Anàlisi de Clústers

Marc Seguí Coll

Universitat Autònoma de Barcelona

Gener 2021

Resum

L'anàlisi de clústers tracta d'agrupar elements d'un conjunt de dades seguin patrons o similituds entre aquests. En aquest article tractarem de definir unes bases formals per poder tractar els diversos mètodes de clustering existents. En tenir clars els conceptes i algorismes descrits, els posarem en pràctica tractant de classificar exoplanetes semblants a la Terra.

Índex

I Introducció

I.1 Motivació

En numeroses ocasions, l'anàlisi multivariant pretén determinar si existeixen grups o clústers d'observacions de manera que els elements d'un grup de dades tenen alguna propietat comuna que els diferencia de la resta.

Les persones som especialment eficients descobrint aquests patrons i relacions quan observem una imatge. Fixem-nos en el següent exemple.

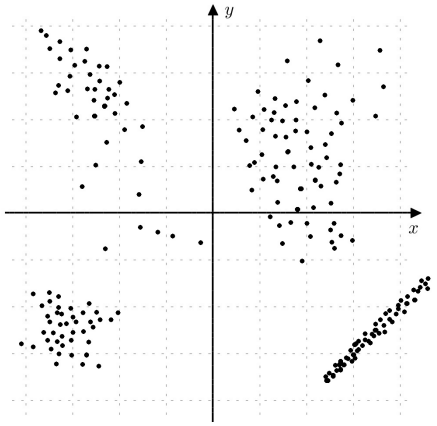

Figura 1: Mostra bivariada d'un conjunt de dades representada en el pla Cartesià (exemple il·lustratiu).

A la Figura 1 observem un conjunt de punts disposats al pla Cartesià. A simple vista podem distingir 4 grups o clústers principals. El més clar de tots és el que sembla un segment al quart quadrant. Els altres tres són aglomeracions més o menys compactes, una a cada quadrant. A més, al voltant de l'origen sembla que hi tenim alguns punts que van per lliure.

En aquest cas, la propietat comuna entre els elements (els punts) és la distància. La nostra ment no associa elements molt distants entre sí. És aquí on sorgeixen una sèrie de preguntes naturals. La primera pot ser relacionada amb la formalització d'aquest procés de reconeixement de patrons. Es pot establir alguna metodologia per poder discernir grups en un conjunt de dades mitjançant l'anàlisi d'aquestes? Acabem de veure un exemple en dues dimensions, però no és complicat visualitzar altres exemples en tres dimensions. En canvi, si tenim més de 4 dimensions, seguiran sent vàlids els arguments?

Així doncs, l'anàlisi de clústers pretén estudiar mètodes i algorismes per tal de determinar aquests grups donat un conjunt de dades i respondre les preguntes que hem formulat, entre altres. Aquest estudi és part del que es denomina classificació no supervisada [4]. Plantejar els Aquesta branca de l'anàlisi multivariant també ataca el problema en totes dimensions, ja que moltes vegades el tamany de les mostres requereix anar més enllà.

Tot i que l'enfocament d'aquest article va dirigit a la vessant més teòrica, aquesta branca de l'anàlisi multivariant té diverses aplicacions com, per exemple, el tractat i filtratge d'imatges. A més, avui dia l'anàlisi de clústers és molt útil en *machine learning* i xarxes neuronals.

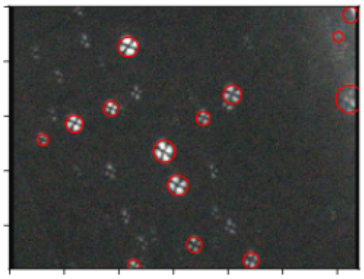

Figura 2: Selecció de cristalls de tamanys concrets a través d'un algorsime de *clustering* (imatge extreta de [10]).

I.2 Continguts i estructura

A la següent secció II presentem el formalisme propi de l'anàlisi de clústers, amb una sèrie de definicions i descripcions dels processos per poder passar més endavant a l'estudi de diversos algorismes i mètodes de *clustering*. A la tercera secció III s'apliquen aspectes explicats anteriorment a un exemple pràctic per tal de classificar exoplanetes. A continuació, a la secció IV es discuteixen els resultats i l'anàlisi realitzat. S'inclouen les conclusions de l'article en aquesta mateixa secció. Després de la bibliografia adjuntem un annex amb el codi emprat.

II Metodologia

II.1 Teoria i definicions bàsiques

Un **clúster** és un conjunt tal que els seus elements que són *semblants* entre sí. Elements de clústers diferents no són *semblants*. Notem que la relació entre dos elements de *ser semblants* no està definida. Aquesta depèn del context, de la naturalesa dels elements considerats i de la finalitat de l'anàlisi de dades que estem realitzant. En anàlisi multivariant, moltes vegades les dades se'ns presentaran de forma numèrica (si les observacions són categòriques, es pot assignar un valor numèric a cada observació) [1].

A partir d'ara direm que un **patró** és la representació d'un element del conjunt d'observacions. Sempre que el context ho permeti, emprarem les paraules element i patró indistintament. En aquest article ens centrarem en els casos en que les representacions són en un espai mètric, és a dir, podem considerar que cada patró $\mathbf{x} = (x_1, \ldots, x_n)$ està format per n atributs [4].

Els **atributs** d'un patró $\mathbf{x}$ són les components escalars individuals x_i de $\mathbf{x}$. Fixem-nos doncs que els atributs es poden interpretar com els paràmetres del sistema de coordenades escollit [4]. D'aquesta manera, parlarem de *proximitat* de patrons per referir-nos a la similitud entre clústers. És a dir, una sèrie d'elements pertanyen al mateix clúster si els patrons que tenen associats són pròxims entre si i estan allunyats d'altres patrons.

Una **mesura de la proximitat** és una distància a l'espai d'atributs emprada per quantificar la proximitat entre patrons [4]. Definir aquesta mesura de la proximitat és part del problema, ja que hem dit que dos elements pertanyen al mateix clúster si els seus patrons associats són pròxims entre si i estan allunyats de la resta.

Davies i Donald [1] estableixen els passos a seguir a l'hora de determinar els clústes d'un conjunt de dades:

1. Representació dels patrons.
2. Definició d'una mesura de la proximitat entre patrons apropiada per al domini on es troben les dades.
3. Agrupar les dades o *clustering*. Quan fem referència a la paraula *clustering*, volem dir agrupació de dades en clústers.
4. Abstracció de les dades (si fos necessari).
5. Valoració i interpretació dels resultats.

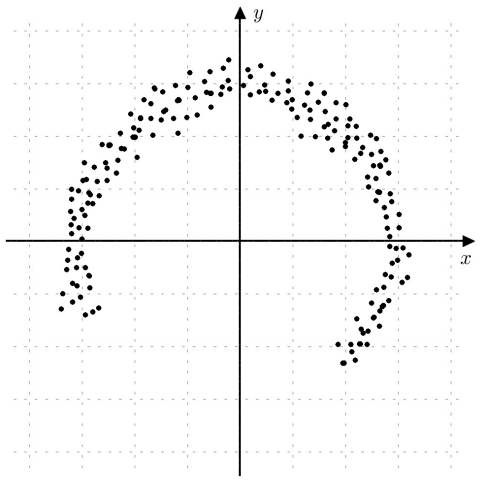

Figura 3: Dades corresponents a una mostra bivariant representada al pla Cartesià. Les dades estan situades concèntricament (exemple il·lustratiu).

Les dues primeres tasques són les més importants en el procés. Fixem-nos que la mesura de la proximitat està definida sobre l'espai d'atributs. És a dir, depenent de la representació que escollim per al nostre sistema de coordenades (representació dels patrons), la mesura de proximitat estarà definida sobre un espai o un altre. Això pot resultar que hi hagi patrons que siguin pròxims en una representació i que estiguin allunyats en una altra.

A la Figura 3 podem observar com les dades estan distribuïdes de forma concèntrica. Suposem que al pla Cartesià definim la distància entre dos punts (x_1, y_1) i (x_2, y_2) com la distància horitzontal entre els punts és a dir, $|x_1 - x_2|$. Els punts propers a l'eix positiu de les x estaran molt allunyats dels punts propers a l'eix negatiu de les x, motiu pel qual el mètode de *clustering* que emprem separi aquests dos conjunts de punts en dos clústers diferents. Suposem ara que

en lloc de tenir els punts donats pels atributs (x, y) -posició en la horitzontal i en la vertical- els tenim segons els atributs (r, θ) -distància a l'origen i angle respecte la horitzontal-. Si tenim la mateixa distància, ara la distància en aquests nous atributs (paràmetres) serà $|r_1 - r_2| \approx 0$ ja que tots els punts estan aproximadament a la mateixa distància de l'origen i per tant, el mètode agruparà tots els punts junts.

Els darrers dos passos es realitzen un cop ja tenim els resultats. L'anàlisi d'aquests pot indicar que algun dels paràmetres emprats ha de ser modificat. Això és més senzill en 2 ò 3 dimensions ja que, com hem comentat abans, podem visualitzar les dades de manera més intuïtiva.

II.2 Mesures de proximitat

Recordem primer la definició de distància. Sigui E conjunt. Una distància d és una aplicació $d : E \times E \to \mathbb{R}$ tal que

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$ per a tot $\mathbf{x}$, $\mathbf{y} \in E$ i $d(\mathbf{x}, \mathbf{y}) = 0$ si i només si $\mathbf{x} = \mathbf{y}$.

2. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ per a tot $\mathbf{x}$, $\mathbf{y} \in E$.

3. Desigualtat triangular:

$$d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$$

   per a tot $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z} \in E$.

Per tant, si E és l'espai d'atributs en el nostre experiment, llavors d serà una mesura de proximitat.

Suposarem doncs a partir d'ara que E és l'espai d'atributs i que el domini de les distàncies que introduirem a continuació és $E \times E$ de manera que seran mesures de proximitat[1]. A més, ens restringirem al cas en què els atributs són continus. Siguin doncs, $\mathbf{x} = (x_1, \ldots, x_n)$ i $\mathbf{y} = (y_1, \ldots, y_n)$ patrons de E, que considerarem vectors fila.

Per mesurar la proximitat entre dos patrons, la manera més intuïtiva de determinar la distància entre aquests és a partir de la norma Euclidiana:

$$d_2(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2}$$

que és un cas particular de la norma a L_p:

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

Tot i això, tal i com suggereixen Jain, Murty i Flynn [4], l'atribut de major magnitud predomina respecte els altres. Per solucionar això, es poden normalitzar els atributs.

D'altra banda, també pot ser que si hi ha molta correlació (lineal) entre els atributs, les mesures de la distància poden estar distorsionades. Una distància que millora els resultats en aquest cas és la distància de Mahalanobis:

$$d_M(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})\Sigma^{-1}(\mathbf{x} - \mathbf{y})^t$$

amb Σ la matriu de covariàncies mostrals entre els atributs (o, si és coneguda i està ben definida, la matriu de covariància entre els atributs). Aquesta distància assigna diferents pesos als atributs. Si Σ és la identitat recuperem la distància Euclidiana i si Σ és diagonal parlem de la distància Euclidiana estandaritzada.

II.3 Algorismes de *clustering*

Per tal de generar els clústers pròpiament dits, una vegada determinada la distància que s'emprarà en l'anàlisi, es requereix d'algun tipus de mecanisme de decisió per separar patrons. A la majoria de texts (veure per exemple [1], [4] i [2]) la distinció principal en els mètodes de *clustering* depèn de si l'algorisme és de tipus jeràrquic o de tipus particional. Tot i això, alguns algorismes no es poden classificar estrictament en aquests dos grans grups, com són els algorismes basats en la densitat del elements o en la seva distribució.

D'entre els algorismes jeràrquics, distingim entre aglomeratius i divisius. Els aglomeratius consideren inicialment que cada clúster està format per un únic element i va agrupant clústers segons els criteris escollits. Els divisius realitzen

[1] Davies i Bouldin [5] van proposar una formulació més extensa del concepte de mesura de proximitat.

la tasca inversa, comencen amb un únic clúster format per totes les observacions, de manera que l'algorisme el divideix seqüencialment en clústers més petits. En ambdues situacions, a cada pas s'obté una partició diferent a l'anterior.

En canvi, els algorismes particionals parteixen d'un nombre prefixat de clústers i tracten de determinar una única partició on els clústers siguin el més similars possible. A més, també es pot tractar de predir quin és el nombre òptim de clústers abans de l'aplicació de l'algorisme [2].

D'ara en endavant $\mathfrak{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_m}\}$ denotarà el conjunt de patrons del nostre experiment amb $\mathbf{x_j} = (x_{j_1}, \ldots, x_{j_n})$ per a tot $j \in \{1, \ldots, m\}$. A cada pas k de l'algorisme, tindrem una partició $\mathfrak{C}^{(k)} = \{C_1^{(k)}, \ldots, C_{m_k}^{(k)}\}$ de $\mathfrak{X}$. És a dir:

- $\bigcup_{i=1}^{m_k} C_i^{(k)} = \mathfrak{X}$ per a tot k.
- $C_i^{(k)} \cap C_j^{(k)} = \varnothing$ si $i \neq j$ i per a tot k.

Així $C_i^{(k)}$ denota el clúster i-èssim a cada pas k. Depenent del nombre de passos de l'algorisme acabarem amb una partició o una altra.

II.3.1 Algorismes jeràrquics

En el cas dels algorismes jeràrquics distingirem entre els aglomeratius i els divisius. En els aglomeratius sempre començarem la partició

$$\mathfrak{C}^{(0)} = \{C_1^{(0)}, \ldots, C_{m_0}^{(0)}\}$$

amb $m_0 = m$ de manera que $C_j^{(0)} = \{\mathbf{x_j}\}$, és a dir inicialment considerem que cada clúster està format per un únic patró. En canvi, en els divisius començarem amb

$$\mathfrak{C}^{(k)} = \mathfrak{X}$$

on disposem d'un únic clúster sobre el qual realitzarem divisions pas a pas.

Ens centrarem en els aglomeratius, però la idea al darrere és en essència la mateixa. Així doncs, calcular la distància entre clústers inicialment és senzill ja que és equivalent a determinar la distància entre patrons, que ens ve donada per la mesura de proximitat que haguem definit. La qüestió sorgeix en com determinar la distància entre clústers formats per més d'un patró [5]. Si considerem dos clústers $C_i^{(k)}$ i $C_j^{(k)}$ existeixen diversesos mètodes per definir i mesurar la distància $d_{ij}^{(k)}$ entre aquests [4]. Ens centrarem en uns quants dels casos més intuïtius i emprats [3], [2]. Denotarem per $p_j^{(k)}$ el nombre d'elements de $C_j^{(k)}$ per a tot $j \in \{1, \ldots, m_k\}$.

- ***Clustering* per enllaç simple**. La distància entre $C_i^{(k)}$ i $C_j^{(k)}$ és la mínima distància entre dos patrons $\mathbf{x}$ i $\mathbf{y}$ de manera que $\mathbf{x} \in C_i^{(k)}$ i $\mathbf{y} \in C_j^{(k)}$. Aquesta és la distància entre els elements més propers de cada clúster.

$$d_{ij}^{(k)} = \min\left\{d(\mathbf{x}, \mathbf{y}) : x \in C_i^{(k)}, \mathbf{y} \in C_j^{(k)}\right\}$$

- ***Clustering* per enllaç complet**. Per mesurar la distància entre $C_i^{(k)}$ i $C_j^{(k)}$ considerem la màxima distància entre dos patrons $\mathbf{x}$ i $\mathbf{y}$ tals que $\mathbf{x} \in C_i^{(k)}$ i $\mathbf{y} \in C_j^{(k)}$. Es correspon a la distància entre els elements més allunyats de cada clúster.

$$d_{ij}^{(k)} = \max\left\{d(\mathbf{x}, \mathbf{y}) : x \in C_i^{(k)}, \mathbf{y} \in C_j^{(k)}\right\}$$

- ***Clustering* per enllaç promig**. Per determinar la distància entre $C_i^{(k)}$ i $C_j^{(k)}$ agafem el promig de les distàncies entre dos patrons $\mathbf{x}$ i $\mathbf{y}$ amb que $\mathbf{x} \in C_i^{(k)}$ i $\mathbf{y} \in C_j^{(k)}$. La idea prové de considerar les distàncies entre cada element d'un clúster i tots els elements de l'altre. Promitjant totes aquestes distàncies determinem la proximitat entre clústers.

$$d_{ij}^{(k)} = \frac{1}{p_i^{(k)} p_j^{(k)}} \sum_{\mathbf{x} \in C_i^{(k)}} \sum_{\mathbf{y} \in C_j^{(k)}} d(\mathbf{x}, \mathbf{y})$$

- ***Clustering* per enllaç de centroides**. La distància entre els clústers serà la distància entre els centroides de $C_i^{(k)}$ i $C_j^{(k)}$. En aquest cas, identifiquem cada clúster pel seu centroide de manera que cada clúster es pot considerar un punt en l'espai de paràmetres, en altres paraules, un patró.

$$d_{ij}^{(k)} = d\left(\mathbf{cent}_i^{(k)}, \mathbf{cent}_j^{(k)}\right)$$

$$\mathbf{cent}_\ell^{(k)} = \frac{1}{p_\ell^{(k)}} \sum_{\mathbf{x} \in C_\ell^{(k)}}$$

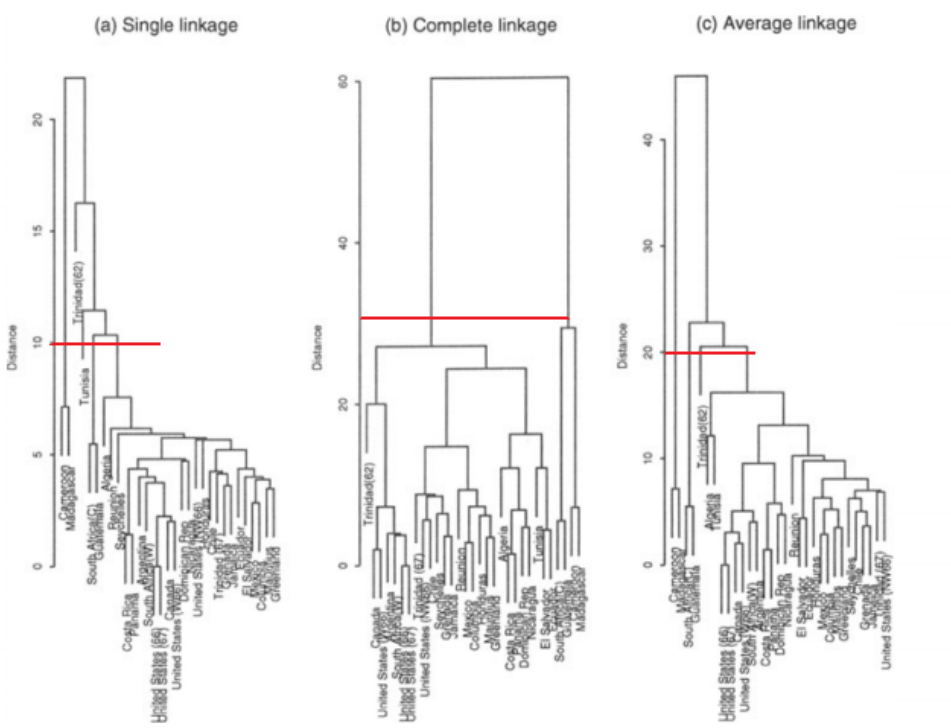

Figura 4: Dendrogrames corresponents a algorismes jeràrquics per enllaç simple, complet i promig d'esquerra a dreta per a un conjunt de dades sobre l'esperança de vida. La línia vermella indica el *threshold* del mètode (opcional). En cadascun tindrem 5, 2 i 4 clústers respectivament segons la línia (imatge extreta de [3]).

D'aquesta manera a cada pas k haurem de realitzar les següents accions:

1. Determinem les distàncies entre cada parella de clústers de $\mathfrak{C}^{(k)}$.

2. Escollim dos clústers diferents de $\mathfrak{C}^{(k)}$ que estiguin més propers que la resta. Triem $C_i^{(k)}$ i $C_j^{(k)}$ amb $i < j$ tals que $d_{ij}^{(k)} \leq d_{st}^{(k)}$ per a tots $s, t \in \{1, \ldots, m_k\}$ amb $s < t$.

3. Agrupem aquests dos clústers per al següent pas i no modifiquem la resta. Definim $C_s^{(k+1)} = C_s^{(k)}$ per a tot $1 \leq s < i$, $C_i^{(k+1)} = C_i^{(k)} \cup C_j^{(k)}$ i $C_t^{(k+1)} = C_{t+1}^{(k)}$ per a tot $j \leq t \leq m_k - 1$. En aquest tipus d'algorismes, $m_{k+1} = m_k - 1$.

4. Si $m_{k+1} = 1$ finalitza[2] el procés.

Podria donar-se el cas que la parella de clústers al segon pas no fos única. En aquest cas hi hauria dues parelles de clústers que entre sí són més propers. Fem notar que depenent de la tria, les particions posteriors poden ser diferents. Fernández i Gómez [6] analitzen criteris per a l'elecció de la parella de clústers adequada. En aquest article no entrarem en detalls i suposarem que la tria es realitza de forma aleatòria.

Amb aquesta classificació jeràrquica podem visualitzar de manera gràfica els passos que va seguint l'algorisme mitjançant un *dendrograma*. Un **dendrograma** és una representació en arbre de la seqüencia de *clustering* jeràrquic. Normalment, a l'eix horitzontal es solen indexar els patrons mentre que a l'eix vertical es fa referència a la distància entre clústers [2].

[2]Si es desitja, el procés pot acabar abans. Una manera senzilla de fer-ho és imposar que es finalitzi l'algorisme si $m_{k+1} = N \in \mathbb{N}$ amb N el nombre és petit de clústers al que es vol arribar. També és pot determinar un *threshold* tal que si la distància entre és superior a aquest llindar l'algorisme finalitzi.

II.3.2 Algorismes particionals

Tot i que els algorismes jeràrquics són molt útils, quan m es fa gran la visualització dels dendrogrames es complica i deixen de ser-ho. En aquest aspecte, els algorismes particionals solventen aquest problema [4]. En aquests, a cada pas treballarem amb N clústers. Així doncs, la partició inicial serà:

$$\mathfrak{C}^{(0)} = \{C_1^{(0)}, \ldots, C_{m_0}^{(0)}\}$$

amb $m_0 = N$. De fet, a cada pas k la partició tindrà la forma

$$\mathfrak{C}^{(k)} = \{C_1^{(k)}, \ldots, C_N^{(k)}\}$$

El problema llavors queda reduït a com organitzar m elements en N grups. A cada pas de l'algorisme s'avalua una de les formes d'agrupar els m patrons en N clústers segons els criteris de similitud que s'hagin establert. Tot i que sembli senzill, anar comprovant cada possible partició és computacionalment inefectiu. De fet, fent $m = 10$ i $N = 4$, existeixen 34105 possibilitats diferents, que tampoc són tantes, però si $m = 19$ i $N = 4$ el nombre de particions possibles és de l'ordre de 10^{10} [1]. La solució més comuna és requerir a un procés iteratiu en el que alguns elements de la partició inicial canvien de clúster. Així, només s'analitzen les particions creades a partir de petites pertorbacions de la partició inicial i s'obté una solució local. En aquest cas la qüestió radica en com decidir la partició inicial, quins elements moure i com moure'ls [3].

Definim **l'error quadràtic de la partició** $\mathfrak{C}^{(k)} = \{C_1^{(k)}, \ldots, C_N^{(k)}\}$ com

$$\mathrm{SE}^{(k)} = \mathrm{SE}\left(\mathfrak{C}^{(k)}\right) = \sum_{j=1}^{N} \sum_{\mathbf{x} \in C_j^{(k)}} d\left(\mathbf{x}, \mathbf{cent}_j^{(k)}\right)^2$$

amb $\mathbf{cent}_j^{(k)}$ el centroide de $C_j^{(k)}$. Per cada clúster, considerem la distància al quadrat entre cada patró i el seu centroide. Sumant respecte tots els patrons de cada clúster i posteriorment sobre tots els clústers obtenim l'error quadràtic de $\mathfrak{C}^{(k)}$.

El criteri més intuïtiu i emprat en aplicar algorismes particionals és trobar la partició que minimitza l'error quadràtic. És a dir, de les particions analitzades, ens quedarem amb la que tingui menor error quadràtic associat.

- ***Clustering* mitjançant l'algorisme *K-Means***[3]. És l'algorisme particional més famós i extés en la comunitat científica tot i tenir certes mancances. Inicialment, es trien de forma aleatòria o amb algun criteri previ K patrons $\{\mathbf{x}_1^{(0)}, \ldots, \mathbf{x}_K^{(0)}\}$ de $\mathfrak{X}$ ò K punts de l'espai de patrons que actuaran com a centres dels clústers inicials. Denotarem el conjunt de centres per $c = \{\mathbf{c_1}, \ldots, \mathbf{c_K}\}$ i farem $C_j^{(0)} = \{\mathbf{c_j}\}$ per a cada j. Llavors, al pas k-èssim es realitzen les accions següents:
  1. Definir els clústers a cada pas. Es defineix $C_j^{(k)}$ com el subconjunt de $\mathfrak{X}$ tal que $x \in C_j^{(k)}$ si $d(\mathbf{x}, \mathbf{c_j}) \leq d(\mathbf{x}, \mathbf{c_i})$ per a tot $i \in \{1, \ldots, K\}$ ($\mathbf{x}$ pertany al clúster que té el centre més a prop).
  2. Redefinir els centres de manera que per a tot j, $c_j = \mathbf{cent}_j^{(k)}$ és el centroide del clúster $C_j^{(k)}$.
  3. Comprovar si els criteris de finalització es satisfan. Normalment la condició per parar l'algorisme s'assoleix si $\mathbf{cent}_j^{(k)} = \mathbf{cent}_j^{(k-1)}$ per a tot j (els centroides no es desplacen) o bé $|\mathrm{SE}^{(k)} - \mathrm{SE}^{(k-1)}| < \varepsilon$ per algun $\varepsilon > 0$ prefixat.

Diverses modificacions i millores d'aquest algorisme han estat emprades en molts articles científics [4].

Ara bé, ens podem preguntar com determinar el valor de K òptim per al nostre problema. Si podem visualitzar les dades d'alguna manera en l'espai de paràmetres (bé perquè l'espai de dimensió inferior a 4 o perquè s'han analitzat les components principals de les dades) llavors podem intuir quin serà el nombre de clústers adequat. En moltes ocasions s'ha tractat de determinar algun procediment efectiu per resoldre aquest problema [4], [3] però no s'han trobat resultats satisfactoris en el sentit que no aporten evidència suficient per poder ser generalitzats.

[3] A la literatura es troba com a *K-Means* degut que les particions són en K clústers.

A continuació explorarem un dels mètodes usuals a l'hora de determinar K sense anàlisi previ de les dades anomenat el mètode del "colze" o "genoll". El procediment a seguir és prou raonable. Considerarem un subconjunt Ω de $\mathbb{N}$ de dimensió finita format per nombres naturals consecutius. Aplicarem l'algorisme *K-Means* sobre $\mathfrak{X}$ per a tot $K \in \Omega$ i ens quedarem amb el valor de SE per a cada K. Evidentment, en augmentar K, disminuirà SE (el cas extrem en què $K = m$ tindrem SE és la suma de les distàncies al quadrat de només els patrons), per tant tenir un SE petit no és indicador sobre quin és el millor K. Al graficar SE vs. K, pot ser trobem un canvi dràstic en passar de K a $K+1$. En aquest cas, considerarem que K és el valor òptim. En la majoria de casos per valors superiors a aquest K (on hem trobat el canvi abrupte) SE no varia de manera significativa respecte a la variació entre K i $K+1$ [3].

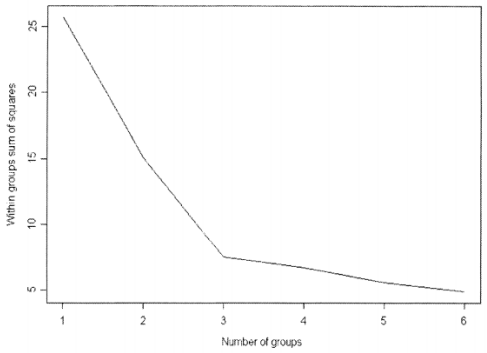

Figura 5: Plot de l'error quadràtic en funció del nombre de particions per a un conjunt de dades de ceràmica (extret de [3]). El "colze" a $K = 3$ i la posterior estabilitat de SE indica que $K = 3$ és un bon valor per realitzar els passos per K-means.

II.3.3 Altres algorismes

Com ja hem esmentat, hi ha algorismes que no es poden encasellar directament als grups anteriors. És el cas, per exemple, dels algorismes basats en la distribució de les dades o la densitat dels patrons.

En ocasions, s'assumeix que les dades segueixen certes distribucions i els algorismes distribucionals tracten de determinar quines distribucions segueixen els clústers i els paràmetres de dites distribucions. El cas més estudiat són els algorismes de mescla Gaussiana que tracten d'identificar mostres Gaussianes d'entre el total de dades.

El concepte de densitat està relacionat amb l'aglomeració de patrons. De manera poc rigorosa, es pot pensar que regions denses són aquelles on la quantitat de patrons és elevada respecte al seu voltant [7]. En l'actualitat l'algorisme DBSCAN és un dels més emprats en anàlisi de clústers juntament amb l'algorisme *K-Means*.

III Anàlisi d'un cas pràctic: classificació d'exoplanetes

Un dels objectius de la recerca en astronomia és trobar planetes o satèl·lits naturals amb medis per acollid vida. Direm que un planeta o satèl·lit és habitable si és capaç de sustentar vida [8].

Com es desconeix l'existència de vida extraterrestre, els criteris per decidir si un planeta és o no és habitable estan basats en les característiques pròpies que té la Terra. Els trets més evidents són el tamany, la massa i la temperatura a la superfície. Una altra propietat en la que ens fixarem és en la excentricitat. La excentricitat té en compte la màxima i la mínima distància a la que es situa el planeta de l'estrella durant la translació. Si hi ha molta diferència entre aquestes distàncies, hi haurà èpoques de molt fred i altres de molta calor. Per profunditzar més, ens podríem preocupar també per les propietats de l'estrella que envolten i comparar-les amb les del Sol.

En aquesta secció analitzarem els elements d'una base de dades d'exoplanetes extreta del Open Exoplanet Catalogue [9]. En aquest conjunt de dades, s'especifica el nom del planeta, la seva massa, el seu radi, el seu període de rotació, la temperatura de la superfície, la seva excentricitat i més característiques. Mitjançant algorismes de *clustering* sobre aquest conjunt de dades, tractarem de determinar quins són exoplanetes són habitables.

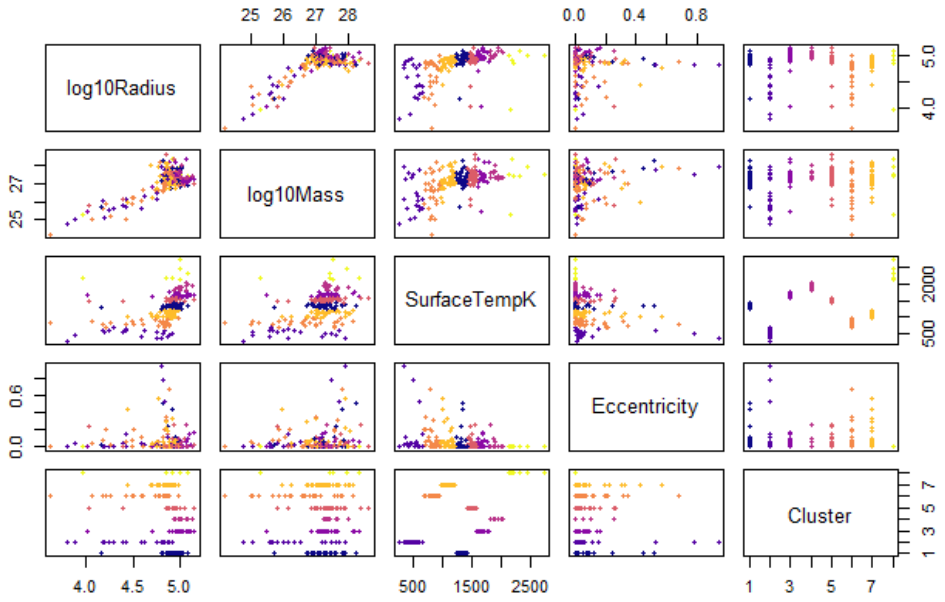

Figura 6: *K-Means* amb $K = 8$ realitzat a **R** sobre el conjunt de dades `oec.csv` de [9].

III.1 Aplicació dels algorsimes de *clustering*

En aquest experiment considerarem que un planeta és habitable si, en totes les proves, es troba al mateix clúster que la Terra. Abans de començar amb els algorismes, comentem els atributs que analitzarem: el radi del planeta, la massa del planeta, la temperatura a la superfície del planeta i la seva eccentricitat. Pel que fa al radi i a la massa, ens interessa que aquestes quantitats siguin de l'ordre dels de la terra. Per aquest motiu, canviarem aquestes magnituds en l'espai d'atributs pel seu logaritme en base 10. L'excentricitat és una quantitat que ja ve escalada per tal que es pugui comparar entre planetes de característiques de tot tipus així que no li aplicarem cap modificació. En quant a la temperatura, la mesura d'aquesta es realitza en escala absoluta. A més, els éssers vius són susceptibles a canvis de temperatura elevats, per tant, comparar temperatures per ordre de magnitud no ens serveix i per tant tampoc la modificarem.

III.1.1 *Clustering* per *K-Means*

Procedirem primer amb l'algorisme K-means genèric (pels 4 atributs) amb la distància Euclidiana. A la Figura 6 observem els resultats. La Terra pertany al clúster número 5 (la podem identificar per la temperatura de la superfície). En aquest grup hi ha 17 planetes que l'algorisme ha considerat similars a la Terra que considerarem habitables. Tot i això no és gaire fi ja que la temperatura pesa prou i hi ha planetes suposadament habitables amb temperatura superior a 500 K, massa calenta per la vida que coneixem nosaltres.

Per millorar això, realitzarem un *K-Means* separat. Primer seleccionarem els planetes segons tamany i massa (Figura 7) i despres segons excentricitat i temperatura a la superfície (Figura 8).

Inicialment, amb aquest mètode arribem que la Terra pertany al clúster més proper als tamanys i masses més petits format per 9 planetes. Ara, agafem aquests patrons i els estudiem a part, per determina-ne les similituds entre els altres dos atributs, seguint el mateix algoritme.

Finalment ens quedem amb la Terra i 2 planetes més, KOI-414 b i Kepler-11 f, ambdós trobats també al K-means global. El més "fred" dels dos és KOI-414 b amb una temperatura a la superfície de 376 K.

Els valors de K emprats a cada algorisme han estat determinats pel mètode del "colze" (veure B).

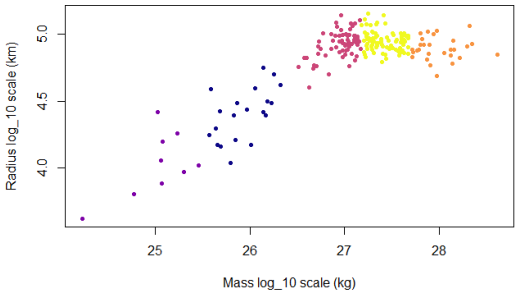

Figura 7: *K-Means* amb $K = 5$ realitzat a **R** sobre el conjunt de dades `oec.csv` de [9] agafant només els atributs de radi i massa.

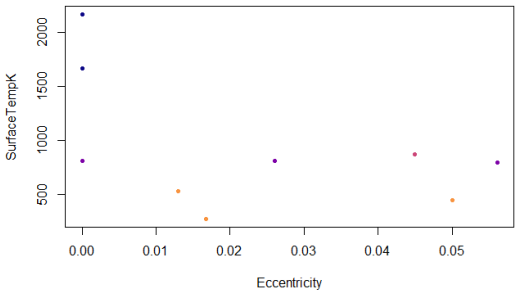

Figura 8: *K-Means* amb $K = 4$ realitzat a **R** sobre el conjunt de dades `oec.csv` de [9] agafant només els atributs de excentricitat i temperatura a la superfície.

IV Discussió i conclusions

Al llarg de l'article hem intentat establir un formalisme bàsic i rigorós sobre l'anàlisi de clústers. Una vegada païts els conceptes i les noves definicions, hem reproduit la teoria mitjançan un exemple pràctic per determinar exoplanetes similars a la Terra.

Tot i no haver obtingut resultats prou satisfactoris, es pot millorar el procediment tenint en compte una distància que no sigui la Euclidiana, com per exemple la de Mahalanobis. Com estem buscant planetes semblants a la Terra, es podria donar més pes a punts propers als punts on està aquesta (coneixem totes les característiques de la Terra que ens fan falta). A més, hem descartat molts planetes inicialment per manca d'alguna de les dades que hem emprat. Si comparéssim uns altres atributs els resultats serien diferents i pot ser millors. Com hem comentat, es podrien utilitzar les propietats de les estrelles del sistema planetari de cada exoplaneta (un planeta és potencialment habitable si pot albergar aigua).

A més, hem atacat el problema amb un algorisme particional, però pel que hem mencionat just ara, pot ser un algorisme basat en la densitat dels patrons milloraria l'*output* del programa. Encara així, el mètode no ha escollit barbaritats dins les opcions de les que disposava, cosa que suma a favor. Hem comprovat també que *K-Means* funciona millor quan no tenim gaires atributs en el nostre experiment. La diferència entre usar 2 ò 4 atributs ha sigut notable.

Referències

[1] A. K. Jain, R. C. Dubes. Michigan State University. *Algorithms for Clustering Data*. Prentice Hall (1988)

[2] W. K. Härdle, L. Simar. Humboldt-Universität zu Berlin, Katholieke Universiteit Leuven. *Applied Multivariate Statistical Analysis*. Springer (2012)

[3] B. S. Everitt. King's College. *An R and S-PLUS® Companion to Multivariate Analysis*. Springer (2005)

[4] A. K. Jain, M. N. Murty, P. J. Flynn. Michigan State University, Indian Institue of Science, The Ohio State University. *Data Clustering: A Review*. ACM Computing Surveys (1999)

[5] D. L. Davies, D. W. Bouldin. University of Tennessee. *A Cluster Separation Measure*. IEEE (1979)

[6] A. Fernández, S. Gómez. Universitat Rovira i Virgili. *Solving Non-Uniqueness in Agglomerative Hierarchical Clustering Using Multidendrograms*. Journal of Classification (2008)

[7] H. Kriegel, P. Kröger, J. Sander, A. Simek. Ludwig-Maximilians-Universität München, University of Alberta. *Density-based clustering*. WIREs Data Knowledge Discovery (2011)

[8] P. Dyches, F. Chou. Jet Propulsion Laboratory, NASA Headquarters. *The Solar System and Beyond is Awash in Water* NASA (2015)

[9] A. Tribick, C. Sturm, H. Rein and more. Múltiples organitzacions. *Open Exoplanet Catalogue Database: All extrasolar planets* Open Exoplanet Catalogue (2021)

[10] C. Falcó, M. Seguí, S. Serrano. Universitat Autònoma de Barcelona. *Estats metaestables: obtenció dels estats amorf i cristal·lí en un polímer* Laboratori de Termodinàmica de la UAB (2019)

A Codi de R

```
#Exoplanetes
library(tidyr)
dades <- as.data.frame(read.csv("oec.csv"))
dades <- dades %>% drop_na(PlanetaryMassJpt,
                           SurfaceTempK,
                           RadiusJpt,
                           Eccentricity)
indexTerra <- which(dades$PlanetIdentifier == "Earth")
dades$log10Radius <- log(dades$RadiusJpt*69911,base=10)
dades$log10Mass <- log(dades$PlanetaryMassJpt*1.898*10^(27),base=10)
mostra <- subset(dades,
                 select = c(log10Radius,
                            log10Mass,
                            SurfaceTempK,
                            Eccentricity))
#Algorisme 1: K-means global
a <- rep(1,nrow(mostra)-1) #Auxiliar
for(i in 1:(nrow(mostra)-1)){
  a[i] <- kmeans(mostra,i)$tot.withinss
}
plot(1:(nrow(mostra)-1),
     a,
     xlim=c(0,20),
     ylab="SE",
     xlab="N",
     pch=20)
mostrakm <- kmeans(mostra,8)
mostrakm
mostra$Cluster <- mostrakm$cluster
plot(mostra,col = plasma(8)[mostrakm$cluster],pch=20)
habitables <- subset(dades,
                     mostrakm$cluster
                     == mostrakm$cluster[indexTerra])
habitables$PlanetIdentifier

#Segon algorisme: k-means dividit
#Primer pas. k-means 2 dimensional amb Massa vs. Radi
MR <- subset(mostra,select = c(log10Mass,log10Radius))
a <- rep(1,nrow(MR)-1)
for(i in 1:(nrow(MR)-1)){
  a[i] <- kmeans(MR,i)$tot.withinss
}
plot(1:(nrow(MR)-1),
     a,
     xlim=c(0,20),
```

```
      ylab="SE",
      xlab="N",
      pch=20)
MRkm <- kmeans(MR,5)
MRkm$cluster[indexTerra]
plot(MR, col = plasma(5)[MRkm$cluster],
      pch=20,
      ylab="Radius log_10 scale (km)",
      xlab="Mass log_10 scale (kg)")
habitables <- subset(dades,
                     MRkm$cluster
                     == MRkm$cluster[indexTerra])
indexTerra <- which(habitables$PlanetIdentifier == "Earth")
#Segon pas. k-means 2 dimensional amb Excen vs. Temp
ET <- subset(habitables,select = c(Eccentricity,SurfaceTempK))
a <- rep(1,nrow(ET)-1)
for(i in 1:(nrow(ET)-1)){
  a[i] <- kmeans(ET,i)$tot.withinss
}
kmeans(ET,8)$tot.withinss
plot(1:(nrow(ET)-1),
      a,
      xlim=c(0,9),
      ylab="SE",
      xlab="N",
      pch=20)
ETkm <- kmeans(ET,4)
plot(ET, col = plasma(5)[ETkm$cluster],pch=20)
habitables <- subset(habitables,ETkm$cluster
                     == ETkm$cluster[indexTerra])
habitables$PlanetIdentifier
```

B Gràfics colze per la determinació de K

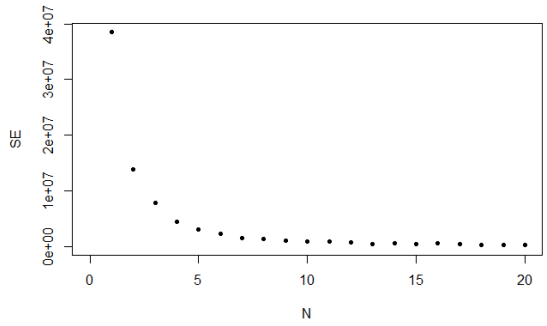

Figura 9: SE per *K-Means* global.

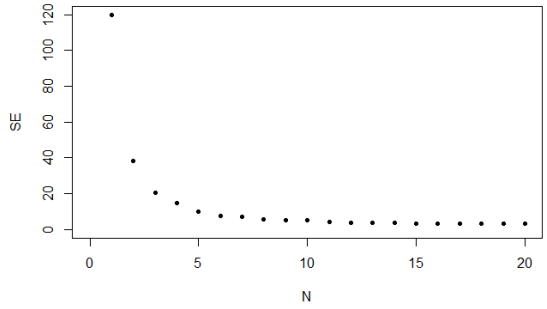

Figura 10: SE per *K-Means* entre la massa i el radi.

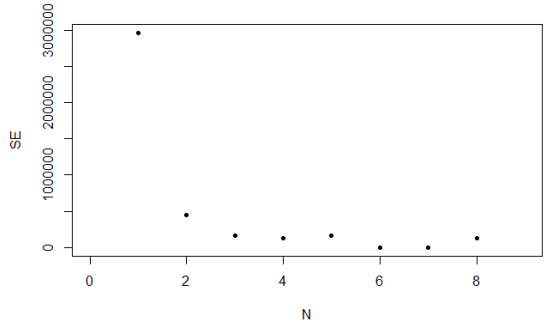

Figura 11: SE per *K-Means* entre la temperatura a la superfície i la excentricitat.