivo data/colon.arff en 2 Clusters

Autores: Iñigo Sanchez Mendez y Josu Rodríguez

25-10-2014

Parámetros de la ejecución:

Iteraciones: 0

Ponderación de comparación de las distancias de la instancia al centroide actualizado: 0.0

La distancia utilizada: Distancia Euclidea

Inicialización: 1

Normalización: 0

Disimilitud codebooks: 1.0

Conjunto de instancias en el archivo arff

CLUSTER 0

Instance 8	Instance 10	Instance 11	Instance 24	Instance 27	Instance 28	Instance 29	Instance 30
Instance 33	Instance 42	Instance 43	Instance 44	Instance 45	Instance 46	Instance 49	Instance 51

CLUSTER 1

Instance 0	Instance 1	Instance 2	Instance 3	Instance 4	Instance 5	Instance 6	Instance 7
Instance 9	Instance 12	Instance 13	Instance 14	Instance 15	Instance 16	Instance 17	Instance 18
Instance 19	Instance 20	Instance 21	Instance 22	Instance 23	Instance 25	Instance 26	Instance 31
Instance 32	Instance 34	Instance 35	Instance 36	Instance 37	Instance 38	Instance 39	Instance 40
Instance 41	Instance 47	Instance 48	Instance 50	Instance 52	Instance 53	Instance 54	Instance 55

Los resultados de la evaluación de la ejecución:
Coeficiente silhouette: 0.23516332146595348

Datos de agrupamiento de la instancias obtenidas del archivo data/bank-data.csv en 2 Clusters

Autores: Iñigo Sanchez Mendez y Josu Rodríguez

25-10-2014

Parámetros de la ejecución:

Iteraciones: 0

Ponderación de comparación de las distancias de la instancia al centroide actualizado: 0.3

La distancia utilizada: Distancia Euclidea

Inicialización: 1

Normalización: 1

Disimilitud codebooks: 1.0

Conjunto de instancias en el archivo csv

CLUSTER 0

Instance 2	Instance 3	Instance 4	Instance 5	Instance 6	Instance 7	Instance 8	Instance 9
Instance 10	Instance 11	Instance 12	Instance 13	Instance 14	Instance 15	Instance 16	Instance 17
Instance 18	Instance 19	Instance 20	Instance 21	Instance 22	Instance 23	Instance 24	Instance 25
Instance 26	Instance 27	Instance 28	Instance 29	Instance 30	Instance 31	Instance 32	Instance 33
Instance 34	Instance 35	Instance 36	Instance 37	Instance 38	Instance 39	Instance 40	Instance 41
Instance 42	Instance 43	Instance 44	Instance 45	Instance 46	Instance 47	Instance 48	Instance 49
Instance 50	Instance 51	Instance 52	Instance 53	Instance 54	Instance 55	Instance 56	Instance 57
Instance 58	Instance 59	Instance 60	Instance 61	Instance 62	Instance 63	Instance 64	Instance 65
Instance 66	Instance 67	Instance 68	Instance 69	Instance 70	Instance 71	Instance 72	Instance 73
Instance 74	Instance 75	Instance 76	Instance 77	Instance 78	Instance 79	Instance 80	Instance 81
Instance 82	Instance 83	Instance 84	Instance 85	Instance 86	Instance 87	Instance 88	Instance 89
Instance 90	Instance 91	Instance 92	Instance 93	Instance 94	Instance 95	Instance 96	Instance 97
Instance 98	Instance 99	Instance 100	Instance 101	Instance 102	Instance 103	Instance 104	Instance 105
Instance 106	Instance 107	Instance 108	Instance 109	Instance 110	Instance 111	Instance 112	Instance 113
Instance 114	Instance 115	Instance 116	Instance 117	Instance 118	Instance 119	Instance 120	Instance 121
Instance 122	Instance 123	Instance 124	Instance 125	Instance 126	Instance 127	Instance 128	Instance 129
Instance 130	Instance 131	Instance 132	Instance 133	Instance 134	Instance 135	Instance 136	Instance 137

Instance 138	Instance 139	Instance 140	Instance 141	Instance 142	Instance 143	Instance 144	Instance 145
Instance 146	Instance 147	Instance 148	Instance 149	Instance 150	Instance 151	Instance 152	Instance 153
Instance 154	Instance 155	Instance 156	Instance 157	Instance 158	Instance 159	Instance 160	Instance 161
Instance 162	Instance 163	Instance 164	Instance 165	Instance 166	Instance 167	Instance 168	Instance 169
Instance 170	Instance 171	Instance 172	Instance 173	Instance 174	Instance 175	Instance 176	Instance 177
Instance 178	Instance 179	Instance 180	Instance 181	Instance 182	Instance 183	Instance 184	Instance 185
Instance 186	Instance 187	Instance 188	Instance 189	Instance 190	Instance 191	Instance 192	Instance 193
Instance 194	Instance 195	Instance 196	Instance 197	Instance 198	Instance 199	Instance 200	Instance 201
Instance 202	Instance 203	Instance 204	Instance 205	Instance 206	Instance 207	Instance 208	Instance 209
Instance 210	Instance 211	Instance 212	Instance 213	Instance 214	Instance 215	Instance 216	Instance 217
Instance 218	Instance 219	Instance 220	Instance 221	Instance 222	Instance 223	Instance 224	Instance 225
Instance 226	Instance 227	Instance 228	Instance 229	Instance 230	Instance 231	Instance 232	Instance 233
Instance 234	Instance 235	Instance 236	Instance 237	Instance 238	Instance 239	Instance 240	Instance 241
Instance 242	Instance 243	Instance 244	Instance 245	Instance 246	Instance 247	Instance 248	Instance 249
Instance 250	Instance 251	Instance 252	Instance 253	Instance 254	Instance 255	Instance 256	Instance 257
Instance 258	Instance 259	Instance 260	Instance 261	Instance 262	Instance 263	Instance 264	Instance 265
Instance 266	Instance 267	Instance 268	Instance 269	Instance 270	Instance 271	Instance 272	Instance 273
Instance 274	Instance 275	Instance 276	Instance 277	Instance 278	Instance 279	Instance 280	Instance 281
Instance 282	Instance 283	Instance 284	Instance 285	Instance 286	Instance 287	Instance 288	Instance 289

Instance 290	Instance 291	Instance 292	Instance 293	Instance 294	Instance 295	Instance 296	Instance 297
Instance 298	Instance 299	Instance 300	Instance 301	Instance 302	Instance 303	Instance 304	Instance 305
Instance 306	Instance 307	Instance 308	Instance 309	Instance 310	Instance 311	Instance 312	Instance 313
Instance 314	Instance 315	Instance 316	Instance 317	Instance 318	Instance 319	Instance 320	Instance 321
Instance 322	Instance 323	Instance 324	Instance 325	Instance 326	Instance 327	Instance 328	Instance 329
Instance 330	Instance 331	Instance 332	Instance 333	Instance 334	Instance 335	Instance 336	Instance 337
Instance 338	Instance 339	Instance 340	Instance 341	Instance 342	Instance 343	Instance 344	Instance 345
Instance 346	Instance 347	Instance 348	Instance 349	Instance 350	Instance 351	Instance 352	Instance 353
Instance 354	Instance 355	Instance 356	Instance 357	Instance 358	Instance 359	Instance 360	Instance 361
Instance 362	Instance 363	Instance 364	Instance 365	Instance 366	Instance 367	Instance 368	Instance 369
Instance 370	Instance 371	Instance 372	Instance 373	Instance 374	Instance 375	Instance 376	Instance 377
Instance 378	Instance 379	Instance 380	Instance 381	Instance 382	Instance 383	Instance 384	Instance 385
Instance 386	Instance 387	Instance 388	Instance 389	Instance 390	Instance 391	Instance 392	Instance 393
Instance 394	Instance 395	Instance 396	Instance 397	Instance 398	Instance 399	Instance 400	Instance 401
Instance 402	Instance 403	Instance 404	Instance 405	Instance 406	Instance 407	Instance 408	Instance 409
Instance 410	Instance 411	Instance 412	Instance 413	Instance 414	Instance 415	Instance 416	Instance 417
Instance 418	Instance 419	Instance 420	Instance 421	Instance 422	Instance 423	Instance 424	Instance 425
Instance 426	Instance 427	Instance 428	Instance 429	Instance 430	Instance 431	Instance 432	Instance 433
Instance 434	Instance 435	Instance 436	Instance 437	Instance 438	Instance 439	Instance 440	Instance 441

Instance 442	Instance 443	Instance 444	Instance 445	Instance 446	Instance 447	Instance 448	Instance 449
Instance 450	Instance 451	Instance 452	Instance 453	Instance 454	Instance 455	Instance 456	Instance 457
Instance 458	Instance 459	Instance 460	Instance 461	Instance 462	Instance 463	Instance 464	Instance 465
Instance 466	Instance 467	Instance 468	Instance 469	Instance 470	Instance 471	Instance 472	Instance 473
Instance 474	Instance 475	Instance 476	Instance 477	Instance 478	Instance 479	Instance 480	Instance 481
Instance 482	Instance 483	Instance 484	Instance 485	Instance 486	Instance 487	Instance 488	Instance 489
Instance 490	Instance 491	Instance 492	Instance 493	Instance 494	Instance 495	Instance 496	Instance 497
Instance 498	Instance 499	Instance 500	Instance 501	Instance 502	Instance 503	Instance 504	Instance 505
Instance 506	Instance 507	Instance 508	Instance 509	Instance 510	Instance 511	Instance 512	Instance 513
Instance 514	Instance 515	Instance 516	Instance 517	Instance 518	Instance 519	Instance 520	Instance 521
Instance 522	Instance 523	Instance 524	Instance 525	Instance 526	Instance 527	Instance 528	Instance 529
Instance 530	Instance 531	Instance 532	Instance 533	Instance 534	Instance 535	Instance 536	Instance 537
Instance 538	Instance 539	Instance 540	Instance 541	Instance 542	Instance 543	Instance 544	Instance 545
Instance 546	Instance 547	Instance 548	Instance 549	Instance 550	Instance 551	Instance 552	Instance 553
Instance 554	Instance 555	Instance 556	Instance 557	Instance 558	Instance 559	Instance 560	Instance 561
Instance 562	Instance 563	Instance 564	Instance 565	Instance 566	Instance 567	Instance 568	Instance 569
Instance 570	Instance 571	Instance 572	Instance 573	Instance 574	Instance 575	Instance 576	Instance 577
Instance 578	Instance 579	Instance 580	Instance 581	Instance 582	Instance 583	Instance 584	Instance 585
Instance 586	Instance 587	Instance 588	Instance 589	Instance 590	Instance 591	Instance 592	Instance 593

Instance 594	Instance 595	Instance 596	Instance 597	Instance 598	Instance 599	Instance 600	Instance 601

Los resultados de la evaluación de la ejecución:
Coeficiente silhouette: 1.0

Referencias

[1] Tutorial 3. Introduction to MOA Clustering,Frederic Stahl October 2013

[2] Introduction to Machine Learning, Second Edition, Ethem Alpaydın

[3] Amorim, R.C. and Mirkin, B., Minkowski Metric, Feature Weighting and Anomalous Cluster Initialisation in K-Means Clustering, Pattern Recognition, vol. 45(3), pp. 1061-1075, 2012