Klaszterezés

Kovács Máté

BME

2012. március 22.

Intuitív meghatározás

Adott dolgokból halmazokat – *klasztereket* – alakítunk ki úgy, hogy az egy klaszterbe tartozók jobban hasonlítsanak egymásra, mint más klaszterekben levőkre.

Formális definíció

Klaszterezés tehát az S elemhalmaz részhalmazainak egy $\mathcal{C}$ kollekciója:

$$\mathcal{C} \subset \mathcal{P}(S)$$
$$\mathcal{C} = \{C_1, \ldots, C_k\}$$

Egy klaszterezés lehet

- szigorú (vs átlapoló)
  $C_1 \neq C_2 \Longrightarrow C_1 \cap C_2 = \emptyset$
- outliereket kezelő
  $\bigcup C_i \neq S$
- hierarchikus
  $C_1 \cap C_2 \neq \emptyset \Longrightarrow C_1 \subseteq C_2 \vee C_2 \subseteq C_1$
- altér-klaszterezés

Osztályozás

- Az adatpontok jellemzése a cél.
- Az osztályok előre adottak.
- Rendelkezésre áll tanítóhalmaz $\rightarrow$ felügyelt tanulás.

Klaszterezés

- Az adathalmaz jellemzése a cél.
- Az osztályok ismeretlenek.
- Nincs tanítóhalmaz $\rightarrow$ felügyelet nélküli tanulás.

Biológia

- Filogenetikai fák automatikus generálása.
- Gének csoportosítása a kifejeződési jegyeik alapján.
- Hasonló gének csoportosítása az emberi genomban.
- Emberi populációk vizsgálata genomok klaszterezése alapján.

Gazdaságtudomány

- Piaci szegmentáció.
- Termékcsoportok azonosítása.
- Portfóliók kockázatcsökkentése.

Információtechnológia

- Képfeldolgozásban objektumok elhatárolása.
- Genetikai algoritmusok javítása.
- Online szociális hálók adatbányászata.
- Online ajánlórendszerek.

„Hozzávalók”

elméletben:

- hasonlósági függvény
- klasztermodell

gyakorlatban:

- algoritmus

Hasonlósági függvény

A hasonlóság inverzét, a különbözőséget definiáljuk:

$$d : S \times S \to \mathbb{R}_0^+$$

Megköveteljük, hogy metrika legyen, vagyis teljesüljenek a következők

- egybeesés
  $d(x, y) = 0 \iff x = y$
- szimmetria
  $d(x, y) = d(y, x)$
- háromszög-egyenlőtlenség
  $d(x, y) \leq d(x, z) + d(z, y)$

Reprezentálás súlyozott gráfként

Tekinthetjük úgy, hogy a távolságokat egy (irányítatlan) teljes gráf éleihez rendeljük hozzá:

$$G = (V, E)$$
$$V = S$$
$$E = S \times S$$
$$d : E \to \mathbb{R}_0^+$$

Néhány algoritmus nem a teljes gráfot, hanem a G_k-val jelölt *k-legközelebbi-szomszéd-gráf*ot használja, amely minden pontra csak annak k legközelebbi szomszédjába futó éleket tartalmazza.

Gyakori távolságfüggvények

Tipikusan $S \subset \mathbb{R}^n$

- háztömb (Manhattan)
  $$d(x, y) := \sum_{i=1}^{n} |x_i - y_i|$$
- euklideszi
  $$d(x, y) := \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$
- Mahalanobis
  $$d(x, y) := \sqrt{(x - y)^{\mathsf{T}} \cdot \Sigma^{-1} \cdot (x - y)}$$
  $$\Sigma = \operatorname{cov}(S) = \mathbb{E}\left[(S - \mu) \cdot (S - \mu)^{\mathsf{T}}\right]$$
  $$\mu = \mathbb{E}[S]$$

Elvárások

1. skálafüggetlen

   Invariáns a távolságfüggvény pozitív konstanssal való szorzására.
   $\forall \alpha \in \mathbb{R}^+ : \quad \mathcal{F}(S, \alpha d) = \mathcal{F}(S, d)$

2. gazdag

   Minden felosztás előállítható alkalmas távolságfüggvényt választva.
   $\forall \mathcal{C} \subset \mathcal{P}(S) : \quad \exists d : S \times S \to \mathbb{R}_0^+ : \quad \mathcal{F}(S, d) = \mathcal{C}$

3. konzisztens

   Invariáns a klaszteren belüli távolságok csökkentésére, illetve a klaszterköziek növelésére.

4. finomítás-konzisztens

   Mint az előző, csak megengedjük, hogy klasztereket részekre bontson.

Elméleti korlátok

A következő eredmények Jon Kleinberg nevéhez fűződnek.

1. Nem létezik skálafüggetlen, gazdag és konzisztens $\mathcal{F}$ klaszterező függvény.
2. Bármely két fenti tulajdonsághoz létezik velük rendelkező $\mathcal{F}$ klaszterező függvény.
3. Az első tétel a konzisztenciát a – gyengébb – finomítás-konzisztencia fogalmára cserélve is igaz.
4. Ha nem követeljük meg, hogy a *mind-külön* felosztás is előálljon, akkor létezik skálafüggetlen, gazdag és finomítás-konzisztens klaszterező függvény.

Klasszikus mértékek

- Legnagyobb klaszterátmérő:

$$f(\mathcal{C}) = \max_{C \in \mathcal{C}} D_{max}(C), \quad D_{max}(C) = \max_{x,y \in C} d(x,y)$$

- Centrális hibák összege:

$$f(\mathcal{C}) = \sum_{C \in \mathcal{C}} E(C), \quad E(C) = \sum_{x \in C} d(x, \mu_C)$$

- k-klaszter:

$$f(\mathcal{C}) = \sum_{C \in \mathcal{C}} D_{sum}(C), \quad D_{sum}(C) = \sum_{x,y \in C} d(x,y)$$

Klasszikus mértékek (folyt.)

- k-medián:

  Válasszunk k darab reprezentáns elemet úgy, hogy az összes többi pontra a legközelebbi reprezentánstól mért távolság összege minimális legyen.

- k-center:

  Mint a k-medián, csak összeg helyett maximummal.

Klasszikus mértékek hiányosságai

- Csak elliptikus klasztereket hoznak létre.
- A klaszterek átmérőjét korlátozzák.
- Az outlierekre érzékenyek.
- A gyakorlatban nem alkalmazhatók sikerrel.

Konduktancia alapú mérték

Térjünk vissza a hasonlóságfüggvényre:

$$w(x, y) := d^{-1}(x, y)$$

Arra a kérdésre keressük a választ, hogy $k = 2$ esetén hogyan járjunk el.

Definiáljuk (a gráf-reprezentáción) egy $(T, V - T)$ vágás *kiterjedését*:

$$\varphi(T) := \frac{w(T, V - T)}{\min(|T|, |V - T|)}$$

ahol $w(T, V - T)$ az „átvágott” élek összsúlya.

Ezt minimalizálva a számláló biztosítja, hogy alacsony hasonlóság mentén vágunk, a nevező pedig azt, hogy a két klaszter közel azonos méretű.

Konduktancia alapú mérték (folyt.)

Hogy a többitől nagyon elütő pontok kevésbé befolyásolják az egyensúlyi tényezőt, módosítsuk a kiterjedés definícióját. Ez a konduktancia:

$$\phi(T) := \frac{w(T, V - T)}{\min(a(T), a(V - T))}$$

$$a(T) := \sum_{x \in T, y \in V} w(x, y)$$

Egy klaszter konduktanciája a $(T, C - T)$ vágásai konduktanciáinak minimuma, a klaszterezésé pedig a halmazai konduktanciáinak minimuma legyen:

$$\phi(C) := \min_{T \subseteq C} \phi(T)$$

$$\phi(\mathcal{C}) := \min_{C \in \mathcal{C}} \phi(C)$$

A konduktanciát maximalizálni szeretnénk:

$$f(\mathcal{C}) = -\phi(\mathcal{C})$$

A naív algoritmus

Számítsuk ki a célfüggvényt minden lehetséges klaszterezésre, és ez alapján válasszuk ki az optimálisat:

$$\mathcal{C}_{\text{opt}} = \arg\min_{\mathcal{C}} f(\mathcal{C})$$

Egy n-elemű halmaz k darab (nemüres) részre történő lehetséges felosztásainak számát a másodfajú Stirling-számok adják meg:

$$\left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n$$

Például:

$$\left\{ {100 \atop 5} \right\} \approx 6.5 \cdot 10^{67} \quad \text{(65 unvigintillió)}$$

Értékelési szempontok

- skálázhatóság
- előzetes ismeretek
- zaj és outlierek hatása
- sorrendérzékenység
- dimenzió
- értelmezhetőség

Centroid-módszerek

- A klaszterek számát előre meg kell mondanunk.
- A klasztereket reprezentáns pontokkal jelölik ki.
- Egy kezdeti felosztást finomítanak iteratívan.
- Mohó lépésekben haladnak; lokális optimumban is megállhatnak.
- Érdemes őket többször futtatni különböző kezdeti felosztásokon.

k-közép

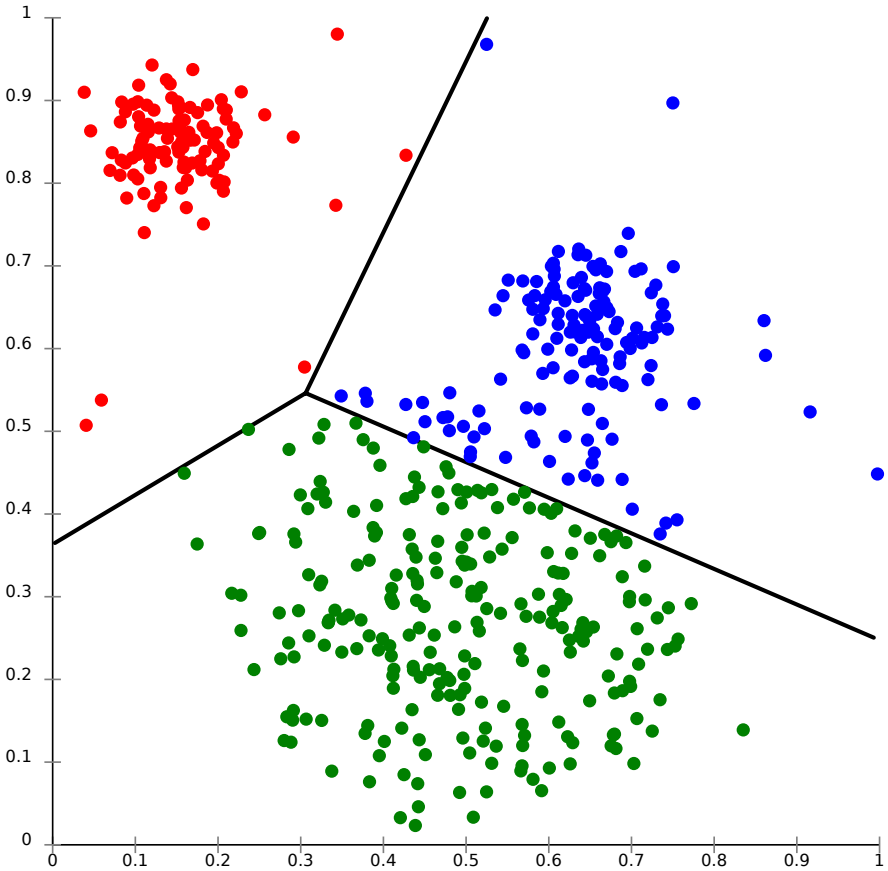
Minden pont a hozzá legközelebbi reprezentáns klaszterébe tartozik.

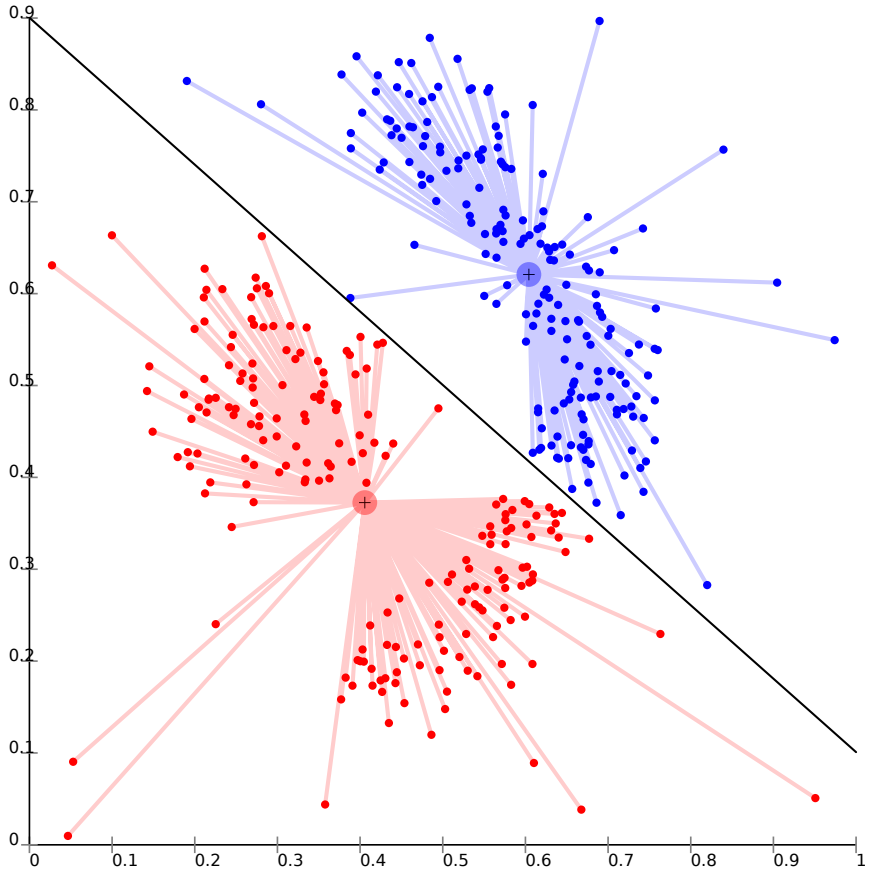
(Minden klaszter a reprezentánsának Voronoi-cellájába eső elemekből áll.)

Az iterációs lépés minden reprezentánst a klaszterének átlagába helyez át, majd újraszámítja a felosztást.

- Egy lépés futásideje $O(k \cdot n)$.
- Csak vektortéren (affin téren) van értelmezve.

k-közép





k-medoid algoritmusok

- A k-közép algoritmus javításai.
- Reprezentánsaik mindig adatpontok is (medoidok).
- Nem csak vektortéren működnek.
- Kevésbé érzékenyek az outlierekre.

PAM

Partitioning Around Medoids

Az iteratív lépés minden (x_m, x) medoid-nemmedoid párra megvizsgálja, hogy felcserélésük hogyan változtatná a hibát.

Ha nincs csökkentő pár, akkor megáll. Egyébként mohón választ, majd újraszámolja a felosztást.

- Egy lépés futásideje $O\left(k \cdot (n-k)^2\right)$.
- Nagy adathalmazokon nem használható.

CLARA, CLARANS

A PAM módosításai: nem vizsgálnak meg minden (x_m, x) párt.

CLARA:

A medoidokat csak egy n'-elemű véletlen mintából választhatja.

- Egy lépés futásideje $O(k \cdot (n' - k) \cdot (n - k))$.

CLARANS:

Egyetlen véletlenszerűen választott párt vizsgál minden lépésben.

- Egy lépés futásideje $O(n - k)$.

Hierarchikus módszerek

- A kimenetük klaszter-hierarchia.
- Két fő típusuk van: egyesítgető, osztogató.
- Lentről felfelé építenek, vagy fentről lefelé bontanak.
- Mohók; lokális optimumban ragadhatnak.

Single-, Complete-, Average Linkage

Egyesítgető eljárások.

Egymástól csak használt klasztertávolság-függvényeikben különböznek.

Single Linkage:

$$d_{min}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$

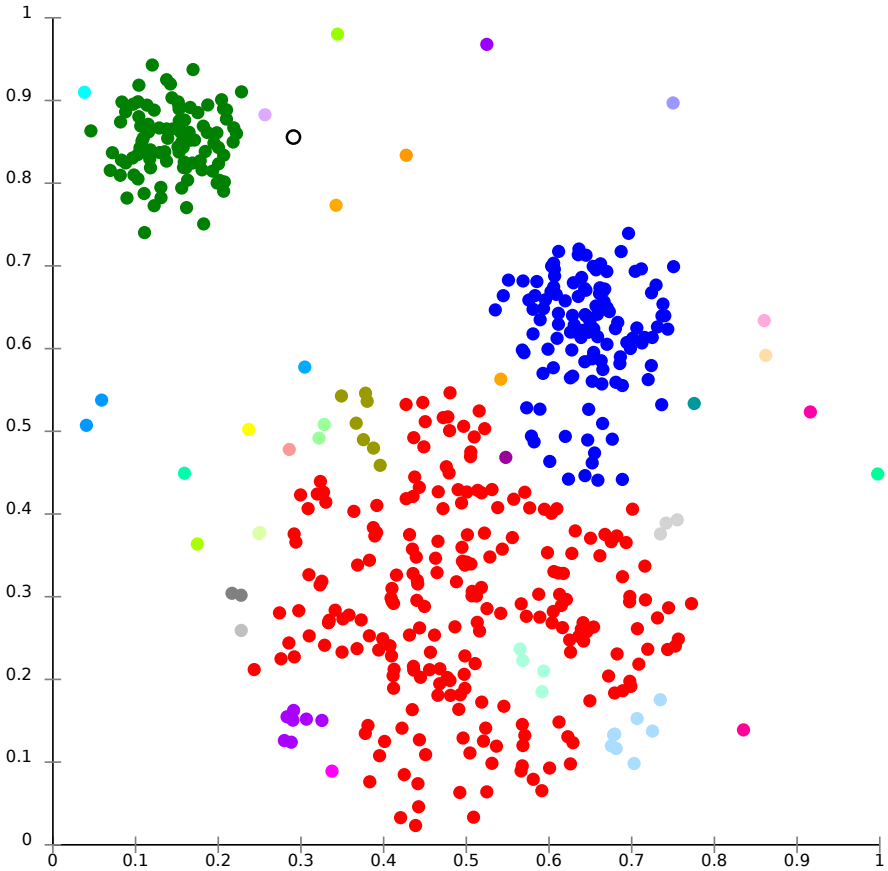
Complete Linkage:

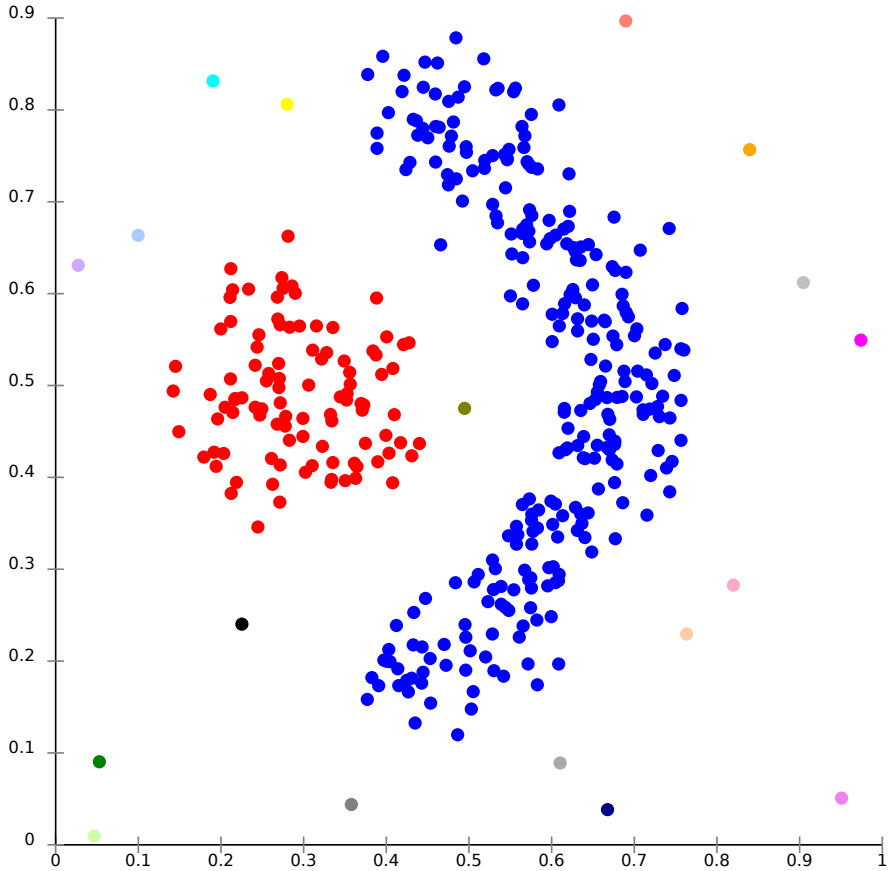
$$d_{max}(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$$

Average Linkage:

$$d_{avg}(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

Single Linkage





BIRCH

- Balanced Iterative Reducing and Clustering using Hierarchies
- Nagyon-*nagyon* nagy adathalmazokhoz.
- Klaszter-reprezentánsok: $\left(|C|, \sum x, \sum |x|^2\right)$
- Elágazás-korlátozott fa, átmérőkorlátozott klaszterek.
- Az első outliereket expliciten kezelő algoritmus volt.
- Többfázisú algoritmus.

CURE

- Clustering Using REpresentatives
- Egy klaszter jellemzésére (maximum) c darab reprezentánst használ.
- Egyesítéskor sorra választ c legtávolabbi pontot a középponttal kezdve.
- Az új reprezentánsokat a középpontjuk felé húzza (outlierek ellen).
- Többfázisú algoritmus.
- A második fázisban számítja ki a tényleges felosztást.

Sűrűség-alapú módszerek

- A (valamilyen értelemben) sűrű régiók alkotják a klasztereket.
- Nem csak elliptikus klasztereket találnak.
- Topológiai fogalmakon alapul a működésük.
- Outlierek felderítésére jól használhatóak.

DBSCAN

Egy $x \in S$ adatpont *belső* pont, ha $|N_r(x)| \geq m$.

Az y pont *elérhető* x-ből ($x \to y$), ha x belső pont és $d(x, y) \leq r$, vagy $\exists z : x \to z \to y$.

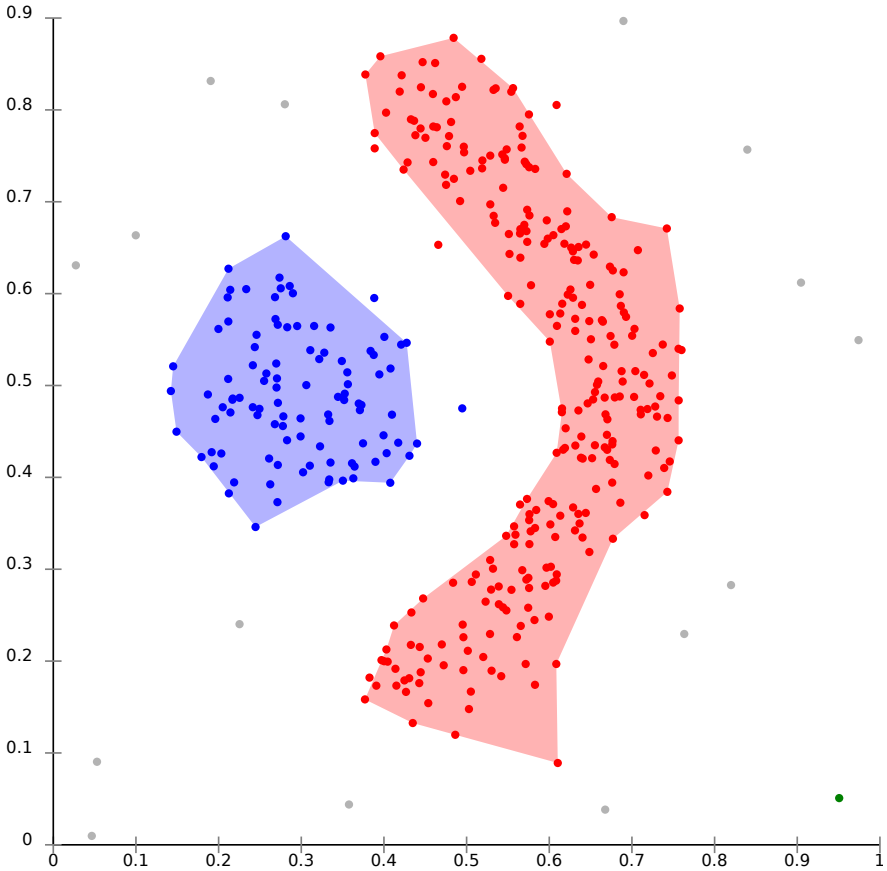
Az $x, y \in S$ pontok *összekötöttek* ($x \longleftrightarrow y$), ha $\exists z : z \to x \vee z \to y$.

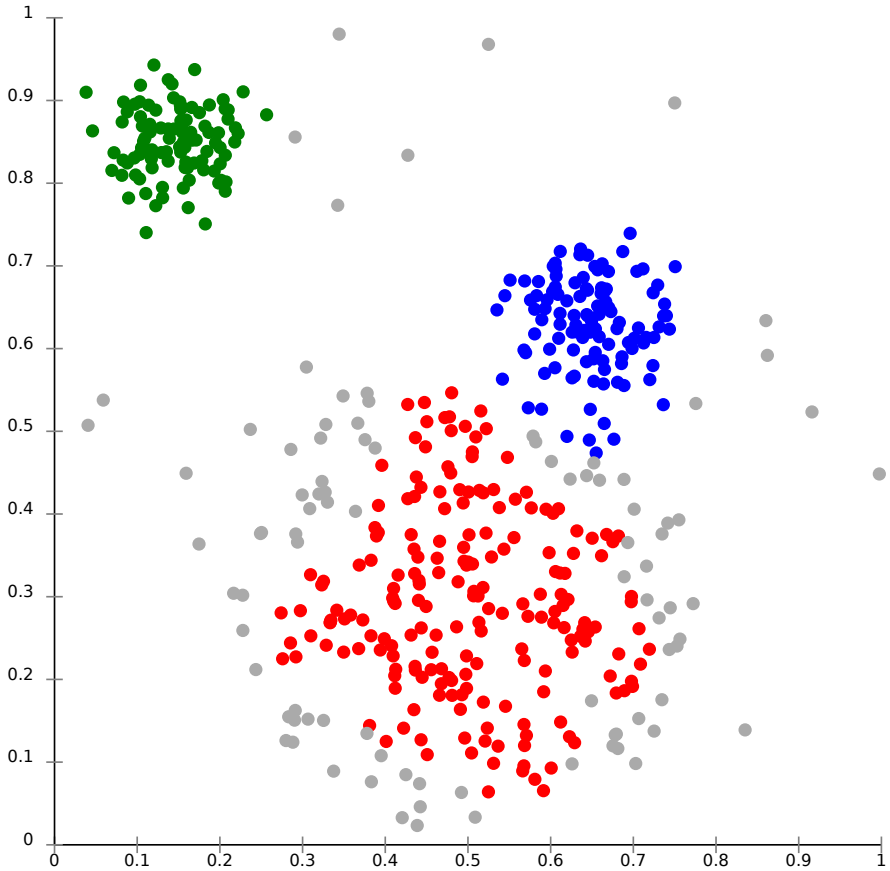
Klasztermodell:

1. $x \in C, x \to y \quad \Longrightarrow \quad y \in C$
2. $x, y \in C \quad \Longrightarrow \quad x \longleftrightarrow y$

Az egyetlen klaszterbe sem tartozó pontok az outlierek.

DBSCAN


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Emlékeztető

- Nincs csodafegyver.
- A megfelelő távolság- és klasztermodell az alkalmazástól függ.
- Az adatsor jellemzőit figyelembe véve válasszunk algoritmust.

Kérdések

Köszönöm a figyelmet!