|  |
| --- |
| Descubrimiento de información en textos |
| Tarea del tema 6: Clustering |
|  |
| Gabriel Vázquez Torres |
|  |

Descubrimiento de información en textos

Tarea del tema 6: Clustering

# Descripción de los datasets

Los dataset utilizados han sido re0.mat y re1.mat, que son los propuestos en el enunciado de este trabajo.

* El fichero re0.mat tiene: 1504 documentos, 2886 términos y 13 clases.
* El fichero re1.mat tiene: 1657 documentos, 3758 términos y 25 clases.

A continuación, se detallan las pruebas realizadas con distintos algoritmos, funciones de similitud y funciones de criterio a estos dos ficheros.

# Pruebas realizadas y análisis de los resultados

## He utilizado la versión del programa CLUTO para Linux, creando un script que generaba los ficheros .txt que me interesaban, acorde con la información que deseaba guardar para realizar su posterior estudio.

## El script se llama ejecutor.sh y contiene el siguiente código:

Sin estadísticas

#!/bin/bash

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i1' re0.mat 10 > r0‐agglo‐cos‐i1.txt

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i2' re0.mat 10 > r0‐agglo‐cos‐i2.txt

./vcluster ‐clmethod='graph' ‐sim='dist' ‐crfun='i1' re0.mat 10 > r0‐graph‐dist‐i1.txt

./vcluster ‐clmethod='graph' ‐sim='jacc' ‐crfun='i1' re0.mat 10 > r0‐graph‐jacc‐i1.txt

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i1' re1.mat 10 > r1‐agglo‐cos‐i1.txt

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i2' re1.mat 10 > r1‐agglo‐cos‐i2.txt

./vcluster ‐clmethod='graph' ‐sim='dist' ‐crfun='i1' re1.mat 10 > r1‐graph‐dist‐i1.txt

./vcluster ‐clmethod='graph' ‐sim='jacc' ‐crfun='i1' re1.mat 10 > r1‐graph‐jacc‐i1.txt

Con estadísticas

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i1' ‐rclassfile=estadisticas/re0.mat.rclass re0.mat 10 > r0‐

agglo‐cos‐i1‐EST.txt

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i2' ‐rclassfile=estadisticas/re0.mat.rclass re0.mat 10 > r0‐

agglo‐cos‐i2‐EST.txt

./vcluster ‐clmethod='graph' ‐sim='dist' ‐crfun='i1' ‐rclassfile=estadisticas/re0.mat.rclass re0.mat 10 >

r0‐graph‐dist‐i1‐EST.txt

./vcluster ‐clmethod='graph' ‐sim='jacc' ‐crfun='i1' ‐rclassfile=estadisticas/re0.mat.rclass re0.mat 10 >

r0‐graph‐jacc‐i1‐EST.txt

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i1' ‐rclassfile=estadisticas/re1.mat.rclass re1.mat 10 > r1‐

agglo‐cos‐i1‐EST.txt

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i2' ‐rclassfile=estadisticas/re1.mat.rclass re1.mat 10 > r1‐

agglo‐cos‐i2‐EST.txt

./vcluster ‐clmethod='graph' ‐sim='dist' ‐crfun='i1' ‐rclassfile=estadisticas/re1.mat.rclass re1.mat 10 >

r1‐graph‐dist‐i1‐EST.txt

./vcluster ‐clmethod='graph' ‐sim='jacc' ‐crfun='i1' ‐rclassfile=estadisticas/re1.mat.rclass re1.mat 10 >

r1‐graph‐jacc‐i1‐EST.txt

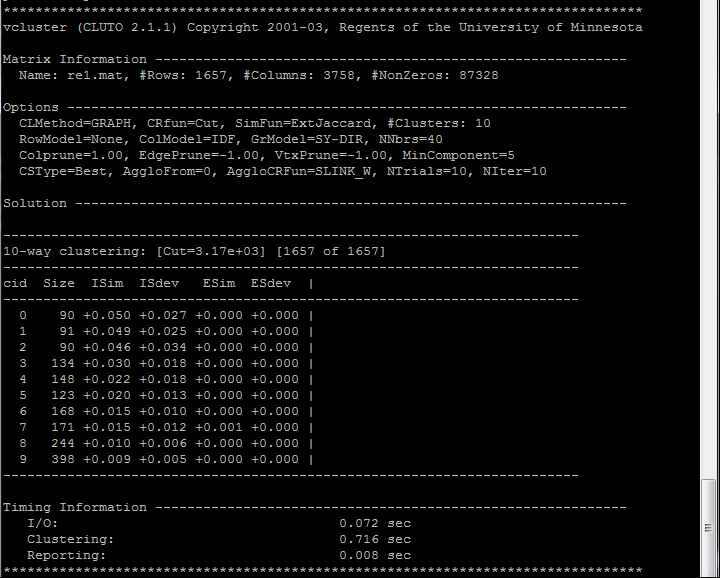
Siguiendo el orden del script, a continuación mostraré las salidas obtenidas y haré un breve análisis, explicando distintos elementos.

## VCluster aplicado al fichero re0.mat

## Con un algoritmo aglomerativo, función de similitud cosine y función de criterio i1

Línea de comando:

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i1' re0.mat 10



En primer lugar, comentar de forma muy breve el significado de cada columna:

 cid: Representa la id de cada cluster creado a partir del documento que hemos querido

empaquetar mediante esta técnica de clustering.

 Size: cantidad de objetos que contiene cada uno de los 10 clusters que hemos creado.

 ISim: Número que muestra la media de similitud entre los objetos del cluster.

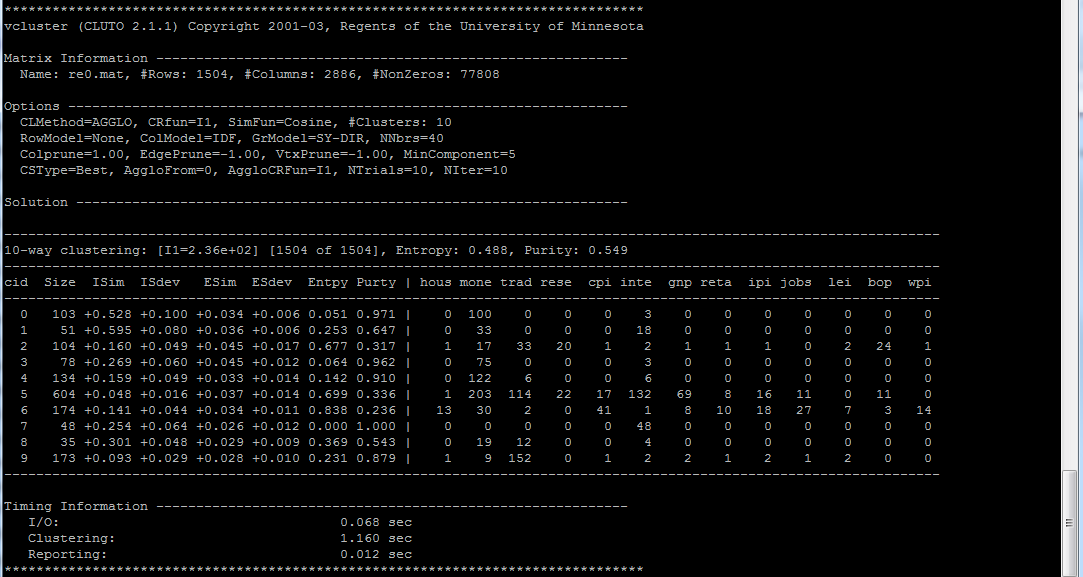
 ISdev: Media de la desviación de las similitudes entre objetos del cluster.

 ESim: Similitud de los objetos de cada cluster y del resto.

 ESdev: Desviación de las similitudes de los objetos de cada cluster y del resto.

Aplicando la siguiente línea de comando:

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i1' ‐rclassfile=estadisticas/re0.mat.rclass re0.mat 10



Se obtienen 2 columnas más de información:

 Entpy (Entropy): Muestra el índice de objetos de distinto tipo empaquetados en un

clúster.

 Purty (Purity): Nos da un valor que varía entre 0 y 1 en función de la similitud de la

clase de los objetos empaquetados en un clúster concreto.

Sabiendo esto, en este primer caso se puede destacar que:

‐ El clúster 7 realiza un empaquetado perfecto: posee una pureza de valor 1 debido a que

contiene 48 objetos que son del mismo tipo 'inte'.

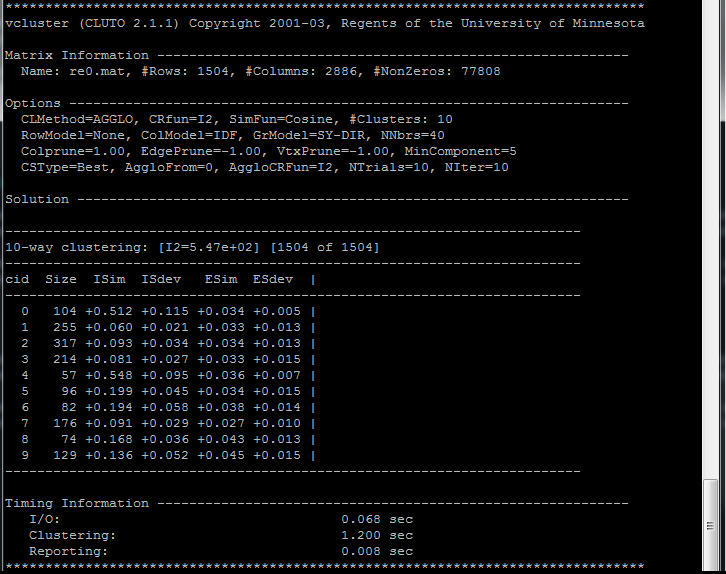
‐ Los clústers 0, 3 y 4 tienen también un alto nivel de pureza mientras que por ejemplo, el

clúster 6 tiene muchos paquetes de distinto tipo, con lo que la pureza es bastante pequeña.

## Con un algoritmo aglomerativo, función de similitud cosine y función de criterio i2

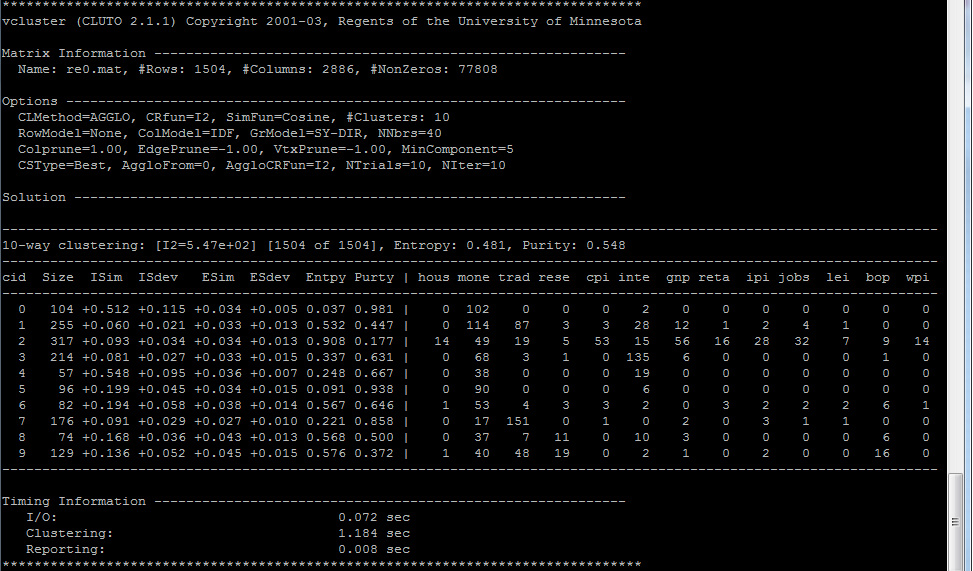
Línea de comando:

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i2' re0.mat 10



Aplicando la siguiente línea de comando:

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i2' ‐rclassfile=estadisticas/re0.mat.rclass re0.mat 10



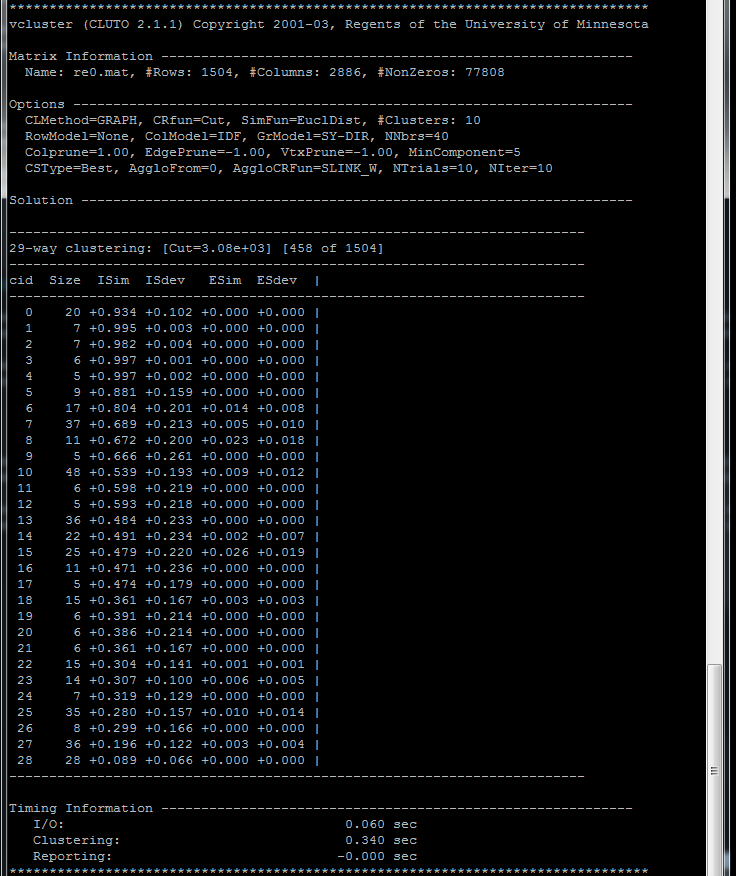
A diferencia de los parámetros usados en el punto anterior, aplicando estos criterios se observa que la Pureza general (que anteriormente era de 0.549) es prácticamente igual a la obtenida con esta variación, habiendo utilizado la función de criterio i2, que ha sido de 0.548.

Sin embargo, los clústers que se han formado son totalmente distintos, destacan el 0 y el 5 que tienen una pureza muy alta, pero sin embargo en el resto de clústers, en lugar de tener 4 clústers con una pureza alta y 6 con pureza algo más baja, tenemos una pureza más o menos acorde con la media general en cada clúster.

## Con un algoritmo de separación, función de similitud de distancia euclidea y función de criterio i1

Línea de comando:

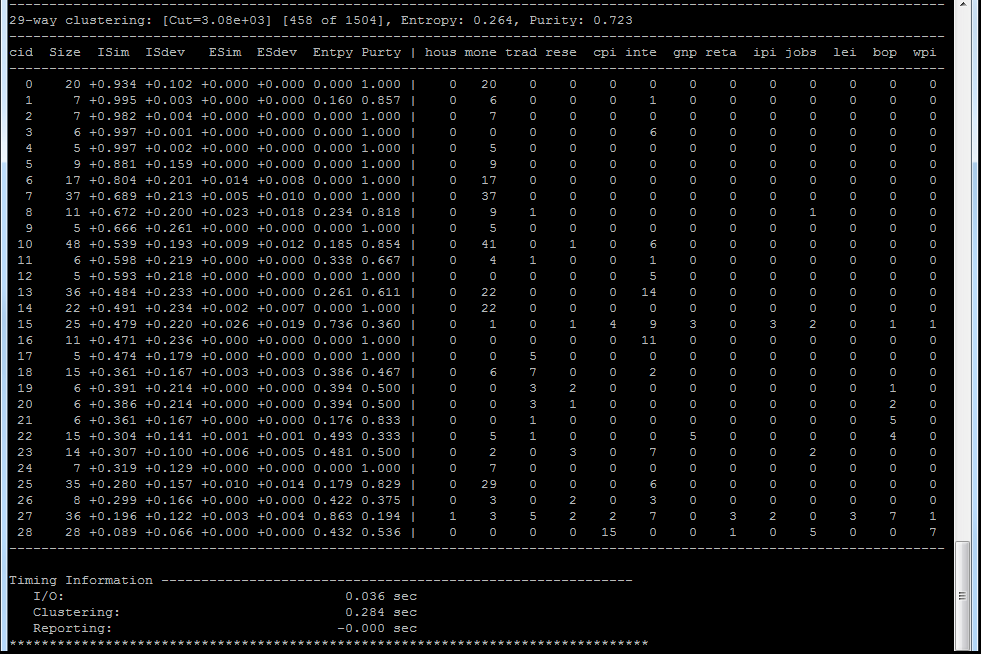
./vcluster ‐clmethod='graph' ‐sim='dist' ‐crfun='i1' re0.mat 10



Comando:

./vcluster ‐clmethod='graph' ‐sim='dist' ‐crfun='i1' ‐rclassfile=estadisticas/re0.mat.rclass

re0.mat 10



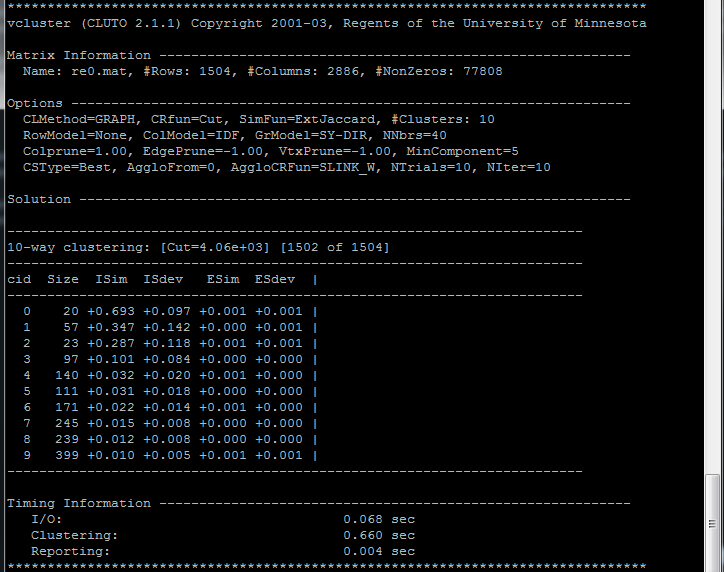
En esta aplicación, se observa que el programa genera un mayor número de clústers, con u menor número de objetos almacenados en cada uno, pero con una pureza bastante alta, la media es de 0.723, frente a 0.548 y 0.549 de la anterior aplicación.

Esto indica que siguiendo un algoritmo de separación y una función de selección mediante distancia euclidea, se obtiene un mejor empaquetado mediante clustering.

## Con un algoritmo de separación, función de similitud del coeficiente de Jaccard y función de criterio i1

Línea de comando:

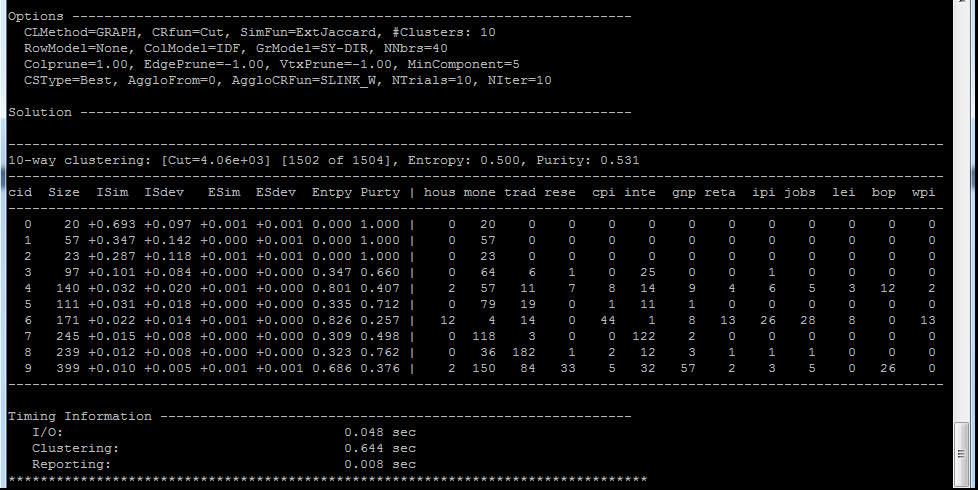
./vcluster ‐clmethod='graph' ‐sim='jacc' ‐crfun='i1' re0.mat 10



Comando:

./vcluster ‐clmethod='graph' ‐sim='jacc' ‐crfun='i1' ‐rclassfile=estadisticas/re0.mat.rclass

re0.mat 10



En esta aplicación, se observa que el programa genera los clústers con una pureza más alta y con un menor número de objetos insertados en los primeros clústeres, pero en los últimos clústers almacena un mayor número de objetos y además de distinto tipo, haciendo que la pureza general sea peor que las anteriores.

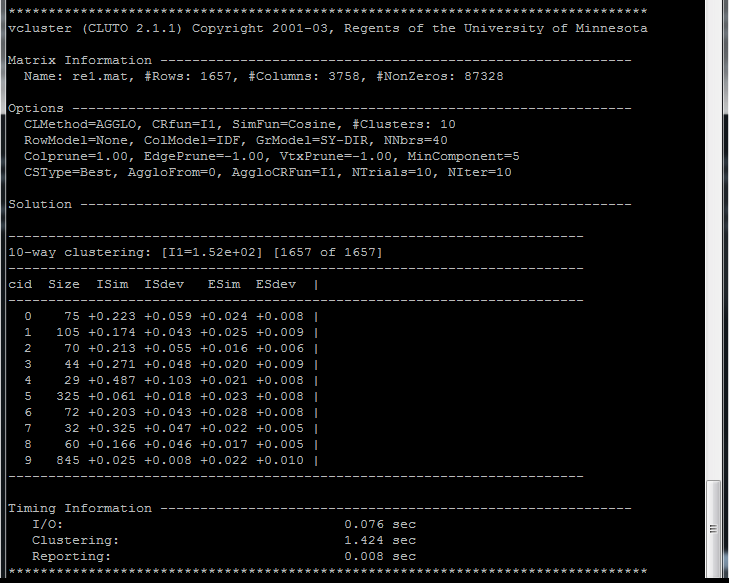
Esto es producido por el cambio en la función de similitud.

## VCluster aplicado al fichero re1.mat

## *Con un algoritmo aglomerativo, función de similitud cosine y función de criterio i1*

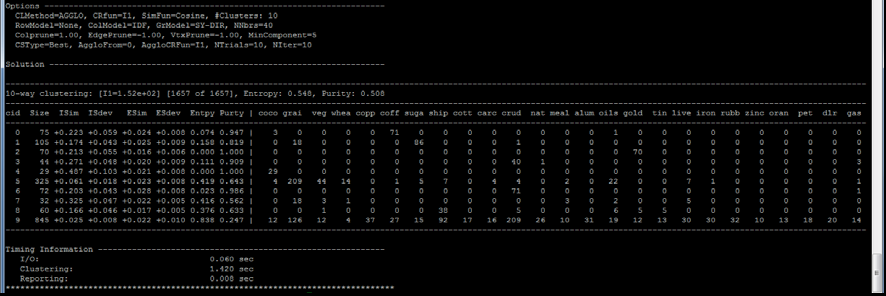
Línea de comando:

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i1' re1.mat 10



Aplicando la siguiente línea de comando:

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i1' ‐rclassfile=estadisticas/re1.mat.rclass re1.mat 10

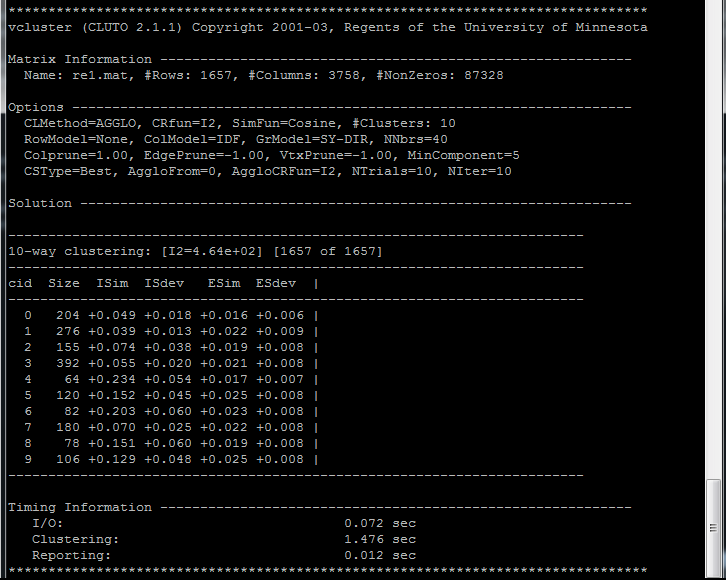


Tiene una pureza general de 0.508, es un algoritmo que consigue unas purezas medias un poco aleatorias, sin seguir un patrón fácil de explicar.

## Con un algoritmo aglomerativo, función de similitud cosine y función de criterio i2

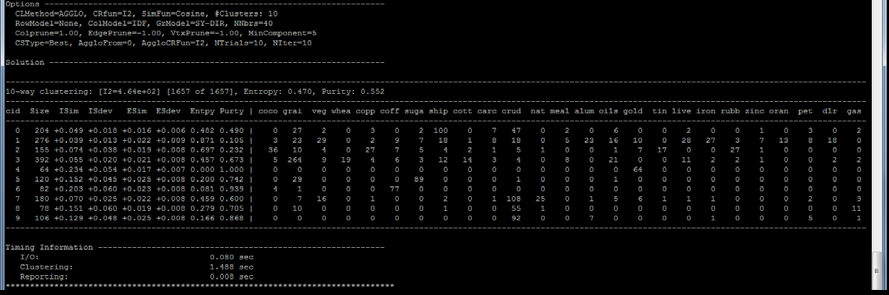
Línea de comando:

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i2' re1.mat 10



Aplicando la siguiente línea de comando:

./vcluster ‐clmethod='agglo' ‐sim='cos' ‐crfun='i2' ‐rclassfile=estadisticas/re1.mat.rclass re1.mat 10

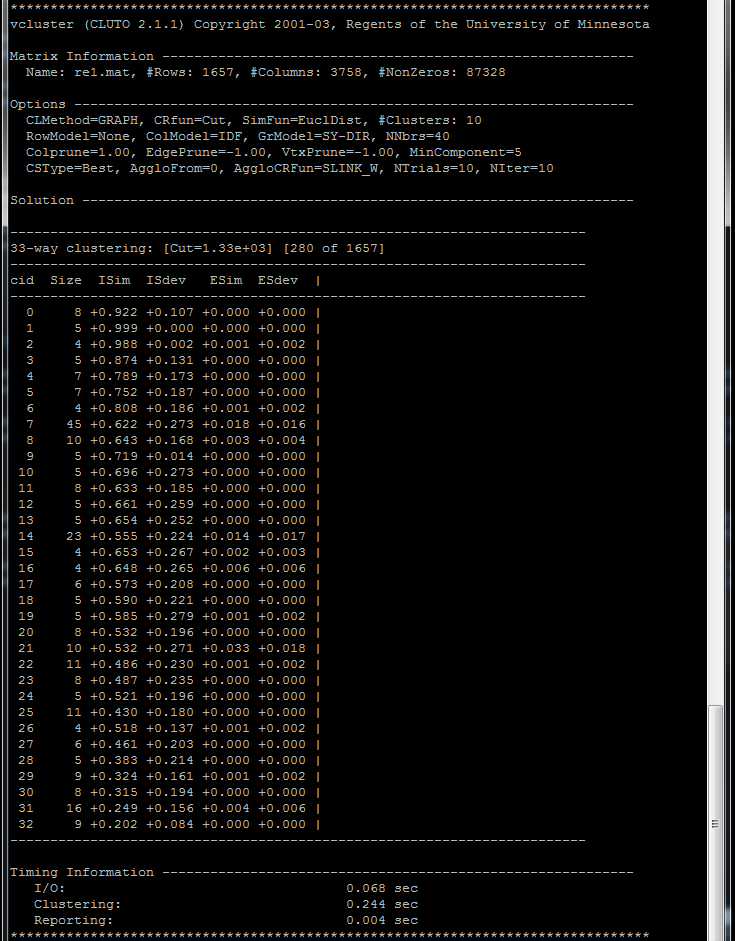


Consigue una mejor pureza general, pero no se nota mucha diferencia al haber cambiado únicamente la función de criterio de i1 a i2.

## Con un algoritmo de separación, función de similitud de distancia euclidea y función de criterio i1

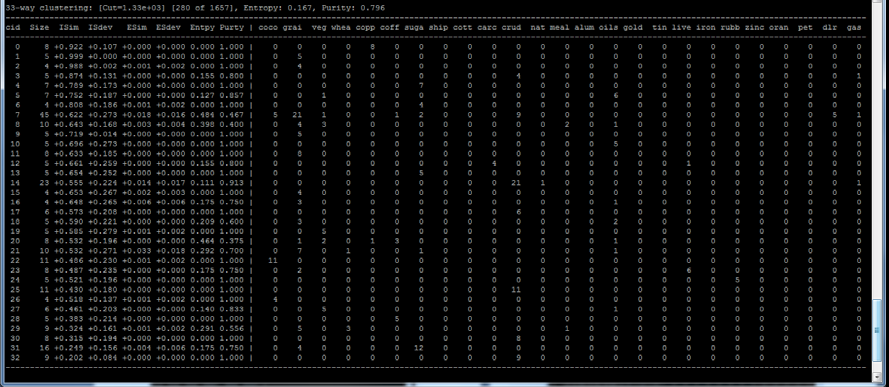
Línea de comando:

./vcluster ‐clmethod='graph' ‐sim='dist' ‐crfun='i1' re0.mat 10



Comando:

./vcluster ‐clmethod='graph' ‐sim='dist' ‐crfun='i1' ‐rclassfile=estadisticas/re1.mat.rclass re1.mat 10



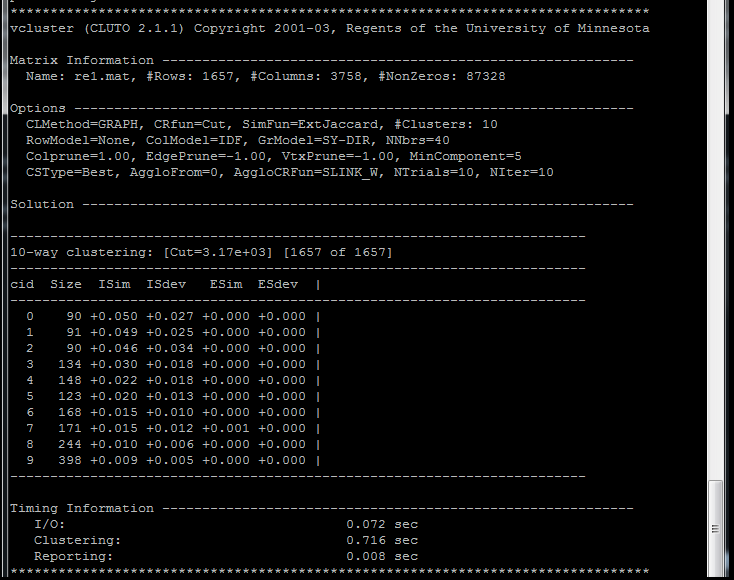
En esta aplicación, se observa que el programa genera un mayor número de clústers, con un menor número de objetos almacenados en cada uno, pero con una pureza bastante alta, la media es de 0.796.

Esto indica que siguiendo un algoritmo de separación y una función de selección mediante distancia euclídea, se obtiene un mejor empaquetado mediante clustering.

## Con un algoritmo de separación, función de similitud del coeficiente de Jaccard y función de criterio i1

Línea de comando:

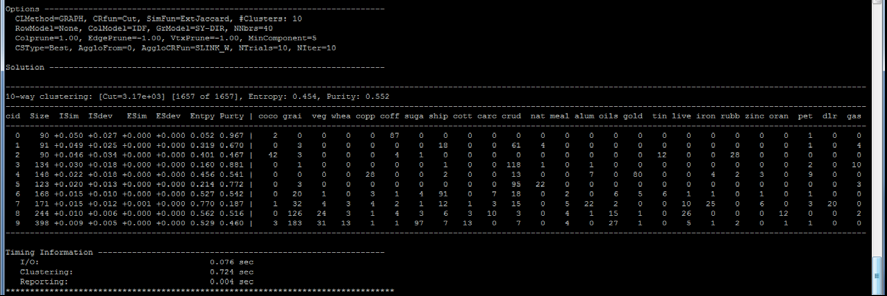
./vcluster ‐clmethod='graph' ‐sim='jacc' ‐crfun='i1' re0.mat 10



Comando:

./vcluster ‐clmethod='graph' ‐sim='jacc' ‐crfun='i1' ‐rclassfile=estadisticas/re1.mat.rclass

re1.mat 10



En esta aplicación, se observa que el programa genera los clústers con una pureza más alta y con un menor número de objetos insertados en los primeros clústeres, pero en los últimos clústers almacena un mayor número de objetos y además de distinto tipo, haciendo que la pureza general sea peor que las anteriores.

Esto es producido por el cambio en la función de similitud.

# Conclusiones generales sobre los distintos tipos de algoritmos a seguir.

Después de ver a las distintas salidas habiendo aplicado los distintos algoritmos, funciones de criterio y de selección, puedo observar lo siguiente:

‐ En la aplicación del algoritmo aglomerativo:

 No existe una gran diferencia de empaquetado por clustering utilizando la función de

criterio i1 o i2.

‐ En la aplicación del algoritmo de partición:

 Cuando se aplica la función de similitud por distancia euclídea, se crea un mayor

número de clusters y se consigue tener un mejor resultado.

 Al aplicar la función de similitud por el coeficiente de Jaccard, se puede observar

perfectamente cómo los primeros clusters tienen una mayor pureza y un menor

número de objetos en su interior, pero casi todos del mismo tipo. Sin embargo, los

clusters creados al final, tienen un mayor número de objetos dentro y un menor

coeficiente de pureza.

‐ En general: el que mejor resultado ha dado sin duda es el algoritmo de partición con la

aplicación de la función de similitud por distancia euclídea.