Algoritmo de Clustering Basado en el Concepto de Densidad Atómica

Oscar Andrés Monroy Medina[1],

[1] Facultad de Ingeniería, Universidad Austral,
Cerrito 1250, Ciudad Autonoma de Buenos Aires, Argentina
{oscar.andres.monroy}@gmail.com

Resumen, El análisis de clúster o Clustering agrupa un conjunto de objetos de datos en clústers o grupos de manera que en cada grupo los objetos sean similares entre si y disimiles de los objetos de otros grupos. En la actualidad, existen distintas técnicas de agrupamiento que permiten cumplir con esta tarea. En búsqueda de un algoritmo más natural se hizo uso del concepto de densidad atómica de los elementos como base para generar uno nuevo. El algoritmo propuesto tiene como ventajas, poseer un método concreto de selección de centroides, además de tener mejores agrupamientos que otros algoritmos basados en centroides como k-means y k-medoids.

Palabras Clave: Data Mining – Clúster – Clustering - Densidad Atómica - Matriz de distancia

1 Introducción

En algunos problemas de data mining llega a ser necesario agrupar *dataset*[1] en diferentes grupos o clases para determinar una acción a ejecutar o para optimizar un algoritmo de clasificación o predicción. El proceso de particionar un conjunto de datos en un conjunto de sub-clases (o grupos) significativas se llama Clustering [1].

En la actualidad existen diferentes algoritmos de clustering que dependiendo de la orientación pueden dar un resultado muy diferente de otro [1, 4, 5, 8, 9], sin embargo, no se puede decir que los clústers resultantes dan una respuesta "errónea", simplemente un algoritmo de clustering se adapta mejor que otro para un problema determinado, esto da a lugar a la posibilidad de crear nuevos algoritmos para determinados problemas.

En búsqueda de un algoritmo más natural se ha establecido una agrupación basada de un estudio que lleva más de un siglo de investigación, la tabla periódica de los elementos. Una de las propiedades que tienen los elementos en la tabla periódica es la densidad atómica que posee diversas aplicaciones como se puede ver [2, 3].

[1] Un *dataset* está definido como un conjunto de datos, termino usualmente usado en minería de datos.

Observando las distintas aplicaciones que posee la densidad atómica, se podría usar el cálculo definido por la teoría [13] aplicado a un espacio n-dimensional (donde *n* es la cantidad de variables del *dataset*) para agrupar los puntos en clústers donde la densidad sea la máxima.

Para definir la fórmula de la densidad sobre un *dataset* de *n* variables se usará la definición inicial dada por [13].

$$d = \frac{m}{v} \tag{1}$$

Donde *m* es la masa y *v* el volumen de la hiperesfera definido en [14]. Para aplicarlo a un *dataset* dado se definirá la masa y el volumen como:

$$m_c(r) = |\{x | x \in D \wedge \|c - x\| \leq r\}| \tag{2}$$

$$V_n(r) = \frac{\pi^{\frac{n}{2}}}{\binom{n}{2}} r^n \tag{3}$$

Donde $\| \quad \|$ es la distancia euclidiana, *r* es el radio de la hiperesfera, *c* el centro, *n* el número de dimensiones y *D* el *dataset*. Según [13] esta fórmula de volumen es influenciada fuertemente por el número de dimensiones, es decir que en un *dataset* con una gran cantidad de variables el volumen tiende a 0. Es por eso que se utilizará el volumen del hipercubo que contiene la hiperesfera para el cálculo de la dimensión.

$$V_n(r) = (2r)^n \tag{4}$$

Luego la fórmula de la densidad para los *dataset* estaría definida como:

$$d_n(c; r) = \frac{m_c(r)}{(2r)^n} \tag{5}$$

2 El conjunto de datos y el algoritmo.

2.1 Extracción del dataset

Para el estudio de un *dataset* real se obtuvo la información de atletas de crossfit y sus resultados para la clasificación de los regionales (el open) de los años 2011, 2012, 2013 y 2014, esta información se encuentra disponible en http://games.crossfit.com/leaderboard. Para esta información se usó un *scraper*[2] llamado scrapy y se guardaron los datos en una base de datos noSQL llamada mongoDB. Una vez obtenidos los datos se unificaron en un csv. El resultado luego de un proceso de extracción, transformación y limpieza(ETL) da un *dataset* de 6153 elementos y 34 variables. Una descripción de las variables se puede ver en la Tabla 1.

Tabla 1. Descripción del *dataset*.

Variable	Tipo	Descripción
Year	Entero	Año de competencia
X	Decimal	Coordenada del lugar de entrenamiento
Y	Decimal	Coordenada del lugar de entrenamiento
How long have you been doing crossfit	Entero	Meses de entrenamiento
Weight	Decimal	Peso del atleta
Clean and jerk	Decimal	Máximo peso que el atleta puede levantar en clea&jerk
filthy 50	Entero	Tiempo en segundos del conjunto de ejercicios filthy 50
I eat whatever is convenient	Binario	1 verdadero /0 falso relativo a la dieta del atleta
I weigh and measure my food	Binario	1 verdadero /0 falso relativo a la dieta del atleta
I eat strict paleo	Binario	1 verdadero /0 falso relativo a la dieta del atleta
I eat 13 full cheat meals per week	Binario	1 verdadero /0 falso relativo a la dieta del atleta
Sprint 400m	Entero	Tiempo que tarda el atleta para correr 400 metros
Back squat	Decimal	Máximo peso que el atleta puede levantar en back squat
Fran	Entero	Tiempo en segundos del conjunto de ejercicios fran
Fight gone bad	Entero	Tiempo en segundos del conjunto de ejercicios fight gone bad
Snatch	Decimal	Máximo peso que el atleta puede levantar en snatch
Max pull ups	Entero	Máxima cantidad de repeticiones seguidas que el atleta puede hacer del ejercicio pull up
Helen	Entero	Tiempo en segundos del conjunto de ejercicios helen
Grace	Entero	Tiempo en segundos del conjunto de ejercicios grace
Gender	Binario	Genero del atleta 0 mujer/1 hombre
Age	Entero	Edad
Deadlift	Decimal	Máximo peso que el atleta puede levantar en deadlift
I have completed the crossfit level 1 certificate	Binario	1 verdadero /0 falso relativo a la formación del atleta
I have attended one or more specialty courses	Binario	1 verdadero /0 falso relativo a la formación del atleta
I have had a life changing experience due to crossfit	Binario	1 verdadero /0 falso relativo a la formación del atleta
I train other people	Binario	1 verdadero /0 falso relativo a la formación del atleta
I began crossfit with a coach eg at an affiliate	Binario	1 verdadero /0 falso relativo a la formación del atleta
I strictly Schedule my rest days	Binario	1 verdadero /0 falso relativo a la formación del atleta
I do multiple workouts in a day 2x a week	Binario	1 verdadero /0 falso relativo a la formación del atleta

[2] Un *scraper* es una herramienta utilizada para obtener datos de una fuente externa, como puede ser una página web.

I usually only do 1 workout a day	Binario	1 verdadero /0 falso relativo a la formación del atleta
I typically rest 4 or more days per month	Binario	1 verdadero /0 falso relativo a la formación del atleta
I typically rest fewer than 4 days per month	Binario	1 verdadero /0 falso relativo a la formación del atleta
I do multiple workouts in a day 1x a week	Binario	1 verdadero /0 falso relativo a la formación del atleta
Rank	Entero	Posición que obtuvo el atleta para esta competencia

2.2 Desarrollo del Algoritmo

Antes de escribir el algoritmo es necesario definir un conjunto más.

$$Den_n(D) = \{d_n(c;r) | c \in D \wedge r \in R^+\} \quad (6)$$

En otras palabras (6) sería el conjunto de todas las densidades posibles para el conjunto *D*. Una descripción del algoritmo de clustering sería el siguiente:

```
Procedimiento
      P:=D;
      C:={};
      mientras(|P|>0)
            Encontrar c y r tal que d_n(c;r) = max(Den_n(P));
            Cluster:={x ∈ P|||x − c|| ≤ r};
            C:=C union {Cluster};
            P:=P-Cluster;
      fin mientras
      retornar C
fin
```

2.3 La matriz de distancias

El algoritmo descrito en 2.1 parece algo sencillo pero el cálculo de $Den_n(D)$ para cada $r \in R^+$ es algo imposible de calcular. Por eso es necesario limitar los valores de r a un conjunto finito de valores. Se dirá entonces, que todos los posibles valores que puede tomar r serían todas las distancias entre los puntos de D, es decir todas las distancias que hacen parte de la matriz de distancias de *D*. Es decir que para un elemento $c \in D$ los posibles valores de *r* serían los valores del vector de distancias de *c*, una descripción de este conjunto se puede ver en (7).

$$M_D(c) = \{r | r = \|x - c\|, \forall x \in D\} \quad (7)$$

$$M_D = \cup_{c \in D} M_D(c) \quad (8)$$

De esta manera se puede limitar los valores de (6) para replantearlo de la siguiente manera:

$$Den_n(D) = \{d_n(c;r) | c \in D \wedge r \in M_D(c)\} \quad \textbf{(9)}$$

Ahora $Den_n(D)$ se convierte en un conjunto finito. Sin embargo, esta definición deja un problema alto de complejidad, dado que para hallar M_D se tiene que realizar un total de $|D|^2$ calculos. En el caso particular de este *dataset* el valor sería de 6153^2, es decir $37'859.409$ calculos de distancia. Por supuesto, aplicando algunas propiedades de distancias como (10) y (11) se pueden reducir los cálculos a menos de la mitad.

$$\forall x, y \; \|x - y\| = \|y - x\| \quad \textbf{(10)}$$

$$\|x - x\| = 0 \quad \textbf{(11)}$$

El total de cálculos se reduciría a (12) que sería el total de cálculos, restando los cálculos que dan 0 y los cálculos que se repetirían.

$$\frac{|D|^2 - |D|}{2} = 18'926.628 \quad \textbf{(12)}$$

Se reducen los cálculos en gran cantidad, sin embargo, aún sigue siendo demasiado costoso. Es por eso que sería necesario realizar un algoritmo tomando muestras representativas y evitando así tener que calcular toda la matriz de distancia.

2.4 Algoritmo por muestras aleatorias

El algoritmo aplicado a muestras aleatorias sería muy similar al anterior

```
Procedimiento
      P:=D;
      C:={};
      mientras(|P|>0)
            si n < |P| entonces:
                  S:=muestra aleatoria de tamaño n;
            Si no:
                  S:=P;
            Encontrar c y r tal que d_n(c;r) = max(Den_n(S));
            Cluster:={x ∈ P|||x − c|| ≤ r};
            C:=C union {Cluster};
            P:=P-Cluster;
      fin mientras
      retornar C
fin
```

3 Los resultados

Finalmente, los dos algoritmos dieron sus resultados. Para la comparación se hará uso de la métrica SSE (Error de suma de cuadrados) la cual se puede aplicar a clusters que son basados en centroides. Además de comparar los dos algoritmos (el algoritmo que usa el *dataset* total y el algoritmo que usa muestras aleatorias) se compararan también con el mismo algoritmo pero tomando el centroide aleatoriamente, y con los algoritmos k-means y k-medoids implementados en rapid miner (https://rapidminer.com/).

3.1 Calculo del SSE

El SSE representa que tan próximos están los elementos de un clúster a su centroide. Clústers bien agrupados tienden a tener un SSE más bajo. El cálculo del SSE en un conjunto de clúster sería la descrita en (13).

$$SSE(C) = \sum_{C_x \in C} \sum_{y \in C_x} \|x - y\| \quad (13)$$

Donde C es el conjunto de todos los clusters y C_x representa un cluster con centro en x.

3.2 Comparación de los SSE

Finalmente se compararon los resultados de SSE entre los algoritmos de densidad atómica usando todo el *dataset*, de densidad atómica usando muestras aleatorias, densidad atómica tomando centroide aleatorio, k-means y k-medoids. Para k-means y k-medoids se realizaron estudios con k igual a el número de clusters resultantes del algoritmo de densidad atómica usando todo el *dataset* y con un máximo número de iteraciones de 100. Los resultados se ven resumidos en la tabla 2.

Tabla 2. Comparación de los valores SSE para los distintos métodos de clustering.

Método Utilizado	SSE	Diferencia %
Densidad Atómica Total	26091428958,495	0
Densidad Atómica 10%	26357521896,261	1.02
Densidad Atómica aleatoria	146454233565,81	461,3116622
k-means	$6.4829711950049 * 10^{18}$	24847129604
k-medoid	$5,7617455041804 * 10^{25}$	$2,20829 * 10^{17}$

Como se puede ver en la tabla 1, las diferencias entre los valores de los SSE son importantes, al usar conjuntos aleatorios de 10% de tamaño del *dataset* total da una diferencia del 1.02% de perdida de SSE con respecto al algoritmo que usa el total del conjunto. Mientras que hay una diferencia significativa si se toma como centroide un elemento aleatorio, perdiendo más del 461% del SSE posible, estos algoritmos a pesar

de perder grandes porcentajes de SSE siguen siendo mucho mejores que k-means y k-medoid generados con el mismo número de clusters.

4 Conclusiones

Como conclusión se puede decir que el algoritmo de clustering basado en la densidad atómica puede obtener excelentes resultados, sin embargo, existe un costo computacional importante ya que depende del cálculo de todas las distancias del centroide. Por otro lado, al igual que k-means la selección de centroides es muy importante, como se pudo ver en los resultados, se obtuvo una baja de más del 461% en el cálculo del SSE al tomar los centroides de forma aleatoria. Finalmente, una salida más viable podría ser la del uso de muestras aleatorias representativas, como por ejemplo la del 10%, que solo tuvo una baja de un 1.02% en el valor del SSE y se reducen los cálculos necesarios para hallar los centroides y clusters ya que no es necesario calcular la matriz de distancias por completo.

Referencias

1. A.K. Jain, M.N. Murty and P.J. Flynn: Data Clustering: A review. 16 March 2000.
2. P. J. Desréa: Homogeneous crystalline nucleation via atomic density fluctuations in the liquid state: Applications. Philosophical Magazine Letters 1994 vol 69, No. 5, 261-268(1994)
3. I.M. Sokolov, D.V. Kupriyanov, R.G. Olave, M.D. Havey: Light trapping in high-density ultracold atomic gases for quantum memory applications. 10 May 2010.
4. Daniel Fasulo: An Analysis of the Recent Works on Clustering Algorithms. 26 Apr 1999.
5. S.B. Kotsiantis, P. E. Pintelas: Recents Advances in Clustering: A brief Survey. 21 Jan 2004.
6. Toni Giorgino: Computing 1-D atomic densities in macromolecular simulations: the Density Profile Tool for VMD. 27 Aug 2013.
7. Romit Chakraborty and David A. Mazziotti: Generalized Pauli Conditions on the Spectra of One-electron Reduced Density. 21 Apr 2014.
8. Chunfei Zhang and Zhiyi Fang: An Improved K-means Clustering Algorithm. 1 Jan 2013.
9. Amandeep Kaur Mann and Navneet Kaur: Survey Paper on Clustering Techniques. Apr 2013.
10. Sutapat Thiprungsri: Cluster Analysis for Anomaly Detection in Accounting Data. 31 Jul 2010.
11. Qi Liu and Miklos Vasaherlyi: Healthcare fraud detection: A survey and a clustering model incorporating Geo-location information. 21 Nov 2013.
12. Clifton Phua, Vincent Lee, Kate Smith and Ross Gayler: A Comprehensive Survey of Data Mining-based Fraud Detection Research. 7 Nov 2007.
13. International Union of Pure and Applied Chemistry: Compendium of Chemical Terminology – Gold Book. 24 Feb 2014.
14. Bruce Cohen and David Sklar: The Gamma Function, Factorials and the Volumes of n-Balls. 1 Dec 2001.
15. Greg Glassman: Understanding Crossfit. The crossfit journal, Greg Glassman. Apr 2007.
16. Mario Köppen: The Curse of Dimensionality. Sep 2000.
17. Michel Deza and Elena Deza: Encyclopedia of Distances. 23 Oct 2012.