Models no supervisats

PID_00284570

Jordi Gironés Roig

Temps mínim de dedicació recomanat: 2 hores

Universitat Oberta de Catalunya

Jordi Gironés Roig

Llicenciat en Matemàtiques per la Universitat Autònoma de Barcelona i diplomat en Empresarials per la Universitat Oberta de Catalunya. Ha desenvolupat la major part de la seva carrera professional al voltant de la solució SAP, en els seus vessants operatius amb S4-HANA i estratègic amb SAP-BI. Actualment treballa en la indústria quimicofarmacèutica com a responsable d'aplicacions corporatives per a Esteve Pharmaceuticals i col·labora amb la UOC en assignatures relacionades amb l'anàlisi de dades.

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Julià Minguillón Alfonso

Primera edició: setembre 2021

Av. Tibidabo, 39-43, 08035 Barcelona
Autoria: Jordi Gironés Roig
Producció: FUOC

Índex

Introducció 5

1. Conceptes preliminars 7
1.1. Distància o similitud 7
1.1.1. Euclides 8
1.1.2. Gauss 8
1.1.3. Mahalanobis 9

2. *Clustering* i segmentació 11
2.1. Agrupació jeràrquica 11
2.1.1. Algorismes de tipus aglomeratiu 11
2.1.2. Algorismes de tipus divisiu 12
2.2. Algorismes particionals 13
2.2.1. *k-means* 13
2.2.2. Criteris per seleccionar *k* 14
2.3. *Canopy clustering* 16

3. Models basats en la densitat 19
3.1. Algorisme DBSCAN 19
3.1.1. Avantatges i inconvenients 20
3.2. Algorisme OPTICS 21
3.2.1. Avantatges i inconvenients 23

4. *Affinity propagation* 24

Introducció

La classificació no supervisada persegueix l'obtenció d'un model vàlid per classificar objectes a partir de la similitud de les seves característiques. Més formalment, podríem dir-ho de la manera següent:

> A partir d'un conjunt d'objectes descrits per un vector de característiques i a partir d'una mètrica que ens defineixi el concepte de similitud entre objectes, es construeix un model o regla general que ens permetrà classificar tots els objectes.

No es tracta de models predictius, sinó de models de descobriment de patrons. Hi ha dues grans famílies d'algorismes de classificació no supervisada:

1) Algorismes jeràrquics: construeixen nodes de manera jeràrquica, uns a partir d'uns altres. La representació dels resultats es fa habitualment mitjançant dendrogrames. Es divideixen en dues subcategories:

- Aglomeratius o *bottom up*. Aquesta aproximació parteix del supòsit que cada objecte és un node o clúster i, a mesura que evolucionen els passos de l'algorisme, els nodes es van agrupant fins a aconseguir un nombre de nodes acceptable.

- Divisius o *top down*. És l'aproximació oposada, és a dir, parteix del supòsit que hi ha un únic node i, a mesura que avança l'algorisme, aquest node es va subdividint en nous nodes, i així successivament.

  Exemples d'algorismes jeràrquics poden ser el mètode del mínim, o *single linkage*, i el mètode del màxim, o *complete linkage*.

2) Algorismes particionals: també anomenats *algorismes d'optimització*, obtenen els nodes a partir de l'optimització d'una funció adequada per al propòsit de l'estudi. Aquesta funció sol estar relacionada amb la mètrica seleccionada per establir el concepte de similitud entre objectes.

1. Conceptes preliminars

Com hem vist, molts models no supervisats basen la seva lògica en el concepte de *distància* o *similitud*, de manera que val la pena dedicar temps a comprendre'l, ja que ens ajudarà a entendre el mecanisme d'algorismes com ara el *k-means* i el *clustering*, entre altres.

1.1. Distància o similitud

Quan parlem de *distància*, en realitat ens referim a una manera de quantificar com de similars són dos objectes, dues variables o dos punts. Plantejat així, el concepte de *distància* és molt abstracte i eteri; per aquest motiu, els científics hi volen posar alguns límits o condicions. Per a un conjunt d'elements X, es considera com a distància qualsevol funció

$$(XxX) \rightarrow R \tag{1}$$

que compleixi les tres condicions següents:

- La no negativitat: la distància entre dos punts ha de ser sempre positiva.

$$d(a,b) \geq 0 \quad \forall a,b \in X \tag{2}$$

- La simetria: la distància entre un punt a i un punt b ha de ser la mateixa que la distància entre el punt b i el punt a.

$$d(a,b) = d(b,a) \quad \forall a,b \in X \tag{3}$$

- La desigualtat triangular: la distància ha de coincidir amb la idea intuïtiva que en tenim com a camí més curt entre dos punts.

$$d(a,b) \leq d(a,c) + d(c,b) \quad \forall a,b,c \in X \tag{4}$$

Amb aquestes tres condicions, ens centrarem en tres definicions de distància molt peculiars: l'euclidiana o clàssica, l'estadística o de Gauss i la que va proposar Mahalanobis.

1.1.1. Euclides

La distància euclidiana coincideix plenament amb el que la nostra intuïció entén per *distància*. La podríem anomenar *distància ordinària*.

La seva expressió matemàtica per a un espai, per exemple, de dues dimensions i per als punts $A(x_1,y_1)$, $B(x_2,y_2)$ seria:

$$d(A,B) = d((x_1,x_2),(y_1,y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (5)$$

Utilitzar aquesta distància en un procés de segmentació presenta un inconvenient. **No té en compte les escales** en què es poden expressar les variables X i Y.

Exemple

Si la variable X representa l'edat de dos ratolins, aleshores prendrà valors d'entre 0 i 3. Si la variable Y representa la longitud de la cua dels ratolins en mil·límetres, pot prendre valors d'entre 90 i 120.

Sembla clar que, quan vulguem comparar i calculem la distància entre dos individus, pesarà injustament molt més la longitud de la cua que l'edat.

1.1.2. Gauss

Per superar la distorsió provocada per les diferents unitats de mesura emprades en les diferents variables estudiades, tenim la distància estadística, que simplement normalitza les variables per situar-les totes a la mateixa escala.

Distància de Gauss

Aquest avenç li devem al brillant matemàtic alemany Carl Friedrich Gauss (1777–1855).

La seva expressió analítica per a un espai, per exemple, de dues dimensions i per als punts $A(x_1,y_1)$, $B(x_2,y_2)$, en què la nostra distribució de punts en aquestes dimensions tingués una desviació estàndard σ, seria:

$$d(A,B) = \sqrt{\left(\frac{x_2 - x_1}{\sigma(X)}\right)^2 + \left(\frac{y_2 - y_1}{\sigma(Y)}\right)^2} \quad (6)$$

Novament, aquest concepte de distància té problemes. **No té en compte la correlació** entre les variables, és a dir, si les nostres variables fossin totalment independents no hi hauria cap problema. Si tenen algun tipus de correlació, però, una influeix sobre l'altra i aquesta influència no queda ben reflectida si usem la distància estadística.

Exemple

Les variables *entrenar* i *rendiment* estan correlacionades, de manera que més entrenament sempre implica més rendiment, però aquesta regla no es compleix infinitament, ja que entrenament infinit no implica rendiment infinit (ho veiem contínuament en l'esport). Si no tenim en compte aquesta correlació i volem comparar dos esportistes, podem arribar a conclusions errònies.

1.1.3. Mahalanobis

Prasanta Chandra Mahalanobis (Índia) el 1936 es va adonar d'aquesta mancança i va proposar corregir la distorsió provocada per la correlació de les variables mitjançant l'expressió següent:

La versió simplificada per a un espai, per exemple, de dues dimensions, amb un conjunt de punts de variància $\sigma^2(X)$,$\sigma^2(Y)$ i covariància $\mathrm{cov}(X,Y)$, seria:

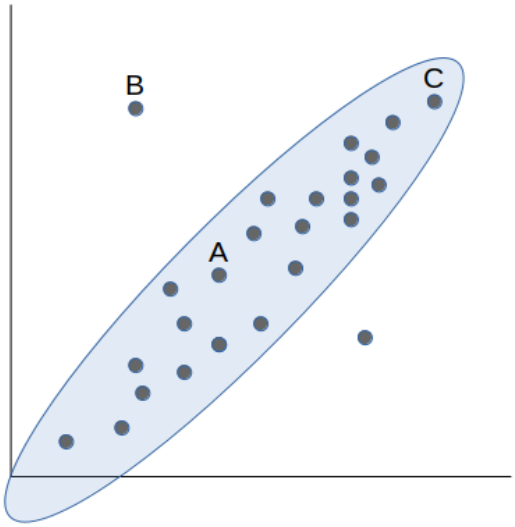
$$\sqrt{(x_1-y_1,x_2-y_2)\begin{bmatrix}\sigma^2(X) & \mathrm{cov}(X,Y)\\ \mathrm{cov}(Y,X) & \sigma^2(Y)\end{bmatrix}^{-1}(x_1-y_1,x_2-y_2)} \qquad (7)$$

La definició de *distància* proposada per Mahalanobis respon a la idea intuïtiva que els punts que es troben en una zona densament poblada s'haurien de considerar més pròxims entre ells que respecte a punts fora d'aquesta zona més densa.

Si prenem com a exemple la figura 1, veiem que una versió clàssica de distància ens diria que el punt *A* està més a prop de *B* que de *C*. Mahalanobis, però, entenent que els punts a la zona densa s'assemblen més, ens dirà que el punt *A* està més a prop de *C* que de *B*.

Que Mahalanobis generalitzi Gauss i Gauss a Euclides forma part del que s'ha anomenat *la bellesa de les matemàtiques*. Vegem-ho.

Figura 1. Distància de Mahalanobis



Si a la nostra distribució de punts, les dues dimensions són totalment independents, la covariància és 0, de manera que la distància de Mahalanobis coincideix amb la distància estadística.

$$\sqrt{(x_2 - x_1, y_2 - y_1)\begin{bmatrix}\sigma^2(X) & 0\\ 0 & \sigma^2(Y)\end{bmatrix}^{-1}(x_2 - x_1, y_2 - y_1)} = \tag{8}$$

$$= \sqrt{\left(\frac{x_2 - x_1}{\sigma(X)}\right)^2 + \left(\frac{y_2 - y_1}{\sigma(Y)}\right)^2} \tag{9}$$

Si a la mateixa distribució amb variables independents, la distribució està normalitzada, és a dir, les dues variables tenen la mateixa escala, aleshores la variància és 1, de manera que la distància estadística coincideix amb la distància d'Euclides.

$$\sqrt{(x_2 - x_1, y_2 - y_1)\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}^{-1}(x_2 - x_1, y_2 - y_1)} = \tag{10}$$

$$= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{11}$$

2. *Clustering* i segmentació

Clustering i *segmentation*, traduïts com a *agrupament* i *segmentació*, constitueixen l'àmbit de coneixement corresponent a les tècniques no supervisades, ja que no tenen com a objectiu predir una etiqueta que marqui cada observació del joc de dades.

El seu objectiu és descriure el joc de dades trobant patrons a partir de la identificació de grups similars.

D'aquesta manera, els conceptes de *distància* i *similitud* seran clau per entendre aquest tipus d'algorismes.

2.1. Agrupació jeràrquica

Dins d'aquesta família, hi distingirem dos tipus d'algorismes: els aglomeratius i els divisius.

2.1.1. Algorismes de tipus aglomeratiu

Els algorismes d'agrupació jeràrquica són **de tipus aglomeratiu** quan, partint d'una fragmentació completa de les dades, es van fusionant fins que s'aconsegueix una situació contrària, és a dir, totes les dades s'uneixen en un sol grup. En aquest cas parlarem de ***clustering*** o agrupament.

Per construir grups, es necessita un concepte de distància entre objectes i un criteri d'enllaç per establir la pertinença a un grup o un altre. Alguns dels criteris més utilitzats per mesurar la distància entre dos grups A i B són els següents:

- Enllaç simple, o *simple linkage*. Prendrem com a criteri la distància mínima entre elements dels grups:

$$min\{d(x,y) \quad |x \in A, y \in B\} \tag{12}$$

  Pot ser apropiat per trobar grups de manera no el·líptica, però és molt sensible al soroll en les dades i pot arribar a provocar l'efecte cadena. Aquest

consisteix en el fet que pot arribar a forçar la unió de dos grups, que *a priori* haurien de quedar ben diferenciats, per la circumstància que comparteixin algun element molt pròxim.

- Enllaç complet, o *complete linkage*. Prendrem com a criteri la distància màxima entre elements dels grups:

$$max\{d(x,y) \quad |x \in A, y \in B\} \tag{13}$$

  No produeix l'efecte cadena, però és sensible als valors *outliers*. No obstant això, sol donar millors resultats que el criteri simple.

- Enllaç mitjà, o *average linkage*. Prendrem com a criteri la distància mitjana entre elements dels grups:

$$\frac{1}{|A||B|}\sum_{x\in A}\sum_{y\in B} d(x,y) \tag{14}$$

  Es tracta d'un criteri que intenta atenuar els inconvenients dels dos anteriors sense acabar de resoldre'ls del tot.

- Enllaç centroide, o *centroid linkage*. La distància entre dos grups serà la distància entre els seus dos centroides. Presenta l'avantatge que el seu cost computacional és molt inferior al dels criteris anteriors, de manera que està indicat per a jocs de dades de gran volum.

Per construir un dendrograma aglomeratiu, inicialment haurem d'establir amb quina mètrica treballarem (distància euclidiana, de Gauss, de Mahalanobis...) i quin criteri d'enllaç de grups o segments utilitzarem (*simple linkage*, *complete linkage*, *average linkage*, *centroid linkage*...).

El pas següent serà considerar cada observació del joc de dades com un grup o segment en si mateix, i a partir d'aquí començarem a calcular distàncies entre grups. En aquest punt entrarem en un procés iteratiu en el qual en cada repetició fusionarem els grups més pròxims.

2.1.2. Algorismes de tipus divisiu

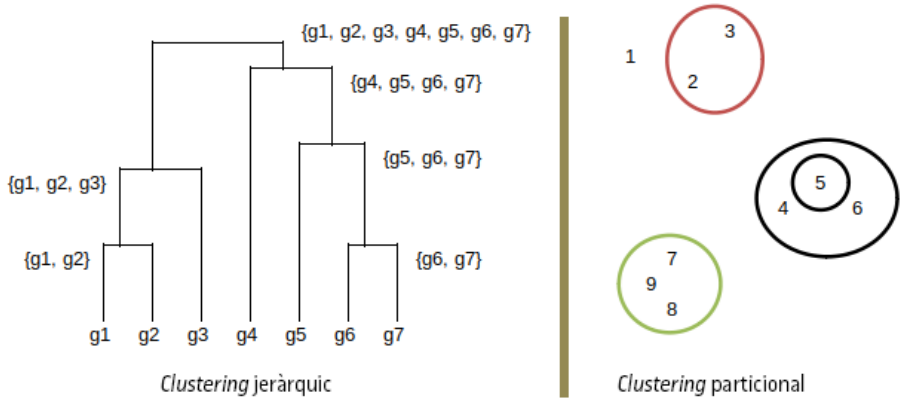
Direm que són **de tipus divisiu** quan, partint d'un grup que conté totes les dades, es procedeix a una divisió progressiva fins que s'aconsegueix tenir un grup per cada observació. En aquest cas parlarem de **segmentació**.

2.2. Algorismes particionals

Els algorismes particionals o no jeràrquics reben aquest nom perquè els segments que acaben produint no responen a cap tipus d'organització jeràrquica. En aquesta categoria d'algorismes hi trobem el ***k-means***.

A la figura 2 hi podem distingir visualment la diferència entre el *clustering* jeràrquic i el particional.

Figura 2. Tipus de *clustering*



2.2.1. *k-means*

L'algorisme *k-means*, o *k-mitjanes*, en català, està considerat com un algorisme de classificació no supervisada. Requereix que es fixin per endavant els k grups que es volen obtenir.

Suposem que disposem d'un joc de dades compost per n observacions. Per exemple, cada cas podria ser un client del qual hem seleccionat m atributs que el caracteritzen.

Anomenarem X aquest joc de dades $X = \{x_1,x_2,...,x_n\}$, on cada x_i podria ser un client amb m atributs $x_i = \{x_{i1},x_{i2},...,x_{im}\}$, com poden ser, per exemple, vendes, promocions, distància al centre de distribució logística, etc.

Per classificar el nostre joc de dades X mitjançant l'algorisme *k-means*, seguirem els passos següents:

1) D'entre les n observacions seleccionarem k, que anomenarem *llavors*, i denotarem per c_j, on $j = 1,...,k$. Cada llavor c_j identificarà el seu clúster C_j.

2) Assignarem l'observació x_i al clúster C_t quan la distància entre l'observació x_i i la llavor c_t sigui la menor entre totes les llavors.

$$d(x_i,c_t) = min\{d(x_i,c_j)\},\ j = 1,...,k \tag{15}$$

3) Calcularem els nous centroides a partir de les mitjanes dels clústers actuals.

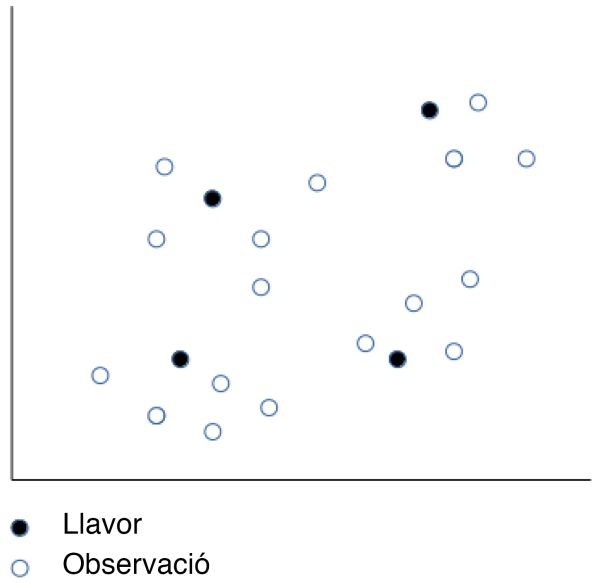
4) Com a criteri d'aturada, calcularem la millora que es produiria si assignéssim una observació a un clúster al qual no pertany actualment. Entenent per millora, per exemple, la minimització de la distància de les diferents observacions als seus respectius centres.

5) Farem el canvi que ens proporcioni la millora més gran.

6) Repetirem els passos 3, 4 i 5 fins que cap canvi no sigui capaç de proporcionar una millora significativa.

A la figura 3 hi podem veure un exemple en dues dimensions, és a dir, per a $m = 2$.

Figura 3. Llavors *k-means*

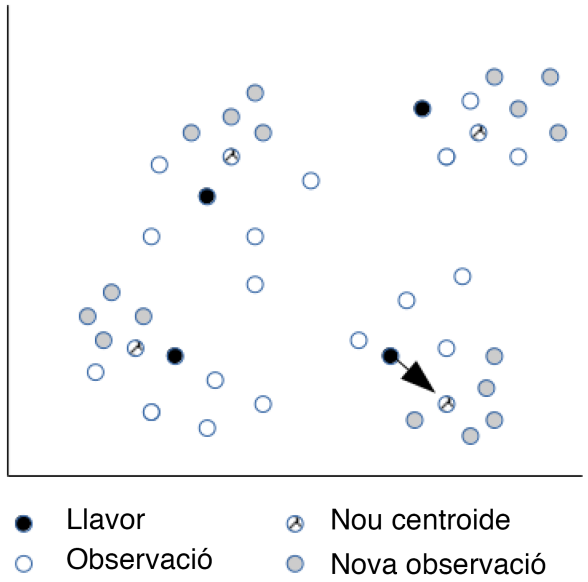


A la figura 4 hi podem apreciar visualment el mecanisme per generar nous centroides.

2.2.2. Criteris per seleccionar *k*

Un dels inconvenients que té *k-means* és el fet que requereix que s'especifiqui per endavant el valor *k* (nombre de clústers).

Figura 4. Nous centroides *k-means*

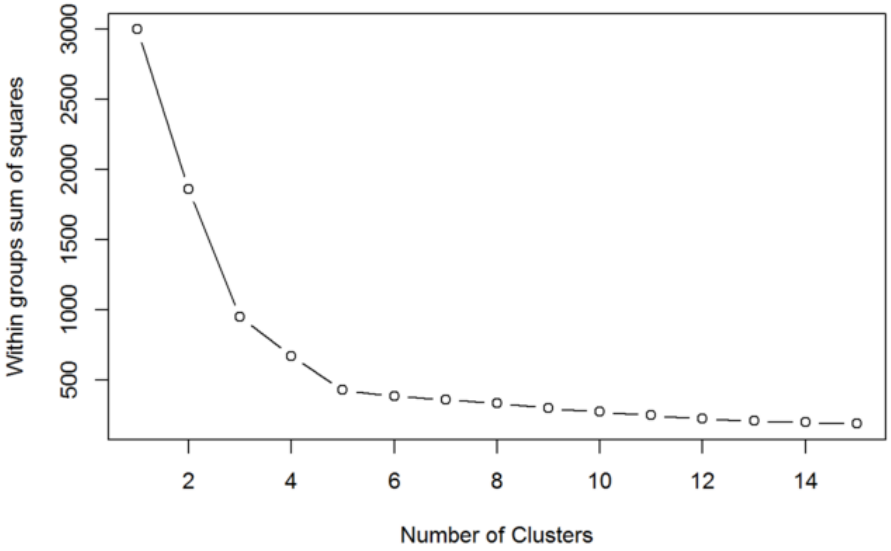


La minimització de distàncies intragrup o la maximització de distàncies intergrup es poden usar com a criteris per establir un valor adequat per al paràmetre *k*.

Els valors *k* per als quals ja no s'aconsegueixen millores significatives en l'homogeneïtat interna dels segments o l'heterogeneïtat entre segments diferents, s'haurien de descartar.

A la figura 5 hem generat un gràfic amb la suma de les distàncies intragrup que obtenim per a cada valor de *k*.

Figura 5. Elecció de *k*



Observem com, a partir de cinc segments, la millora que es produeix en la distància interna dels segments passa a ser insignificant. Aquest fet hauria d'indicar que cinc segments és un valor adequat per a *k*.

2.3. *Canopy clustering*

Podem pensar en aquesta tècnica com una generalització dels algorismes particionals.

La idea brillant que hi ha darrere aquesta tècnica és que podem reduir dràsticament el nombre de càlculs que requereixen els algorismes particionals com *k-means*, introduint un procés previ de generació de **grups superposats** (*canopies*) a partir d'una **mètrica més senzilla** de calcular (*cheapest metric*).

D'aquesta manera, només calcularem distàncies amb la mètrica inicial, més estricta i feixuga en càlculs, per als punts que pertanyen al mateix *canopy*.

Ho podríem resumir dient que, prèviament, mitjançant una mètrica simple, decidim quins punts estan definitivament lluny; per tant, per a aquests punts allunyats ja no valdrà la pena malgastar més càlculs amb una mètrica més exigent.

En realitat, el mètode del *canopy clustering* divideix el procés de segmentació en dues etapes:

- A la primera, usarem una mètrica senzilla en càlculs amb l'objectiu de generar els *canopies* o subgrups superposats de punts. A més, ho farem de manera que cada punt pugui pertànyer a més d'un *canopy* i, al seu torn, tots els punts hagin de pertànyer almenys a un *canopy*.

- A la segona, utilitzarem un mètode de segmentació tradicional, com per exemple el mètode *k-means*, però ho farem amb la restricció següent: no calcularem la distància entre punts que no pertanyin al mateix *canopy*.

Per facilitar la comprensió del mecanisme de l'algorisme, ens situarem en els dos casos extrems:

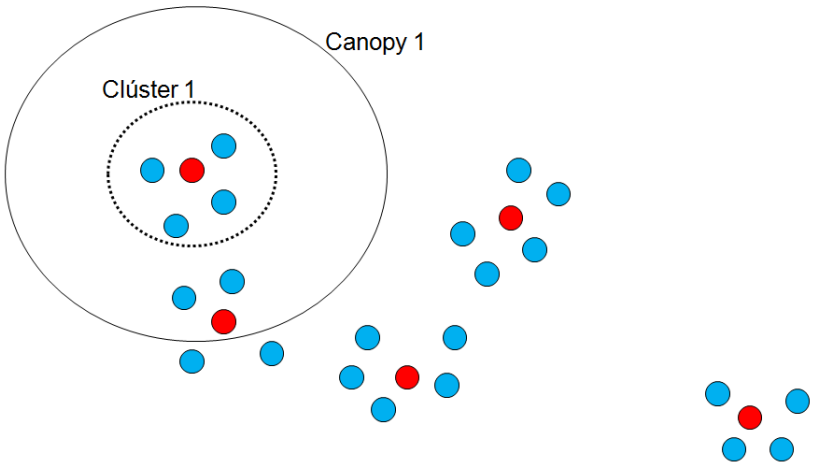
1) Suposem que, com a conseqüència de la primera etapa, el nostre univers de punts cau per complet en un sol *canopy*. Aleshores, el mètode de segmentació per *canopies* seria exactament igual al mètode de segmentació tradicional seleccionat, és a dir, *k-means* al nostre exemple.

2) Suposem que, com a resultat de la primera etapa, generem *canopies* relativament petits i amb molt poca superposició. En aquest cas, en aplicar la tècnica tradicional només dins de cada *canopy*, ens haurem estalviat un gran nombre de càlculs.

Per il·lustrar d'una manera gràfica el procés de construcció de *canopies* a partir d'una mètrica simple i el procés de construcció de clústers a partir d'una mètrica més exigent, proporcionem una sèrie de tres figures.

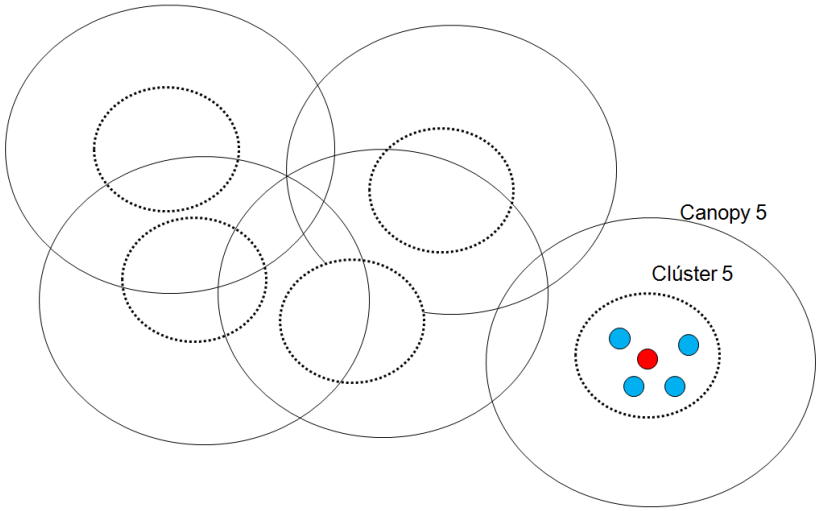
Inicialment, a la figura 6 s'hi aprecia com els *canopies* es construeixen d'una manera àmplia i amb superposicions.

Figura 6. Construcció del *canopy* 1



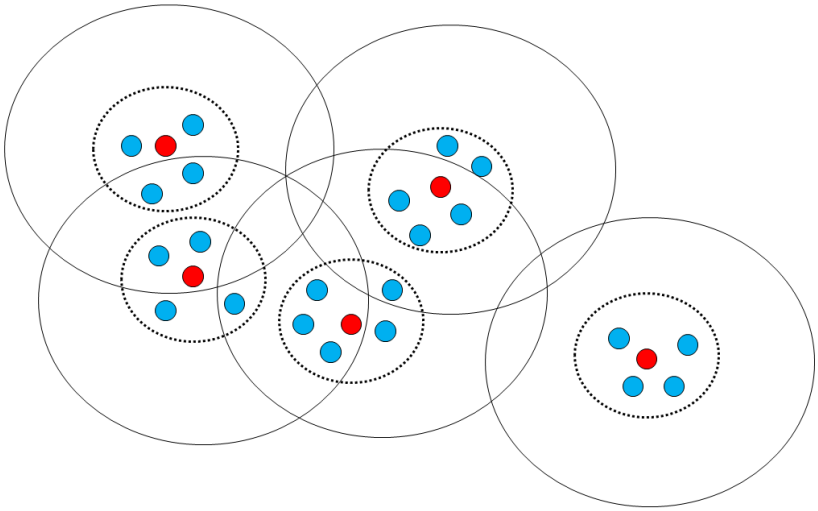
A la figura 7 hi apreciem com els clústers es calculen sense superposicions i sempre dins del *canopy* corresponent.

Figura 7. Construcció del *canopy* 5



Finalment, a la figura 8 hi podem veure el resultat d'un procés de segmentació mitjançant la tècnica del *canopy clustering*.

Figura 8. *Canopy clustering*



3. Models basats en la densitat

Els models de *clustering* basats en la densitat constitueixen una família d'algorismes que han donat molt bons resultats i, en conseqüència, han despertat l'interès de molts analistes. Aquest tipus d'algorismes s'especialitzen a identificar zones d'alta concentració d'observacions separades entre si per zones amb una menor densitat d'observacions.

Els dos algorismes més coneguts són DBSCAN i OPTICS.

3.1. Algorisme DBSCAN

Density-based Spatial Clustering of Applications with Noise, un nom complex per a un algorisme que en el fons és molt senzill. Vegem com funciona.

Inicialment, requereix que se l'informi de dos paràmetres:

- El valor ϵ (**èpsilon**): màxim radi de veïnatge.

  Considerarem que dos punts o observacions són pròxims si la distància que els separa és menor o igual a ϵ.

- El valor ***minPts***: mínim nombre de punts a l'ϵ-veïnatge d'un punt.

  Hi podem pensar com el valor que marcarà el nostre criteri de què considerem com a dens.

D'aquesta manera, DBSCAN anirà construint esferes de radi ϵ que almenys incloguin *minPts* observacions.

La lògica que segueix l'algorisme per construir els clústers o zones densament poblades és la següent:

- Es considera que un punt p és un **punt nucli**, *core point*, si almenys té *minPts* punts a una distància inferior o igual a ϵ. Dit d'una altra manera, conté *minPts* a l'ϵ-veïnatge.

- Un punt q és **assolible** des de p (*p-reachable)*, on p és nucli, si la distància entre tots dos és inferior o igual a ϵ. Dit d'una altra manera, si està dins de l'ϵ-veïnatge de p.

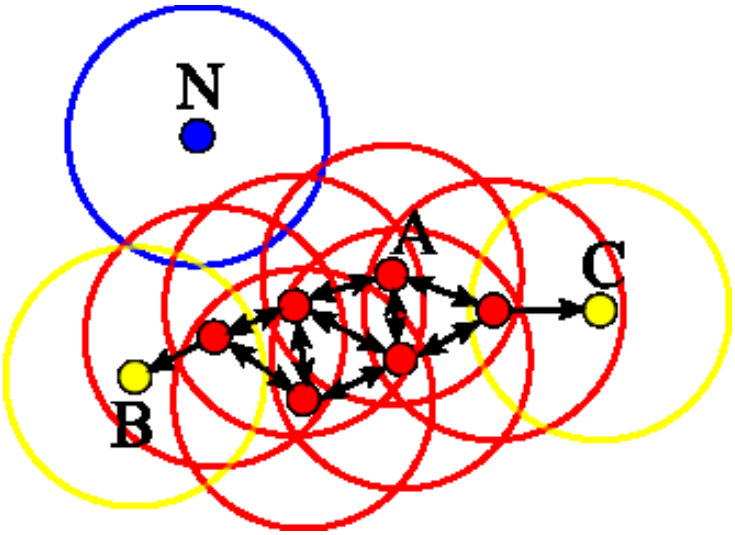
- Un punt q és assolible des de p si hi ha un camí de punts nucli que els connecta.

  Explicat més formalment, si hi ha $p_1,...,p_n$, amb $p_1 = p$ i $p_n = q$, on cada p_{i+1} és assolible per p_i i tots els $p_1,...,p_{n-1}$ són punts nucli.

- Qualsevol punt no assolible es considerarà punt extrem o *outlier*.

La figura 9 ens mostra, d'una manera esquemàtica, el procés de construcció de zones de densitat. En aquest exemple es pren *minPts* = 4.

Figura 9. Procés de construcció de zones de densitat



Font: By Chire - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=17045963

Els punts *B* i *C* corresponen a la frontera del clúster, és a dir, són punts assolibles des d'un punt nucli, però ells mateixos no són punts nucli perquè no inclouen *minPts* al seu ϵ-veïnat.

Els punts *A* són punts nucli, ja que com a mínim cadascun d'ells té 4 punts en un radi ϵ prefixat.

Finalment, el punt *N* es considera extrem o *outlier*, perquè no és assolible des de cap punt del joc de dades.

3.1.1. Avantatges i inconvenients

L'avantatge principal de DBSCAN és que és capaç d'identificar **clústers de qualsevol forma geomètrica**, no només circular, perquè només necessita que hi hagi la combinació de zones amb alta i baixa densitat de concentració de punts. És especialment bo **identificant valors extrems**. Contràriament a altres algorismes, per DBSCAN no suposa cap inconvenient treballar amb un joc de dades amb aquest tipus de valors.

DBSCAN **tampoc no requereix que prefixem el nombre de clústers** que volem que identifiqui. L'única cosa que necessita és que hi hagi zones de baixa densitat de punts per poder marcar bé les fronteres entre clústers.

Pel que fa als inconvenients, el principal és el fet d'haver de fixar com a paràmetres d'entrada els valors ϵ i *minPts*. **Encertar el valor òptim d'aquests paràmetres** requereix experiència i coneixement tant sobre l'algorisme en si com sobre el joc de dades.

3.2. Algorisme OPTICS

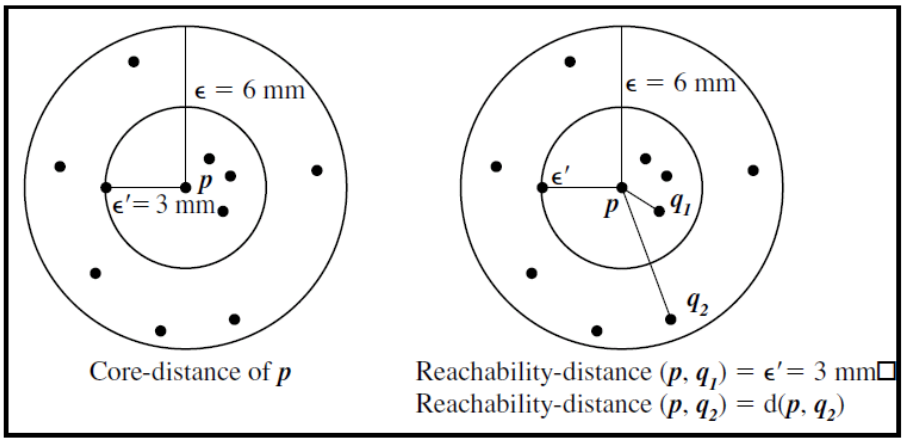
Ordering Points to Identify Cluster Structure és un algorisme que d'alguna manera generalitza DBSCAN i en resol l'inconvenient principal: els paràmetres inicials.

OPTICS requereix un radi ϵ i un criteri de densitat *minPts*, igual que DBSCAN, però en el cas d'OPTICS el valor de radi ϵ no determinarà la formació de clústers, sinó que servirà per ajudar a reduir la complexitat del càlcul en el mateix algorisme.

En realitat, OPTICS no és un algorisme que generi una proposta de clústers a partir d'un joc de dades d'entrada, com DBSCAN. De fet, el que du a terme és ordenar els punts del joc de dades en funció de la seva distància d'assolibilitat, o ***reachability-distance***, en anglès.

Per entendre bé aquest concepte nou, ens basarem en la figura 10, on hem pres *minPts* = 5.

Figura 10. Distància d'assolibilitat



La ***core-distance*** del punt *p* és el radi ϵ' mínim tal que el seu ϵ'-veïnatge conté almenys *minPts* = 5 punts.

La **distància d'assolibilitat** d'un punt q respecte d'un punt nucli (*core-point*) p serà la més llarga de les dues distàncies següents:

- *core-distance* del punt p,
- distància euclidiana entre els punts p i q, que denotarem per $d(d\ p,q)$.

Seguint amb l'exemple de la figura 10, veiem com la distància d'assolibilitat dels punts p i q_1 és la *core-distance* del punt p, perquè és més llarga que la distància euclidiana entre els punts p i q.

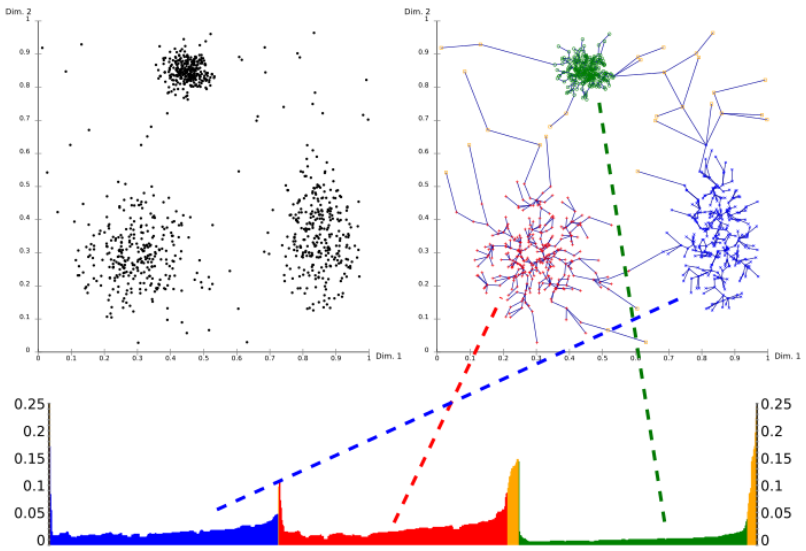
D'altra banda, la distància d'assolibilitat dels punts p i q_2 és la distància euclidiana entre ells, perquè és més llarga que la *core-distance* del punt p.

OPTICS, com a algorisme, el que ens farà és assignar a cada punt del joc de dades una distància d'assolibilitat.

Aclarits aquests conceptes bàsics, podem avançar en la comprensió de la utilitat de disposar d'aquesta ordenació. Per a això, usarem un tipus de gràfic específic per a aquest algorisme, el *reachability-plot*.

Per entendre bé què és un *reachability-plot*, observem la figura 11. Al gràfic inferior hi veiem la distància d'assolibilitat assignada a cada punt, i hi apreciem com hi ha zones amb valors alts que es corresponen amb els punts *outliers* i zones amb valors molt baixos que es corresponen amb punts situats en zones denses.

Figura 11. *Reachability-plot*



Font: By Chire - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=10293701

Fixem-nos que, a l'hora de generar els clústers, podrem decidir quina és la distància d'assolibilitat límit que ens marca què considerem com a clúster.

Podrem calibrar o ajustar aquest valor límit fins a aconseguir una generació de clústers adequada.

La possibilitat de calibrar la distància d'assolibilitat límit fa que en realitat OPTICS ens doni una ordenació de punts per distància d'assolibilitat. En conseqüència, serà el mateix analista qui podrà generar múltiples combinacions de clústers en funció del límit que es vulgui fixar.

3.2.1. Avantatges i inconvenients

DBSCAN pressuposa que la densitat que trobarà en tots els clústers és un valor constant. Per contra, OPTICS permet que el valor de densitat sigui variable en un joc de dades, precisament per l'habilitat de fixar el límit al punt que vulguem.

4. *Affinity propagation*

Es tracta d'un algorisme de *clustering* que basa la seva lògica en l'intercanvi de missatges entre els diferents punts del joc de dades.

Una diferència significativa entre un altre algorisme de *clustering* com *k-means* i *affinity propagation* és que a *k-means* comencem amb un nombre predefinit de clústers i una proposta inicial de centres potencials, mentre que a *affinity propagation* cada punt del joc de dades es tracta com un centre potencial o *exemplar*.

Com a entrada, l'algorisme ha de tenir dos jocs de dades:

- Una matriu de similituds S, on se sol prendre la distància euclidiana $s(i,k) = -(x_i - x_k)^2$ i on quedarà representat com és d'adequat que dos punts $\{i,k\}$ pertanyin al mateix clúster.

- Una relació de preferències en què reflectirem quins punts considerem més apropiats per exercir el paper d'exemplars.

A partir d'aquí, *affinity propagation* centrarà el seu procediment en dues matrius:

- La matriu de responsabilitats R: $r(i,k)$ ens indicarà com encaixa de bé el punt k com a punt exemplar de i.

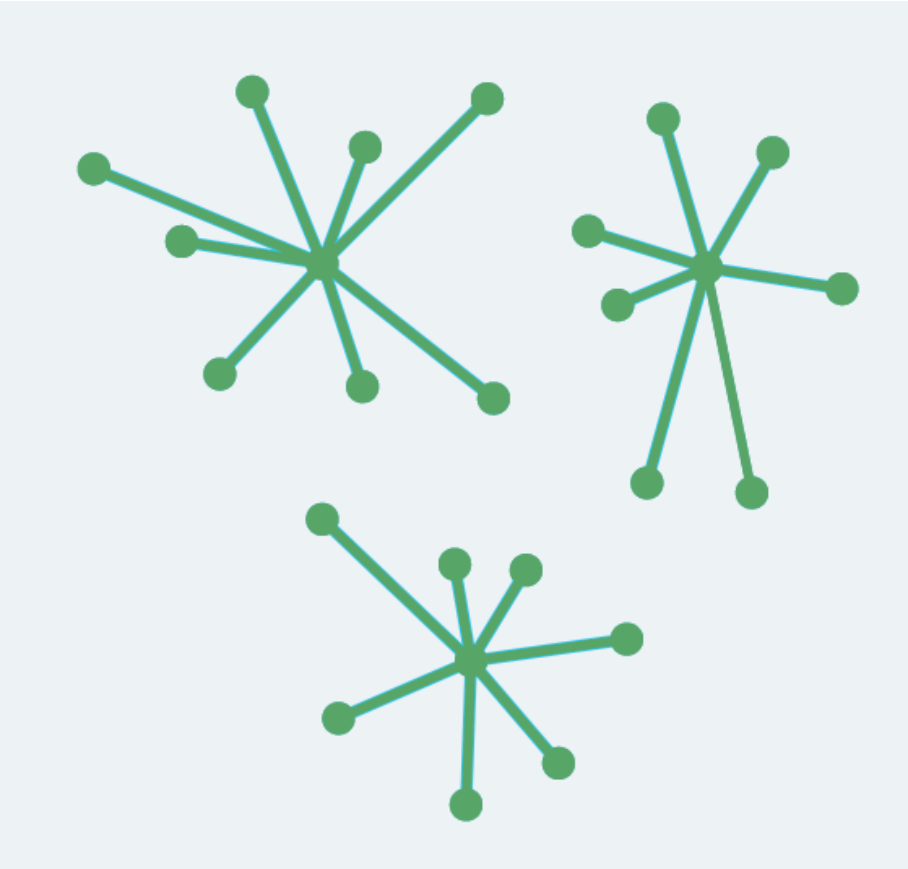
- La matriu de disponibilitat A: $a(a,i\ k)$ ens indicarà com d'adequat és per al punt i considerar k com el seu exemplar.

Totes dues matrius es poden interpretar com a probabilitats logarítmiques i, per aquest motiu, en considerarem els valors en negatiu.

A la figura 12 hi veiem com els diferents punts del joc de dades es relacionen amb el seu punt exemplar de referència.

El mecanisme intern de l'algorisme consisteix en un intercanvi de missatges entre punts. Aquests s'alineen entre ells formant clústers locals, de manera que, a mesura que avança el procés iteratiu, continuen intercanviant missatges fins que s'arriben a formar clústers més grans i estables.

Figura 12. *Affinity propagation*



A l'expressió formal de la matriu de responsabilitats R (equació 16), hi observem com reflecteix la prova acumulada de com és d'adequat el punt k per servir com a exemplar per al punt i, tenint en compte uns altres possibles exemplars per al punt i.

$$r(i,k) = s(i,k) - max_{k\prime \notin \{i,k\}}\{a(i,k\prime) + s(i,k\prime)\} \tag{16}$$

D'altra banda, veiem com l'expressió formal de la matriu de disponibilitat *A availability matrix* (equació 17) reflecteix la prova acumulada de com d'apropiat seria per al punt i triar el punt k com el seu exemplar, tenint en compte el suport d'altres punts perquè el punt k fos un exemplar.

Es divideix en dues parts:

$$a(i,k) = min\{0,r(k,k) + \sum_{i\prime \notin \{i,k\}} max\{0,r(i\prime,k)\}\} \tag{17a}$$

$$a(k,k) = \sum_{i\prime \neq k} max\{0,r(i\prime,k)\} \tag{17b}$$

- La primera, per als punts fora de la diagonal de A, és a dir, per als missatges que van d'un punt a un altre.

- La segona, per als punts a la diagonal de A, és a dir, per al missatge de disponibilitat que un punt s'envia a si mateix.

Affinity propagation, des que va aparèixer el 2007, ha anat guanyant adeptes com un dels millors algorismes en tasques de *clustering* i pel seu bon rendiment en multitud de situacions.