

Universidad Andrés Bello

Facultad de Ingeniería

Ingeniería Civil Informática

Tarea3

Aprendizaje No-Supervisado

Sistemas Inteligentes

David Andrés Molina Garrido

Profesor Alejandro Figueroa

Ayudante Jean Contreras

Santiago - Chile.

Octubre, 2017.

**Introducción**

El siguiente informe consta de aprender los conceptos relacionados con aprendizaje no-supervisado tanto los modelos, métricas de evaluación, y la metodología de trabajo. Para esto, utilizaremos dos métodos de agrupamiento de datos, K-MEANS y DBSCAN.

El set de datos que se entrega a los algoritmos de agrupamiento (clustering) se obtiene de un conjunto de perfiles de usuarios provenientes de plataformas (cQA), los cuales fueron etiquetadas según las siguientes clases:

-**USA only** (Si está relacionado con sólo lugares dentro de USA).

-**Non-USA** (Si está relacionado con sólo lugares fuera de USA).

-**World** (Si está relacionado con lugares dentro y fuera de USA).

-**Undetermined** (Si no provee ninguna pista de algún lugar geográfico relacionado al usuario).

Posteriormente, estos perfiles son representadas por un vector utilizando las palabras contenidas en ese perfil como los atributos de este vector (como vimos en la Tarea2).

A diferencia de las Tareas anteriores nuestros datos no tendrán como una variable dependiente la etiqueta que anteriormente le dimos a cada una de nuestras instancias, sino que todas las instancias tendrán el mismo status.

Lo que se intenta obtener con K-MEANS es encontrar grupos de observaciones (*clusters*) con características semejantes y éstos ser diferentes a los otros grupos o *clusters*. Técnicamente se intenta maximizar la variación inter-*cluster* y minimizar la variación intra-*cluster*.

Para el caso de DBSCAN (*Density-based spatial clustering of applications with noise*), lo que se intenta encontrar son grupos diferenciados por medio de la densidad de éstos, así como también ruido, datos que no se asemejan a ningún grupo. Para esto, DBSCAN ocupa dos parámetros: un radio ε , para poder revisar la vecindad a un punto específico; y una cantidad de puntos mínimos, que depende del radio ε.

**Descripción del Problema**

Primeramente, para generar la salida de K-MEANS utilizaremos los datos representados vectorialmente en la Tarea 2, éstos se utilizan para generar los *clusters* que, a su vez, estarán determinados por 4 centroides que representan nuestras 4 clases. Se utilizará la implementación de *Clustering* K-MEANS de Yakmo.

Los centroides son la media de todos los puntos de cada *cluster*. Al comienzo, se elegirán k centroides que se irán recalculando, en nuestro caso k=4 , ya que el objetivo es que cada *cluster* represente una clase. En el siguiente ejemplo con k=2 vemos el modo de reasignación de los centroide en sus *clusters*.

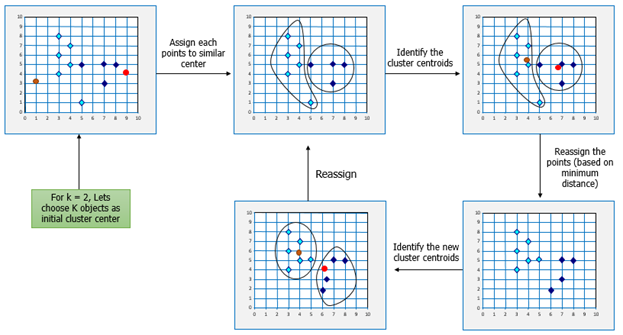


Imagen1: Ejemplo reasignación de un centroide.

La reasignación se basa en las medias de todos los puntos de cada *cluster*, La distancia desde un punto (instancia) xj al centroide mj se calcula:

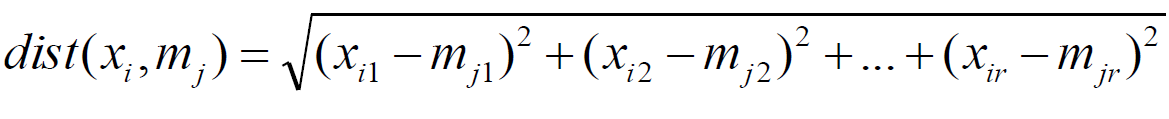


Imagen2: Distancia de un punto al centroide.

Para generar la representación de agrupamiento en DBSCAN se utilizará un algoritmo especifico, utilizando una función de distancia euclidiana.

Se le entrega nuestro set de datos, para posteriormente designar un radio ε , con el cual poder revisar la vecindad a un punto específico; y una cantidad de puntos mínimos, que depende del radio ε, para poder incluirlos dentro de un *cluster*.

Este método recursivo se representa en la siguiente imagen:

En el ejemplo ε = 1 y los puntos mínimos de la vecindad = 4 .

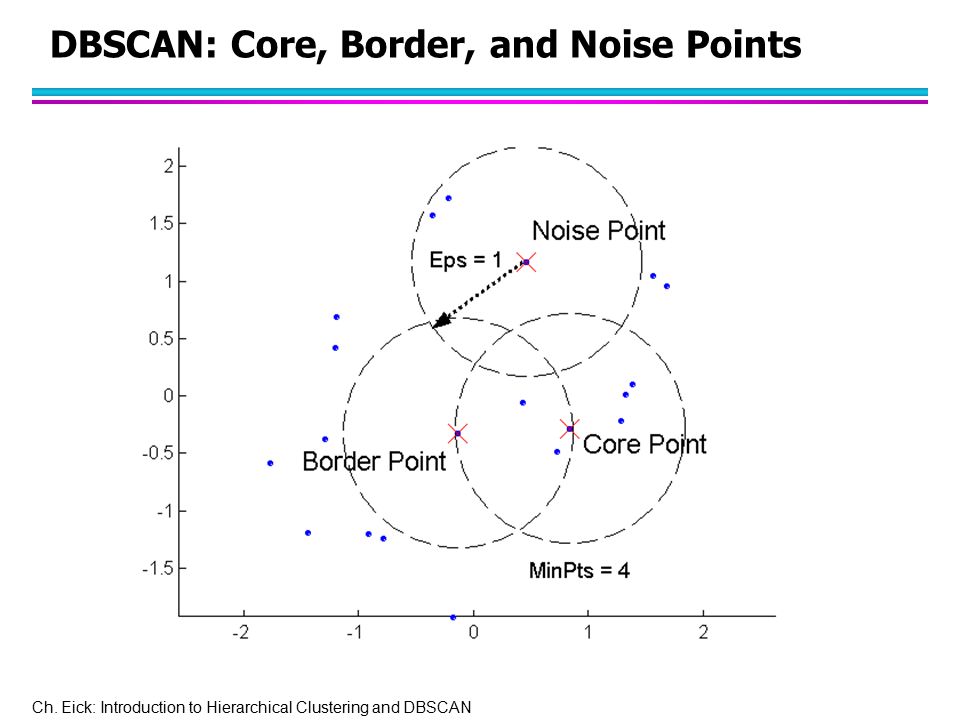
****

Imagen2: Variables DBSCAN .

**Nota:** DBSCAN nofuncionó como debería, me arrojaba datos sin sentidos con una implementación y con la implementación en WEKA me arrojó error probando con diversos parámetros, se intentó buscar la solución, sin éxito.

**Análisis del Resultado**

Para que en las distintas clases de las preguntas sea viable su modelamiento, se le entregó un valor numérico que las represente, estos fueron:

|  |  |
| --- | --- |
| **Clase** | **Número** |
| Non-USA | 1 |
| USA only | 2 |
| Undetermined | 3 |
| World | 4 |

Imagen3: Tabla de clase y su número asignado.

A través de la implementación de *Clustering* K-MEANS de Yakmo se les entregó nuestro set de datos, para luego asignarle una clase en particular para cada *cluster*.

Se asignó para cada *cluster* un valor dentro de las 4 clases, esto para que cada *cluster* se represente como una clase y sea posible el cálculo de métricas posteriormente. Por lo tanto, a cada *cluster* se le asignó el valor más representativo, ósea, el que más apareció en ese *cluster*, los datos son los siguientes:

|  |  |  |
| --- | --- | --- |
| Clases | Nº Clase | Cluster |
| Non-USA | 1 | C0 |
| USA-Only | 2 | C1 |
| Undetermined | 3 | C2 |
| World | 4 | C3 |

Imagen4: Tabla de clase por cada *cluster.*

Como habíamos visto en la Tarea 1 la distribución de nuestros perfiles por clase (*Imagen5*) tuvo a **USA Only** y **World** con la mayor y menor probabilidad respectivamente. En el caso de **USA Only** al ser la clase que más instancias tiene, nos encontramos que en tres *cluster* fue la clase mayoritaria. Por lo tanto, se procedió a desempatar, asignándole la segunda clase mayoritaria a los *clusters (C0 y C3)*, esto porque la clase mayor ya ha sido usada anteriormente.

|  |  |  |  |
| --- | --- | --- | --- |
| Clases | Nº Clase | Total por Clase | Probabilidad |
| Non-USA | 1 | 579 | 0,2895 |
| USA only | 2 | 817 | 0,4085 |
| Undetermined | 3 | 527 | 0,2635 |
| World | 4 | 77 | 0,0385 |
| Total |  | 2000 | 1 |

Imagen5: Distribución de preguntas.

El detalle de cada *Cluster* según la clase y la cantidad es el siguiente:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| C0 |  |  |  |  | C1 |  |  |
| Clases | Cantidad | Clase Mayoritaria |  |  | Clases | Cantidad | Clase Mayoritaria |
| 1 | 184 | 1 |  |  | 1 | 338 | 2 |
| 2 | 204 |  |  |  | 2 | 519 |  |
| 3 | 147 |  |  |  | 3 | 273 |  |
| 4 | 23 |  |  |  | 4 | 43 |  |
|  |  |  |  |  |  |  |  |
| C2 |  |  |  |  | C3 |  |  |
| Clases | Cantidad | Clase Mayoritaria |  |  | Clases | Cantidad | Clase Mayoritaria |
| 1 | 10 | 3 |  |  | 1 | 47 | 4 |
| 2 | 10 |  |  |  | 2 | 84 |  |
| 3 | 31 |  |  |  | 3 | 76 |  |
| 4 | 2 |  |  |  | 4 | 9 |  |

Imagen6: Instancias por Clusters Yakmo.

Los 10 datos más representativos de todos los conjuntos de *clusters* fueron los siguientes:

En orden de más cercano a su centroide, con la etiqueta respectiva.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Posición | C0 | C1 | C2 | C3 |
| 1º | 2 | 2 | 3 | 4 |
| 2º | 1 | 2 | 3 | 2 |
| 3º | 1 | 2 | 3 | 4 |
| 4º | 2 | 1 | 4 | 2 |
| 5º | 1 | 4 | 3 | 3 |
| 6º | 3 | 2 | 3 | 3 |
| 7º | 2 | 2 | 1 | 2 |
| 8º | 2 | 3 | 3 | 1 |
| 9º | 2 | 2 | 3 | 2 |
| 10º | 1 | 2 | 1 | 3 |

Imagen7: 10 Instancias más representativas.

Como vemos, las 10 instancias más representativas de cada *cluster* son de cierta manera representativas de la clase mayoritaria que habíamos obtenido, en el *cluster* 1 y 2 se ve que concuerdan de mayor manera en contraste con el *cluster* 0 y 4. Esto a su vez se corrobora ya que fueron los únicos dos *clusters* que su clase mayoritaria fue efectiva (no se tuvo que hacer un desempate) en contraparte con los otros dos *cluster* (0 y 4) que se adecuó una clase para hacer un desempate. con los estuvieron solo en 3 *clusters*: C2, C4 y C6.

Esto se entiende ya que, según el análisis en la Tarea1, la clase 2(USA Only) y 3(Undetermined) pertenecientes (mayoritariamente) a los *clusters* C1 y C2 respectivamente, fueron las clases donde más fácil se lograba identificar a ellas, esto según la misma facilidad en el etiquetado, en el caso de USA Only ya que eran frecuentes las ciudades y repetitivas, y en el caso de Undetermined no poseían un lugar (un atributo) característico.

Como vemos en la *Imagen8* de los *clusters* que se generaron C1 y C2 fueron los que mayormente se diferenciaron respecto a los otros, ósea, se maximizó la variación Inter-*Cluster* y minimizó la Intra-*Cluster*.

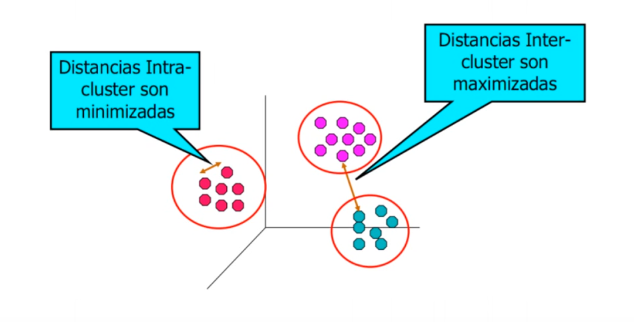


Imagen8: Distancias *Clusters*.

Métricas

Nuestros datos con la implementación en Yakmo, representados en una matriz de confusión:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | C0 | C1 | C2 | C3 |
|  | Prediccion / Clase | | | |
| Real | 1 | 2 | 3 | 4 |
| 1 | 184 | 338 | 10 | 47 |
| 2 | 204 | 519 | 10 | 84 |
| 3 | 147 | 273 | 31 | 76 |
| 4 | 23 | 43 | 2 | 9 |
| Total por Clústers | 558 | 1173 | 53 | 216 |
| Prob por Clústers | 0,279 | 0,5865 | 0,0265 | 0,108 |
| Total | 2000 |  |  |  |

Imagen 11: Matriz de confusión de datos por *clusters*.

La matriz de confusión representa en las columnas nuestras clases mayoritarias pertenecientes a un *cluster*; y en las filas, la clase a la cual pertenece esa distribución de datos.

Como se muestra en la *Imagen11* cada *clusters* no es fiel representante de su clase, existen más clases en un mismo *clusters*, además no siempre se representa la clase mayoritaria ya que ésta es mayoritaria de igual manera en otros *clusters*.

Es por esto, que existe un alto porcentaje de error en el agrupamiento que se genera, pero ese error está relacionado a nuestro etiquetado, es posible que según nuestros datos y atributos de las instancias el método K-MEANS lo haya particionado de buena manera.

Las métricas de cada *cluster* con su respectiva clase mayoritaria fueron:

|  |  |  |  |
| --- | --- | --- | --- |
| Clase 1 Cluster C0 |  |  |  |
| Precision | 0,3297491 | Purity | 0,329749104 |
| Recall | 0,31778929 | Entropy | 0,513823969 |
| f1-score | 0,32365875 |  |  |
|  |  |  |  |
|  |  |  |  |
| Clase 2 Cluster C1 |  |  |  |
| Precision | 0,44245524 | Purity | 0,442455243 |
| Recall | 0,63525092 | Entropy | 0,451485933 |
| f1-score | 0,52160804 |  |  |
|  |  |  |  |
|  |  |  |  |
| Clase 3 Cluster C2 |  |  |  |
| Precision | 0,58490566 | Purity | 0,58490566 |
| Recall | 0,05882353 | Entropy | 0,138803391 |
| f1-score | 0,10689655 |  |  |
|  |  |  |  |
|  |  |  |  |
| Clase 4 Cluster C3 |  |  |  |
| Precision | 0,04166667 | Purity | 0,041666667 |
| Recall | 0,11688312 | Entropy | 0,346776853 |
| f1-score | 0,06143345 |  |  |

Imagen 12: Métricas por *clusters*.

Como vemos, el *Cluster* 2 que representa a la clase 3 (Undetermined) tuvo los mayores índices, esto en relación con lo que logramos concluir de la Tarea 2:

1.Undetermined fue una clase bien característica al no poseer como atributos las ciudades-paises, no así como las otras clases.

2.Posee un alto porcentaje de instancias.

Por otro lado, los demás *cluster* tuvieron muchas más instancias que el cluster 2, pero su distribución estuvo más “repartida” entre las clases. es por esto, sus bajas *Precision* y *Recall*, que corresponden a que tienen muchos falsos positivos y falsos negativos.

El *Purity* y Entropía total de los *clusters* fue el siguiente:

|  |  |
| --- | --- |
| Purity Total | 0,34969417 |
| Entropy Total | 1,45089015 |

Imagen 13: *Purity* y *Entropy* totales.

El purity total se obtuvo de la sumatoria de dividir el total de instancias de un *cluster* i por el total de datos (2000), multiplicado por el *purity* del *cluster* i.

El *purity* total nos dice que existe poca pureza en ellos, que se corresponde y se entiende de manera específica en la *precision* y *recall*. Esta poca pureza se basa en que cada *cluster* no toma de manera regular su clase mayoritaria, sino que varía con otras clases.

Por otro lado, para la entropía se utilizó logaritmo en base 2 para cada *cluster*, con el propósito de hacerla más entendible. La entropía total nos dice que el grado de certeza de los *clusters* con respecto a su clase mayoritaria es “regular”, casi ½ de certeza en la totalidad de *clusters*, además vemos que la entropía que nos arrojó es bastante similar (1.73) a la que sacamos luego de obtener los datos después del etiquetado.

**Conclusión**

Como conclusión podemos ver que el método de agrupamiento en cluster nos indica de manera más visible la información contenida en nuestro set de datos y los agrupa de manera de encontrar semejanzas en ellos.

Como habíamos visto en K-MEANS la información resultante tanto de la distribución de instancias etiquetadas y el particionamiento de Yakmo nos arroja datos parecidos, pero sólo en algunos *clusters* (C1 y C2) , y esto también porque sus clases mayoritarias (USA Only y Undetermined) fueron las que en el set de datos más instancias tuvieron, y por lo tanto, mejor entrenamiento se puso obtener a partir de éstas.

En la implementación con Yakmo, en particular el cluster 2, tuvo los mejores índices, teniendo en cuenta que fue un cluster que no agrupó la mayor cantidad de datos, con un porcentaje de 0,0265, por lo que se vio que la tendencia fue muy fuerte en ese cluster, reuniendo características específicas y diferenciadas del resto.

Por otra parte, se deduce que los atributos como palabras quizá no es la mejor manera de representar vectorialmente las instancias, esto ya que utilizando métodos de agrupamiento no se reflejan de la mejor manera las diferencias entre una y otra.