Adatbányászat: Klaszterezés Haladó fogalmak és algoritmusok

9. fejezet

Tan, Steinbach, Kumar

Bevezetés az adatbányászatba

előadás-fóliák

fordította

Ispány Márton

Logók és támogatás

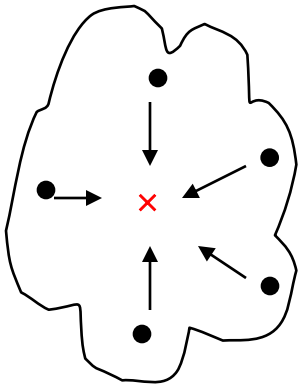
A tananyag a TÁMOP-4.1.2-08/1/A-2009-0046 számú Kelet-magyarországi Informatika Tananyag Tárház projekt keretében készült. A tananyagfejlesztés az Európai Unió támogatásával és az Európai Szociális Alap társfinanszírozásával valósult meg.

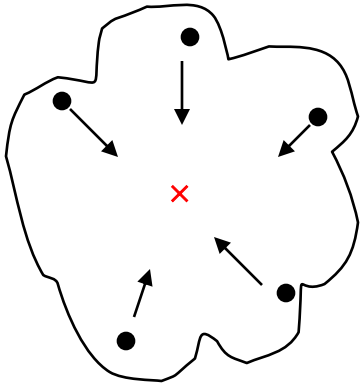
Újratárgyalt hierarchikus klaszterezés

- Beágyazott klaszterek létrehozása
- Az összevonó klaszterező algoritmusok annak függvényében változnak, hogy hogyan számoljuk két klaszter hasonlóságát.
  - MIN (egyszerű kapcsolás): zajra és kiugró pontokra érzékeny
  - MAX/ÁTLAG: nem működik jól nem gömb alakú klaszterekre
  - CURE algoritmus megpróbálja mindkét problémát kezelni
- Gyakran a közelségi mátrix kiszámításával indul
  - A gráf alapú algoritmusok egy típusa

CURE: Egy másik hierarchikus megközelítés

- Több pontot használ egy klaszter reprezentálására.





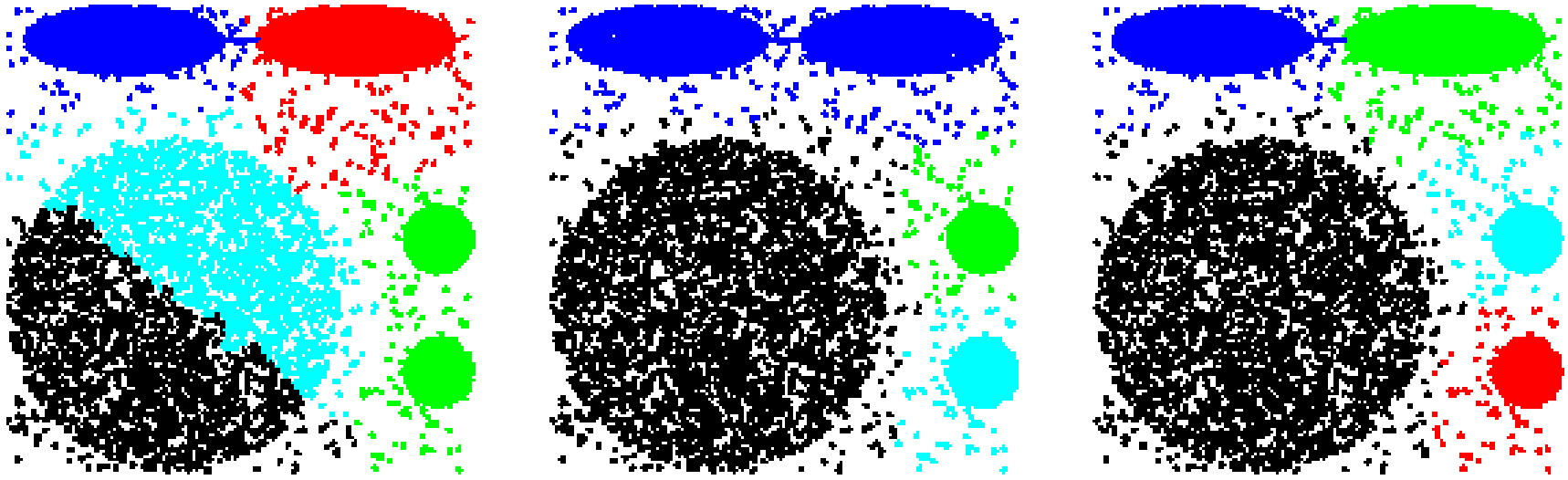
- Úgy kapjuk meg a reprezentatív pontokat, hogy egy klaszterből konstans számú pontot választunk ki, majd a klaszter középpontja felé zsugorítjuk őket.
- A klaszter hasonlóság a különböző klaszterek legközelebbi reprezentatív pontjai közötti hasonlóság.

CURE

- Zsugorítsuk össze a reprezentatív pontokat a középpont felé haladva, hogy elkerüljük a zajból és a kiugró értékekből származó problémákat.

- A CURE jobban tudja kezelni a tetszőleges alakú és méretű klasztereket.

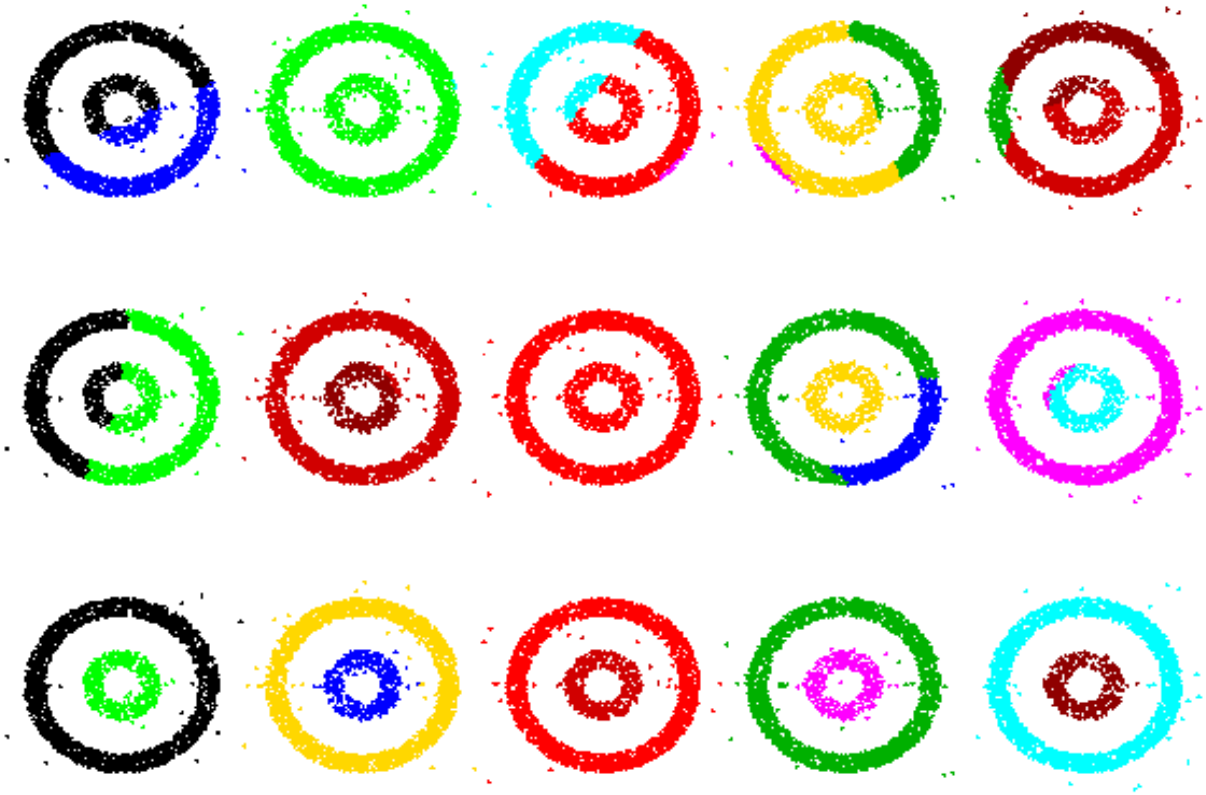
Kisérleti eredmények: CURE



a) BIRCH b) MST METHOD c) CURE

A kép a *CURE*, Guha, Rastogi, Shim, c. munkából származik.

Kisérleti eredmények: CURE



a) BIRCH

(centroid)

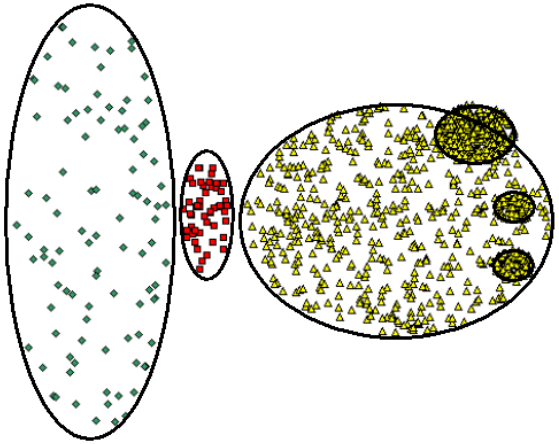
b) MST METHOD

(single link)

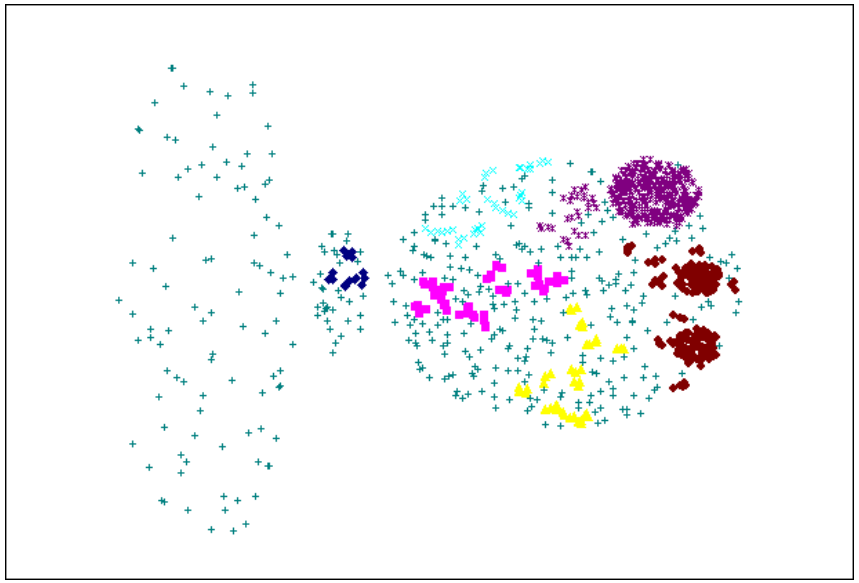
c) CURE

A kép a *CURE*, Guha, Rastogi, Shim, c. munkából származik.

A CURE nem képes kezelni a különböző sűrűségeket



Eredeti pontok



CURE

Gráf alapú klaszterezés

- A gráf alapú klaszterezés a közelségi gráfot használja.
  - Induljunk ki a közelségi mátrixból.
  - Tekintsünk minden pontot egy gráf egy csúcsának.
  - Minden két csúcs közötti élnek van egy súlya, amely a két pont közötti közelség (távolság).
  - Kezdetben a közelségi gráf teljesen összefüggő.
  - MIN (egyszerű kapcsolás) és MAX (teljes kapcsolás) az ebből a gráfból való kiindulásnak is tekinthető.
- A legegyszerűbb esetben a klaszterek a gráf összefüggő komponensei lesznek.

Gráf alapú klaszterezés: ritkítás

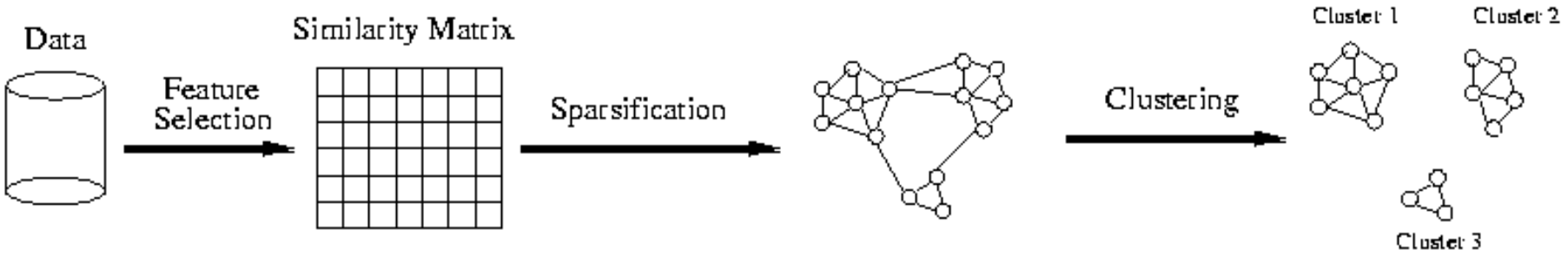
- A feldolgozandó adattömeget drámaian csökkenthetjük.
  - A ritkítás a közelségi mátrix több mint 99%-t eliminálja.
  - Az adatok klaszterezéséhez szükséges idő drámaian csökken.
  - Növekszik a kezelhető méretű problémák száma.

Gráf alapú klaszterezés: ritkítás

- A klaszterezés jobban működhet utána
  - A ritkítási módszerek megőrzik egy pont legközelebbi társaihoz való kapcsolatot míg a kevésbé hasonló pontokhoz való kapcsolatot megszüntetik.
  - Egy pont legközelebbi társa hajlamos ugyanahhoz az osztályhoz tartozni, mint ahova maga a pont tartozik.
  - Ez csökkenti a zaj és kiugró értékek hatását és erősíti a klaszterek közötti megkülönböztetést.

- A ritkítás megkönnyíti a gráf particionáló algoritmusok vagy azokon alapuló más algoritmusok használatát.
  - Chameleon és Hipergráf alapú klaszterezés

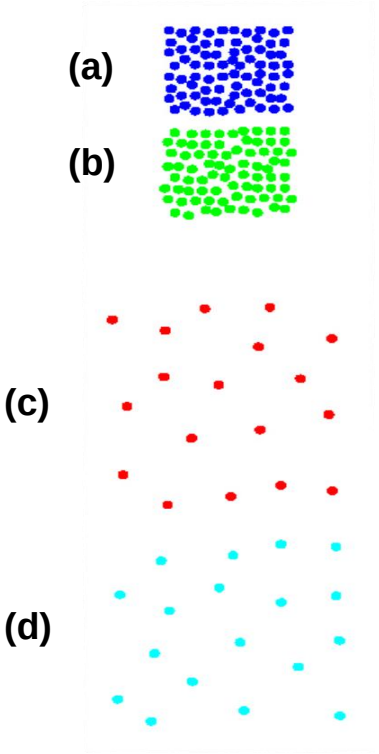
Ritkítás a klaszterezési eljárásban



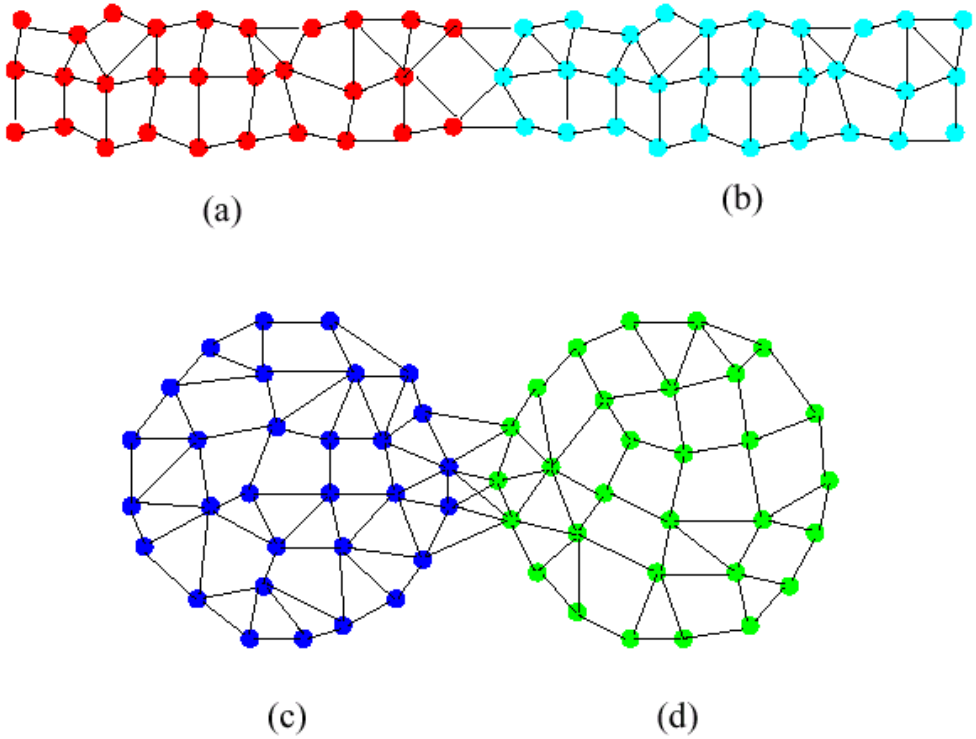
Az összevonó eljárások korlátai

- A hierarchikus klaszterező algoritmusok ismert összevonó eljárásai természetükből fakadóan statikusak.
  - MIN vagy CURE:
    - két klaszter összevonása a *közelségükön* (vagy minimális távolságukon) alapszik
  - CSOPORT-ÁTLAG:
    - két klaszter összevonása az *összekapcsolhatóságukon* alapszik

Az összevonó eljárások korlátai



A közelségi eljárás (a)-t és (b)-t vonja össze



Az átlagos kapcsolás eljárás (c)-t és (d)-t vonja össze

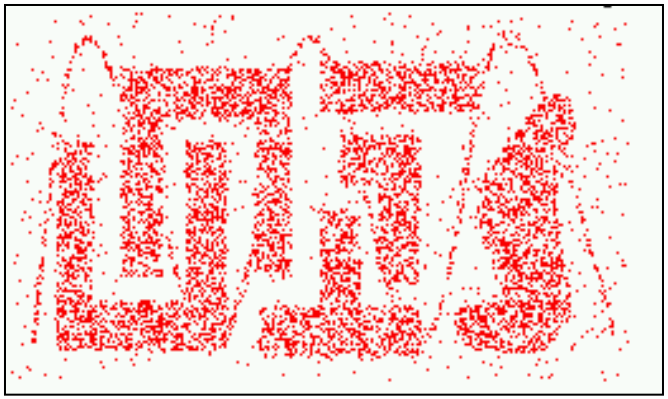
Chameleon: Klaszterezés dinamikus modellezéssel

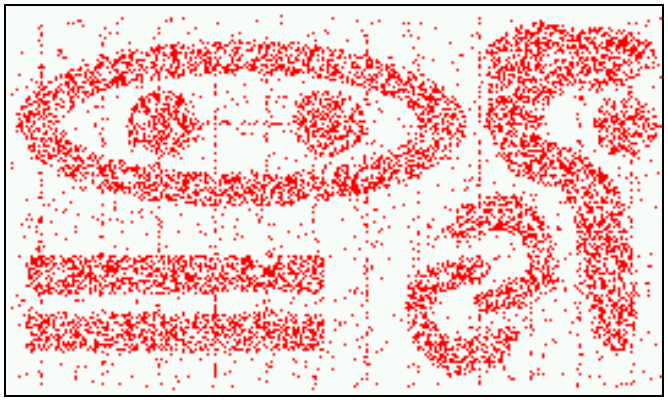
- Alkalmazkodjunk az adatok természetéhez, hogy természetes klasztereket találjunk.
- Használjunk egy dinamikus modellt a klaszterek közötti hasonlóság mérésére.
  - Fő tulajdonság a klaszterek relatív közelsége és összefüggősége.
  - Két klasztert akkor vonunk össze, ha az eredményül kapott klaszter rendelkezik az egyesítendő klaszterek bizonyos *tulajdonságaival* .
  - Az összevonó eljárás megőrzi az *önhasonlóságot .*

- Egyik alkalmazási terület a térbeli adatok vizsgálata.

Térbeli adatállományok jellemzői

- **A klasztereket a tér sűrűn lakott tartományaiként definiáljuk.**
- **A klaszterek tetszőleges alakúak, irányultságúak és nem egyenletes méretűek lehetnek.**
- **Különbség a klaszterek közötti sűrűségben és ingadozás a klasztereken belüli sűrűségben.**
- **Speciális jelenségek (*csíkok*) és zaj megléte.**





Egy klaszterező algoritmusnak kezelnie kell a fenti jellemzőket és minimális beavatkozást kell, hogy igényeljen.

Chameleon: Lépések

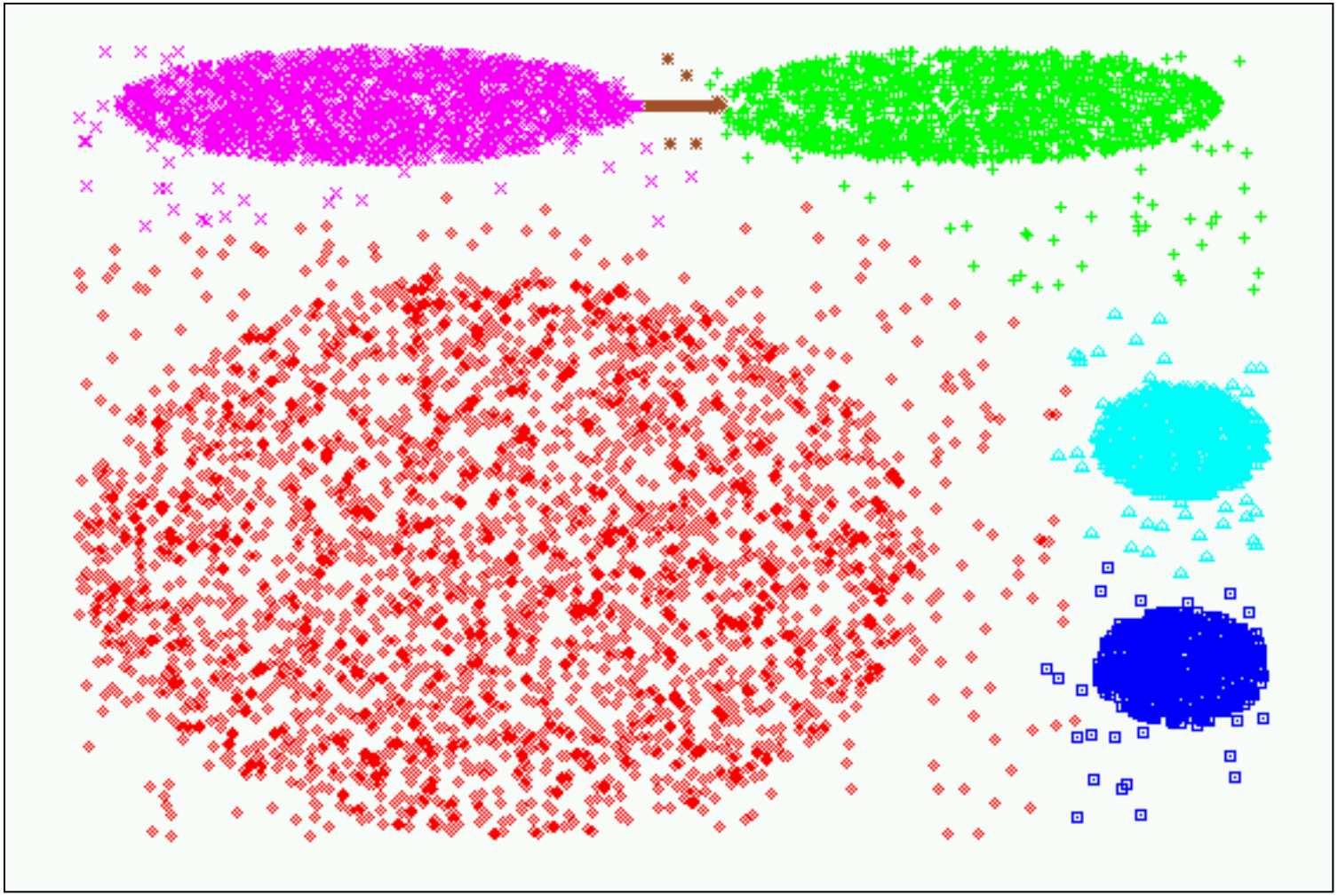
- Előfeldolgozó lépés:
  Szemléltessük az adatokat egy gráffal.
  - Egy adott ponthalmazra állítsuk elő a k legközelebbi társ (k-NN) gráfot, hogy megragadjuk egy pont és k legközelebbi társának a kapcsolatát.
  - A szomszédság fogalmát kezeljük dinamikusan (akkor is a ha a tartomány ritka).

- 1. fázis: Használjunk egy több-szintű gráf particionáló algoritmust, hogy megtaláljuk az összefüggő élek legtöbb elemszámú klasztereit.
  - Minden klaszter egy ,,igazi" klaszter legtöbb pontját fogja tartalmazni, azaz annak egy részklasztere lesz.

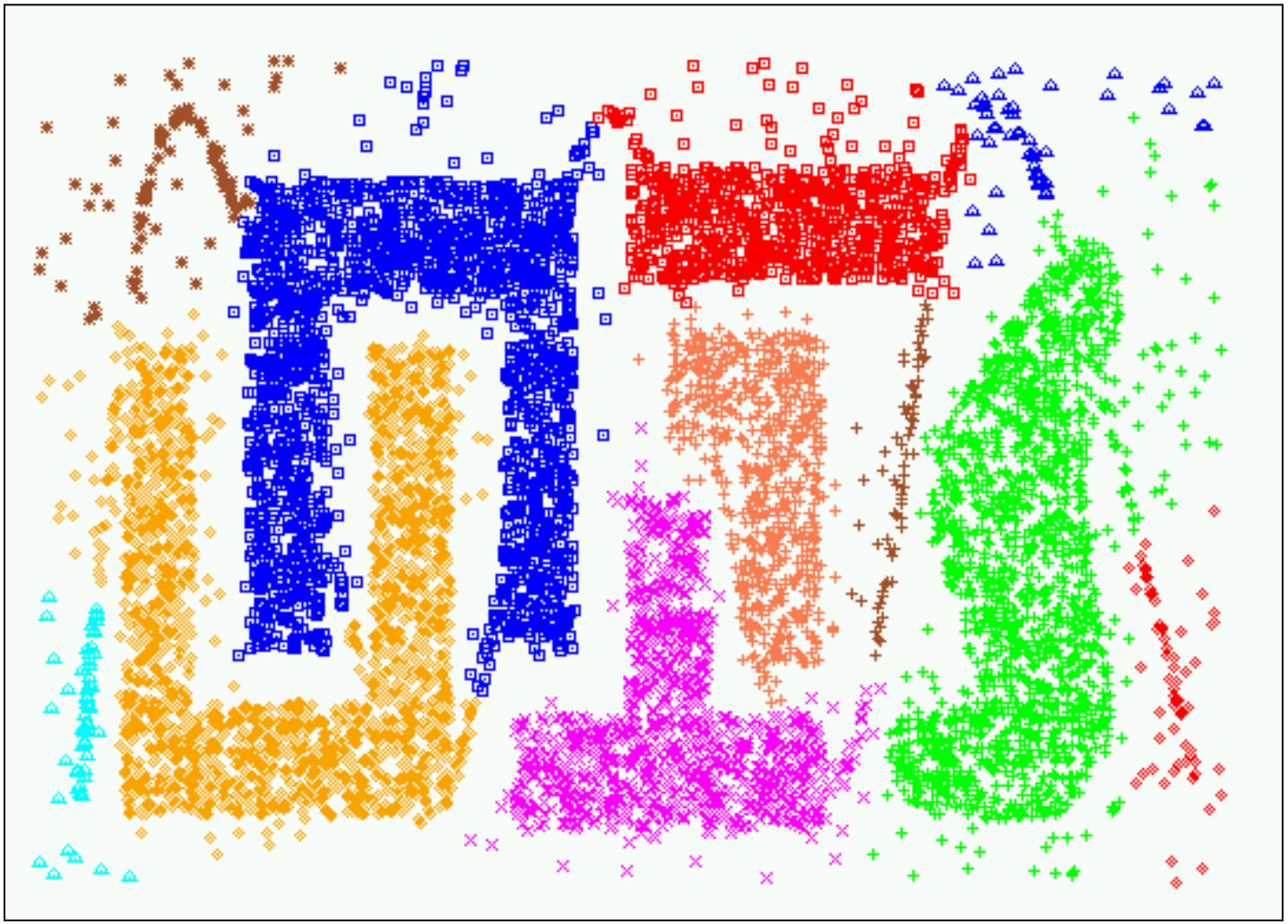
Chameleon: Lépések

- 2. fázis: Használjunk hierarchikus összevonó klaszterezést a részklaszterek egyesítésére.
  - Két klasztert akkor egyesítünk, ha az eredményül kapott klaszter rendelkezik az *egyesítendő klaszterek bizonyos tulajdonságaival.*
  - A klaszter hasonlóság modellezésére két alaptulajdonságot használunk:
    - Relatív összekapcsolhatóság: Két klaszter abszolút összekapcsolhatósága lenormálva a klaszterek belső összefüggőségével
    - Relatív közelség: Két klaszter abszolút közelsége lenormálva a klaszterek belső közelségével

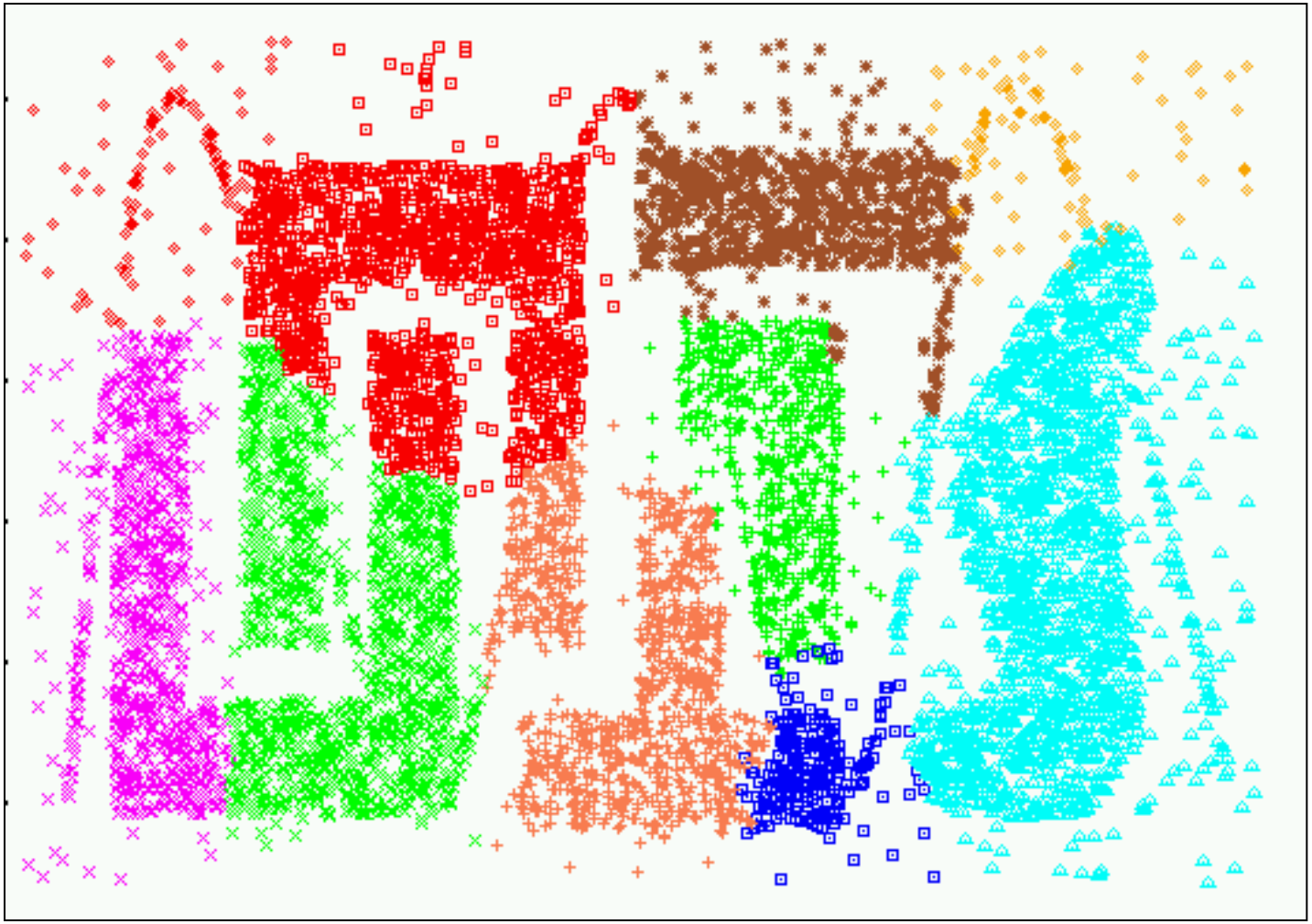
Kísérleti eredmények: CHAMELEON



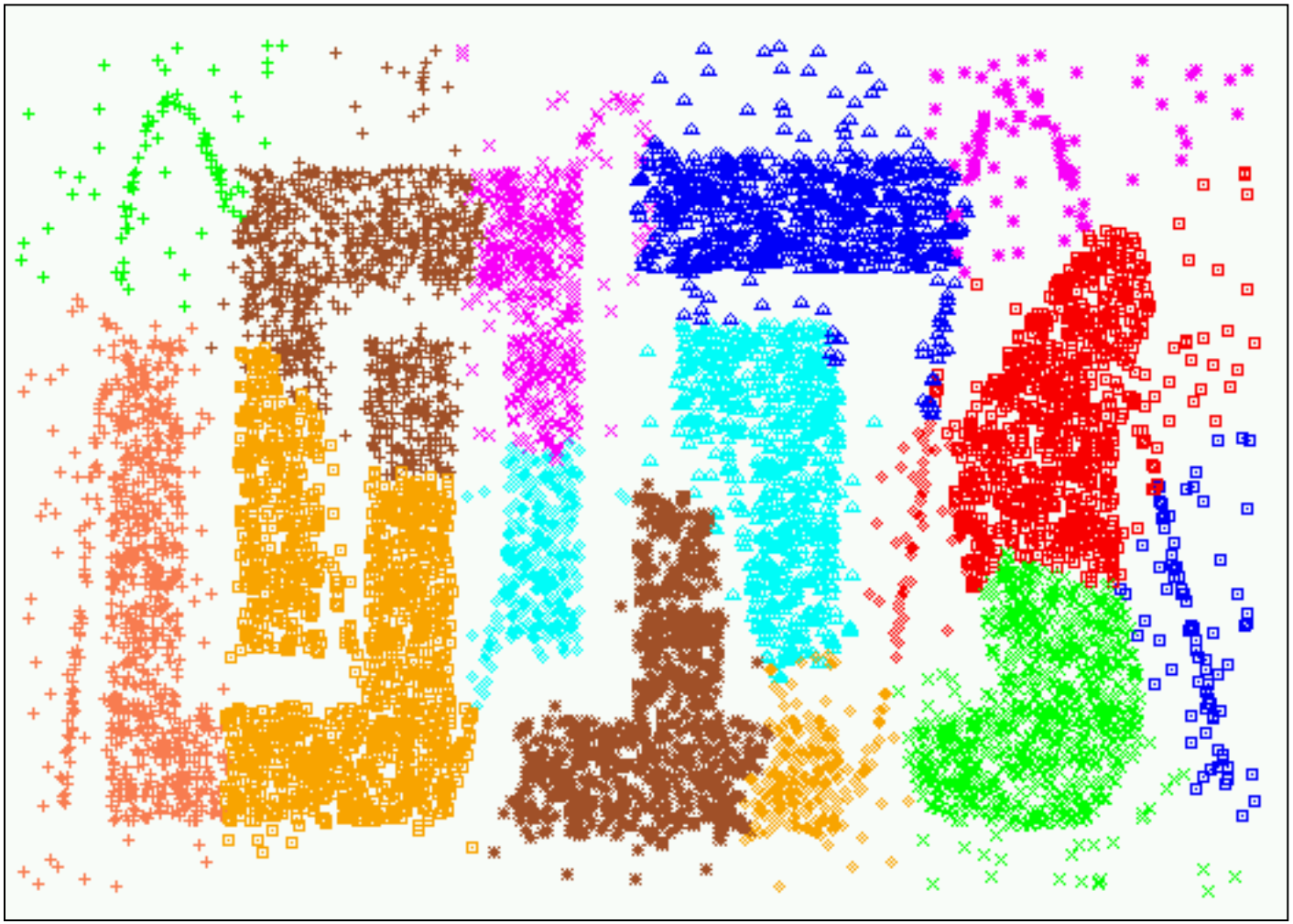
Kísérleti eredmények: CHAMELEON



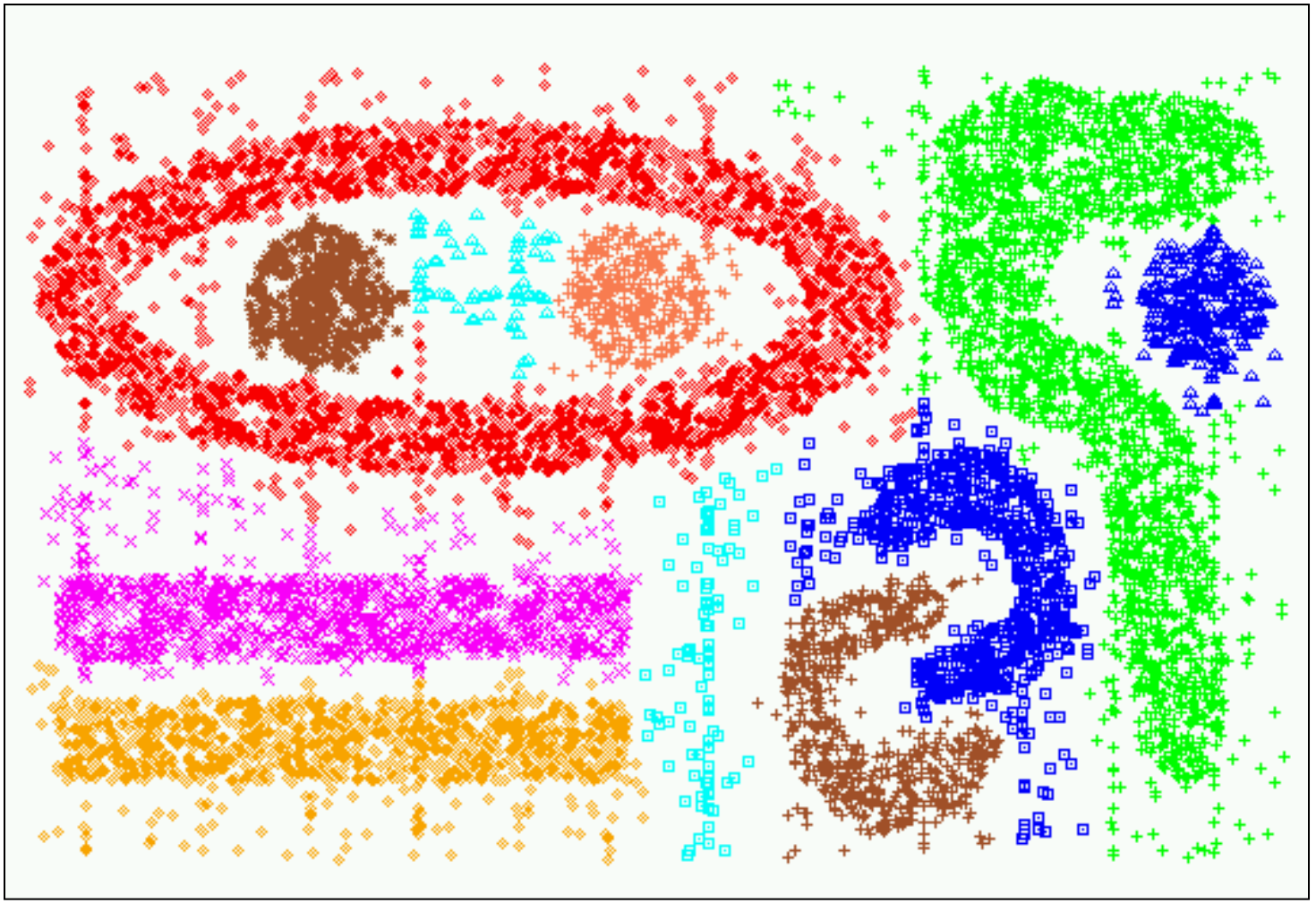
Kísérleti eredmények: CURE (*10 klaszter*)



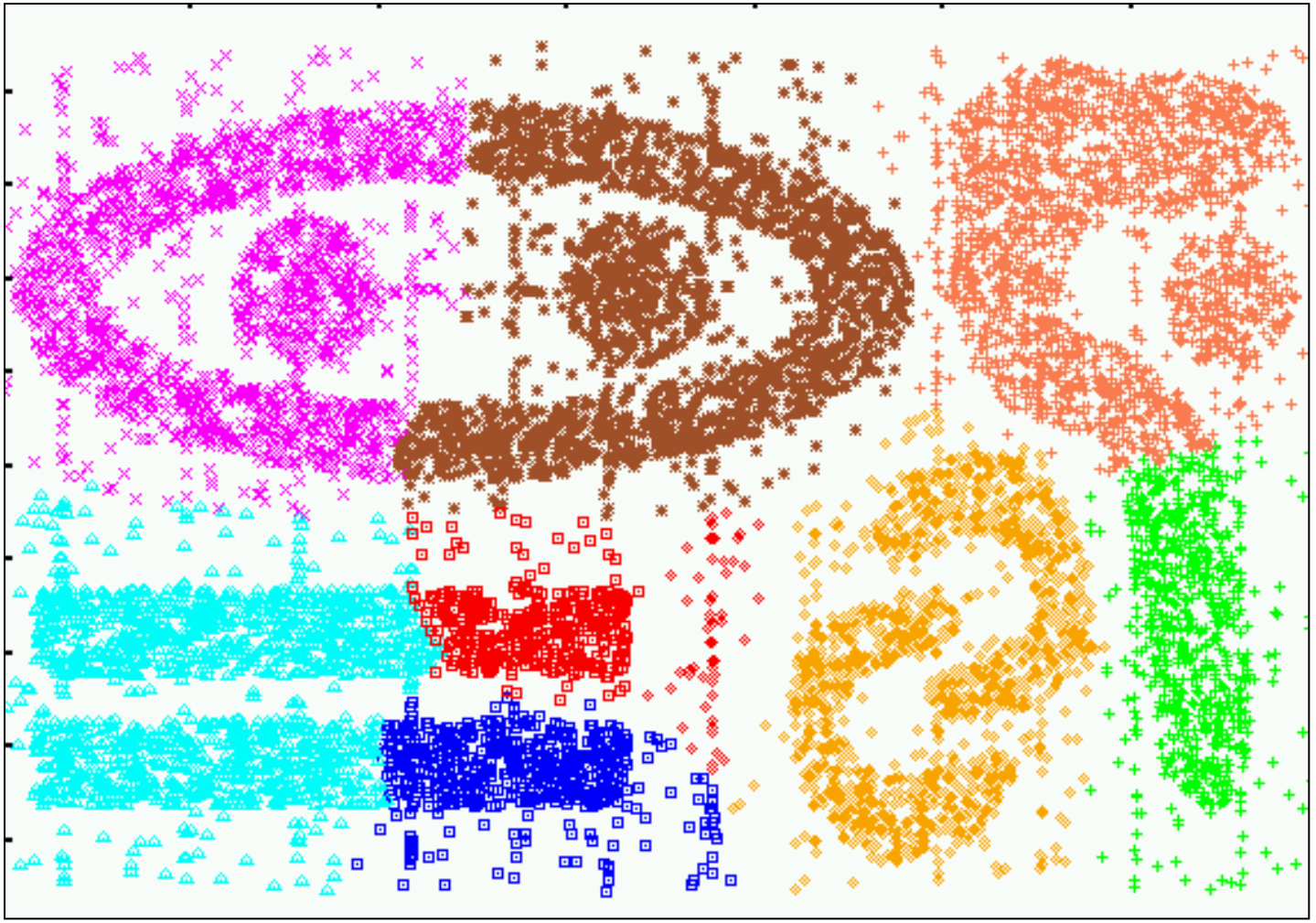
Kísérleti eredmények : CURE (*15 klaszter*)



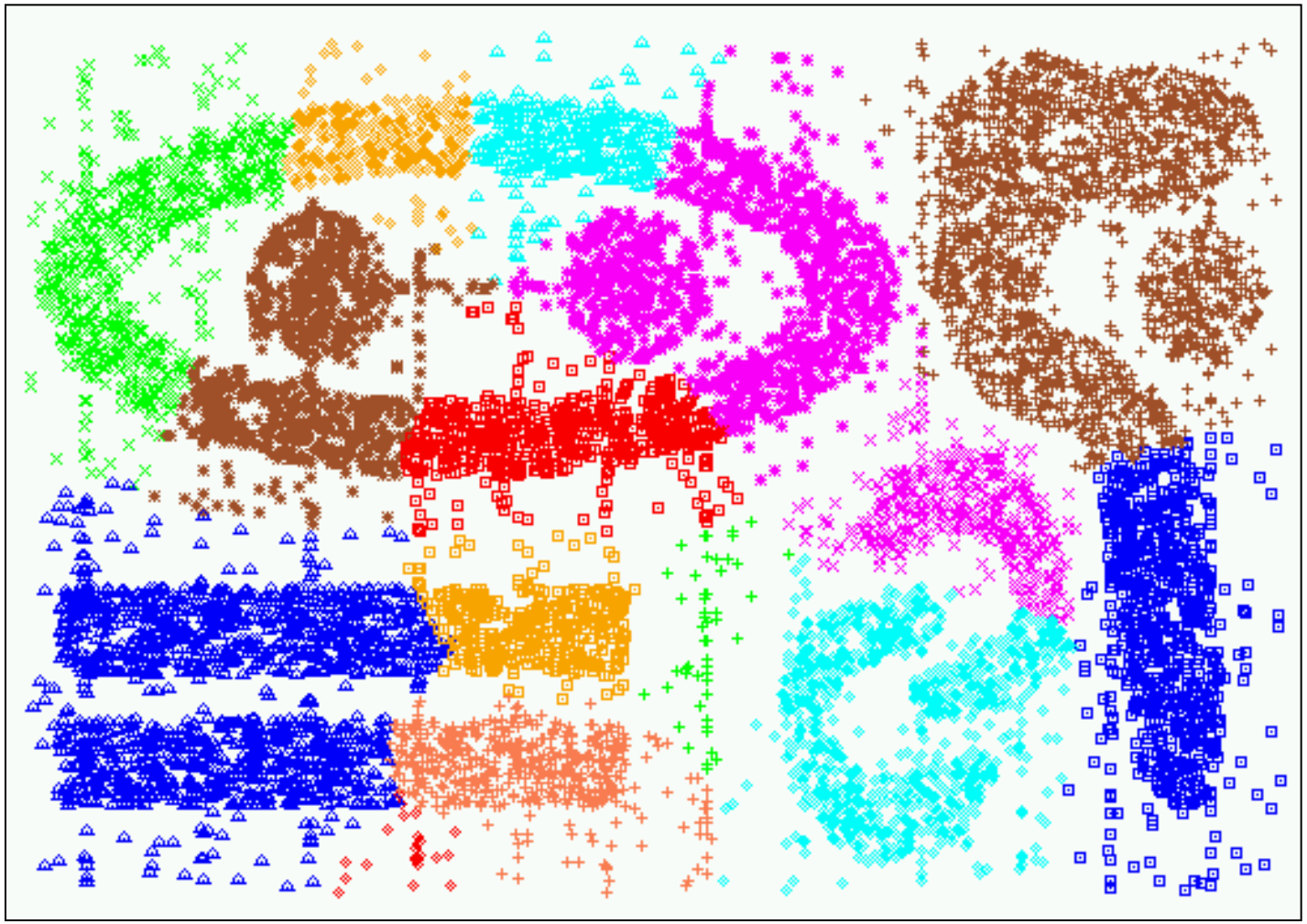
Kísérleti eredmények : CHAMELEON



Kísérleti eredmények : CURE (*9 klaszter*)

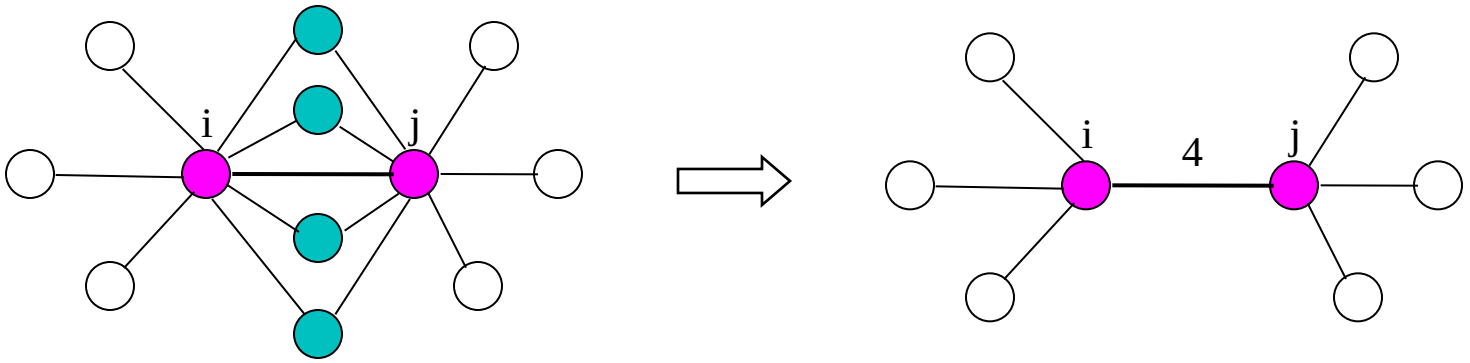


Kísérleti eredmények : CURE (*15 klaszter*)

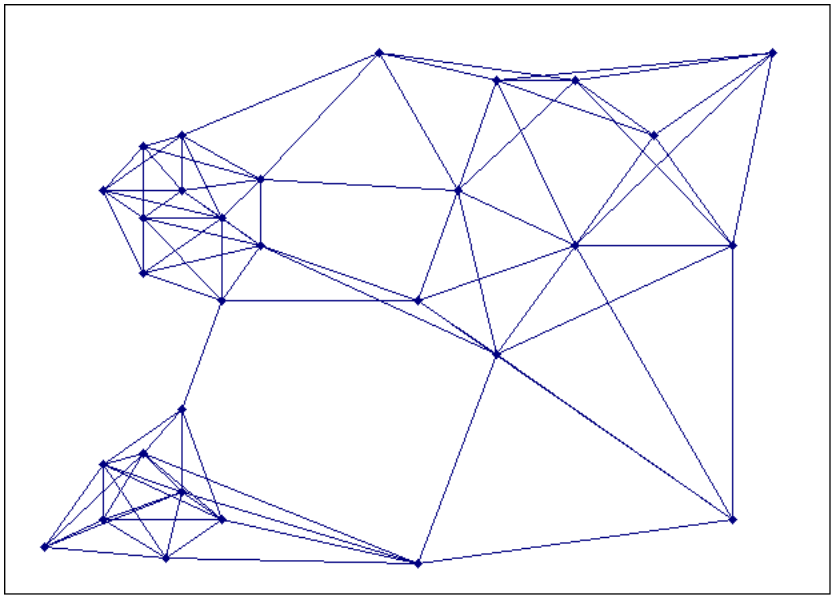


Megosztott legközelebbi társ módszer

SNN gráf: egy él súlya az élek között megosztott szomszédok száma azon feltétel mellett, hogy az élek összefüggőek.

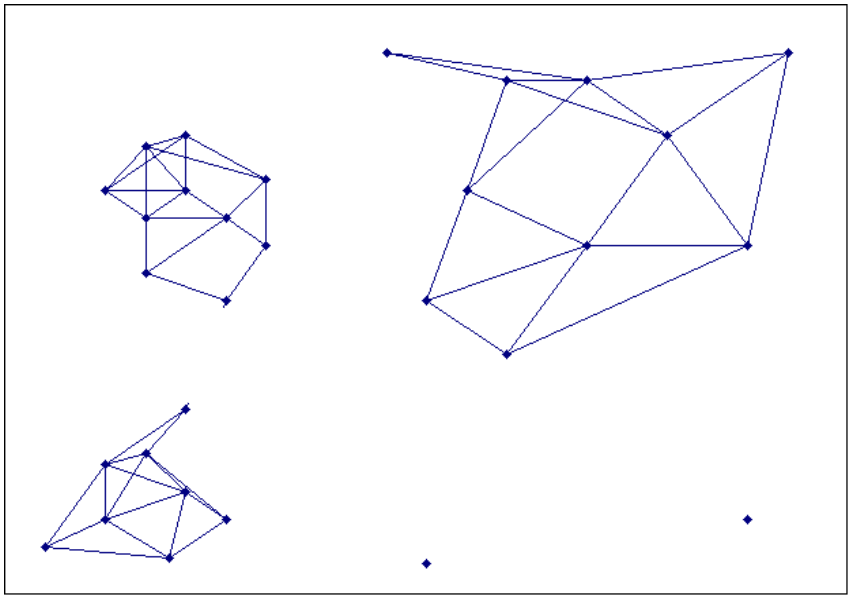


Az SNN gráf létrehozása



Ritka gráf

Az összekötő súlyok a szomszédos pontok közötti hasonlóságok



Megosztott legközelebbi társ gráf

Az összekötő súlyok a megosztott legközelebbi társak száma

ROCK (Kapcsolatokat használó robusztus klaszterezés)

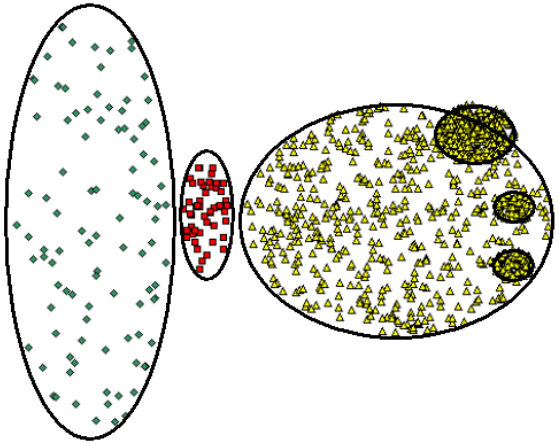
- Klaszterező algoritmus kategorikus és bináris attributumú adatok számára.
  - Egy pontpárt egymás szomszédjainak nevezünk, ha a hasonlóságuk meghalad egy küszöb értéket.
  - Alkalmazzunk egy hierarchikus klaszterező eljárást az adatok klaszterezésére.

1. Vegyünk egy mintát az adatállományból.
2. Számoljuk ki a kapcsolódási értéket minden egyes ponthalmazra, azaz transzformáljuk az eredeti hasonlóságokat (melyeket a Jaccard együttható alapján számolunk) olyan hasonlóságokká, melyek figyelembe veszik a közös szomszédokat is.
3. Végezzünk összevonó hierarchikus klaszterezést az adatokon a ,,közös szomszédok számát” használva mint hasonlósági mértéket, és maximalizáljuk a ,,közös szomszédok számát” mint célfüggvényt.
4. Rendeljük hozzá a fennmaradó pontokat a talált klaszterekhez.

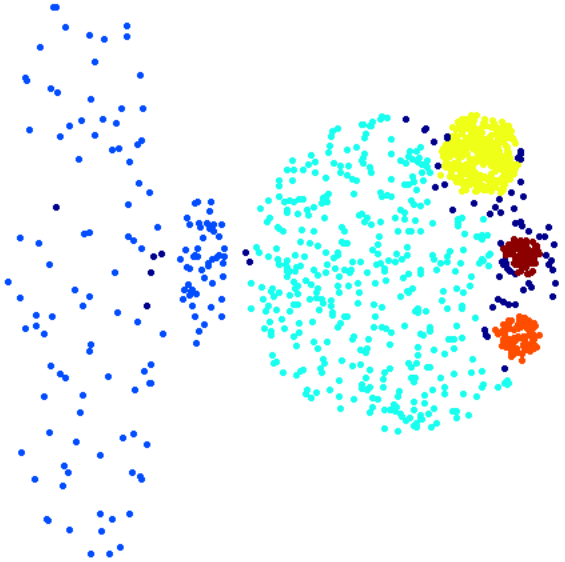
Jarvis-Patrick klaszterezés

- Először határozzuk meg az összes pont k legközelebbi társát.
  - Gráfelméleti terminológiában ez azt jelenti, hogy a k legerősebb él kivételével töröljük az összes többit a közelségi gráfból.
- Egy pontpár ugyanabba a klaszterbe kerül, ha
  - több mint T számú szomszédon osztoznak;
  - a két pont egymás k legközelebbi társ szomszédja közé tartozik.
- Például a legközelebbi társ számot 20-nak választhatjuk és a pontokat ugyanabba a klaszterbe rakjuk ha több mint 10 közös szomszédjuk van.
- A Jarvis-Patrick klaszterezés túl törékeny.

Amikor a Jarvis-Patrick jól működik



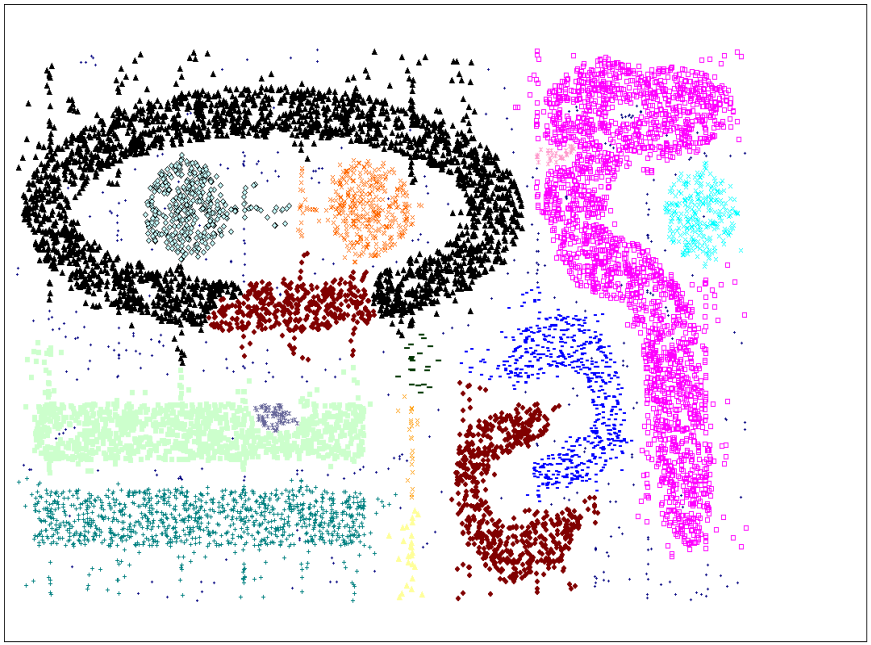
Eredeti pontok

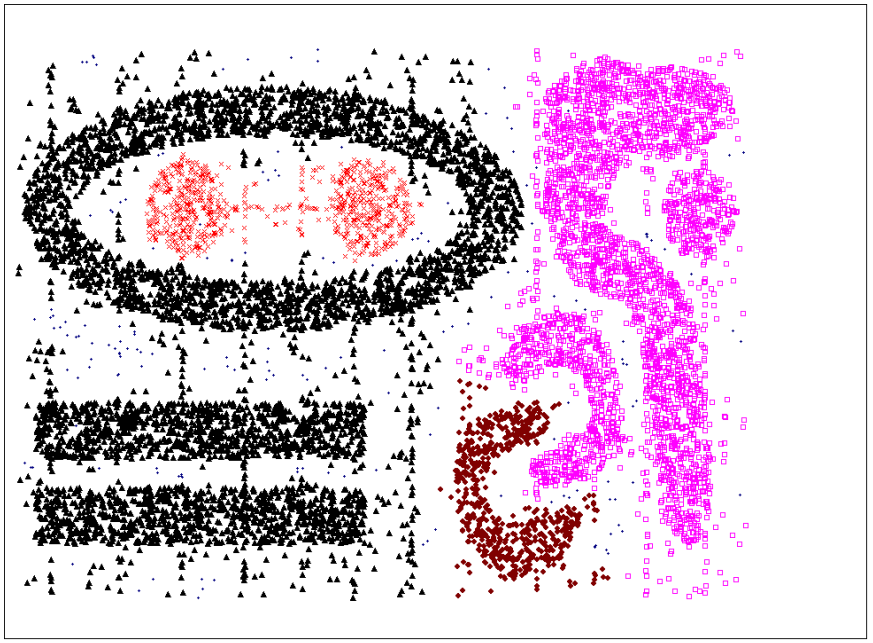


Jarvis Patrick klaszterezés

6 közös szomszéd a 20-ból

Amikor a Jarvis-Patrick nem jól működik





Az a legkisebb T küszöb, amely még nem vonja össze a klasztereket.

T – 1 küszöb

SNN klaszterező algoritmus

1. **Számítsuk ki a hasonlósági mátrixot**
   Ez az adatpontokhoz tartozó csúcsoknak felel meg, ahol az élek súlyai a pontok közötti hasonlóságok.

2. **Ritkítsuk a hasonlósági mátrixot a k leghasonlóbb szomszéd megtartásával**

   Ez annak felel meg, hogy csak a *k* legerősebb kapcsolatot tartjuk meg a hasonlósági gráfban csúcsonként.

3. **Állítsuk elő a közös szomszédok gráfját a ritkított hasonlósági mátrixból**
   Ebben a pontban alkalmazhatjuk a hasonlósági küszöböt az összefüggő komponensek megtalálására a klaszterekhez (Jarvis-Patrick algoritmus).

4. **Határozzuk meg mindegyik pont SNN sűrűségét**
   Egy *Eps* a felhasználó által meghatározott paraméterrel keressük meg azokat a pontokat, amelyek SNN hasonlósága nagyobb mint *Eps* minden pont esetén. Ez a pont SNN sűrűsége.

SNN klaszterező algoritmus

5. **Találjuk meg a középpontokat**
   Egy felhasználó által meghatározott *MinPts* paraméter segítségével keressük meg a középpontokat, azaz azokat, amelyek SNN sűrűsége nagyobb mint *MinPts.*

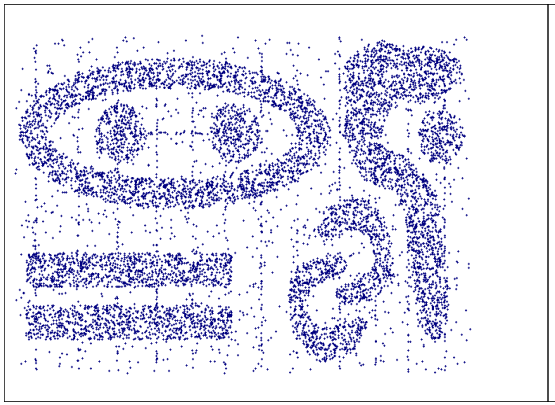
6. **Képezzünk klasztereket a középpontokból**
   Ha két középpont egy *Eps* sugáron belül van egymáshoz képest, akkor azok ugyanabba a klaszterbe kerülnek.

7. **Hagyjuk figyelmen kívül a zajos pontokat**
   Az összes nem középpontot, amely nincs egy középpont *Eps* sugarán belül figyelmen kívül hagyjuk.

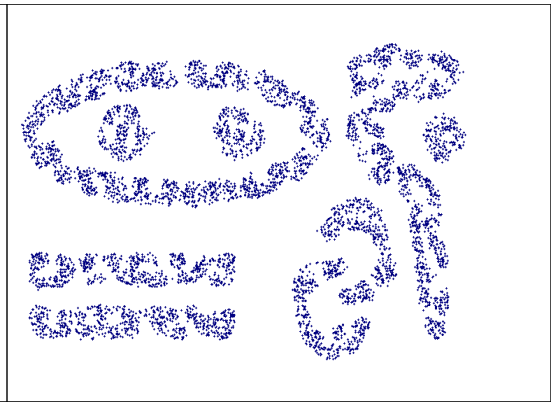
8. **Rendeljük hozzá az összes nem-zajos és nem középpont pontot a klaszterekhez**
   Ez megtehető az ilyen pontok legközelebbi középponthoz, így annak klaszteréhez rendeléssel.

(Note that steps 4-8 are DBSCAN)

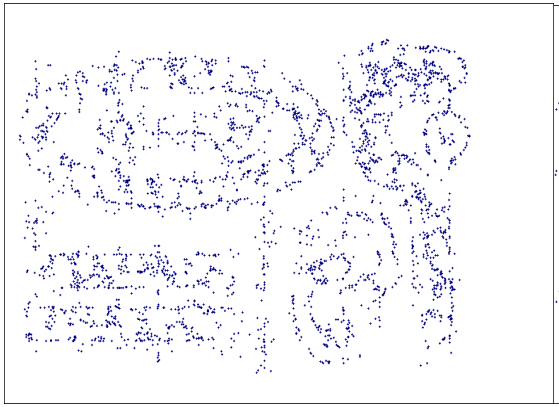
SNN sűrűség



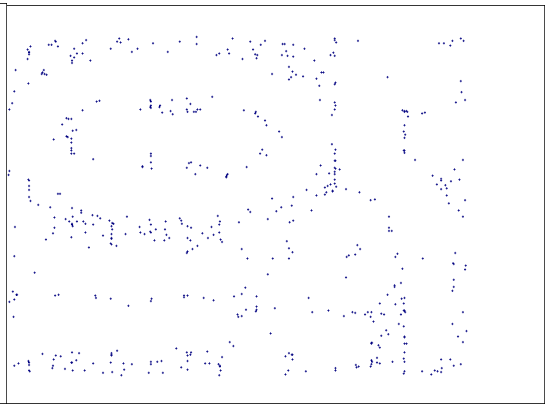
a) Minden pont



b) Nagy SNN sűrűség

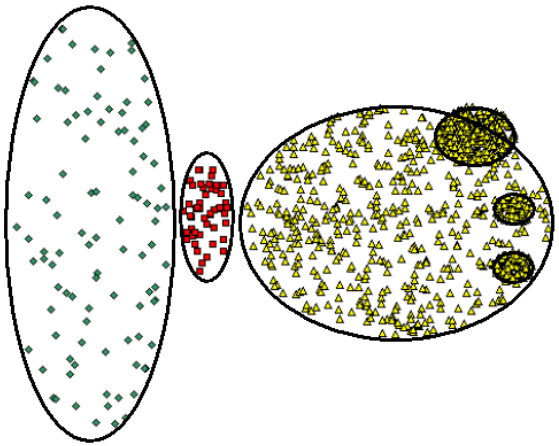


c) Közepes SNN sűrűség

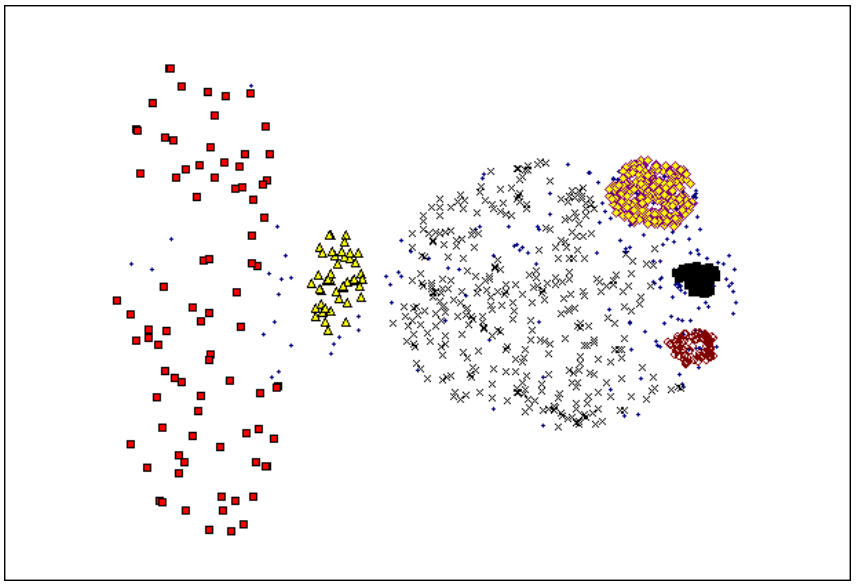


d) Kis SNN sűrűség

Az SNN klaszterezés képes kezelni az eltérő sűrűségeket

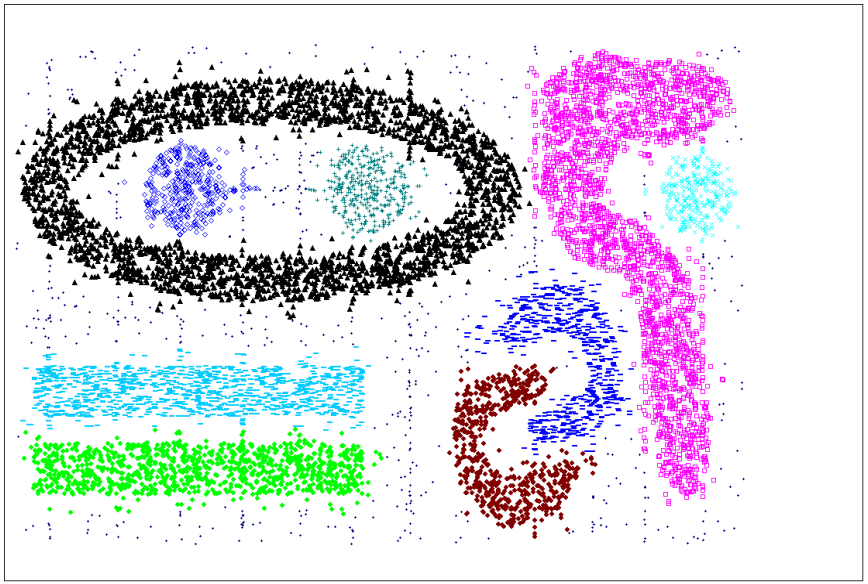


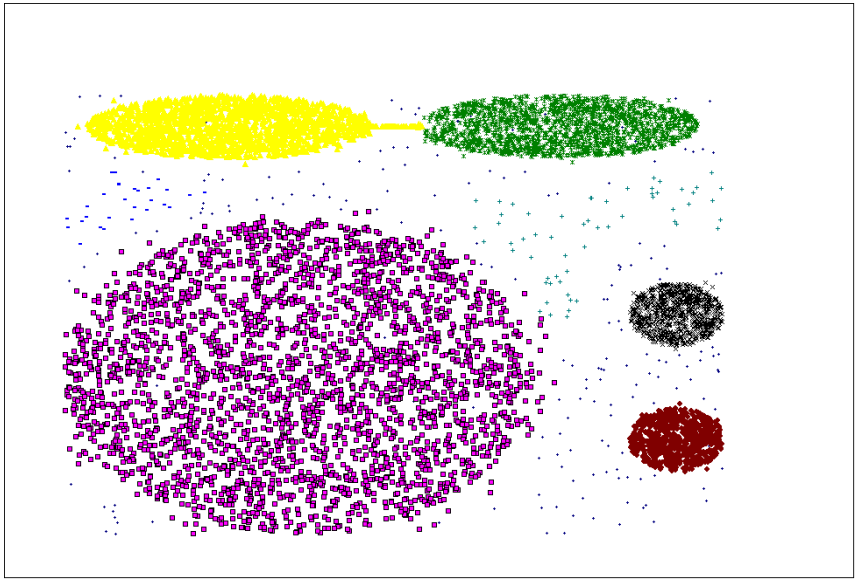
Eredeti pontok



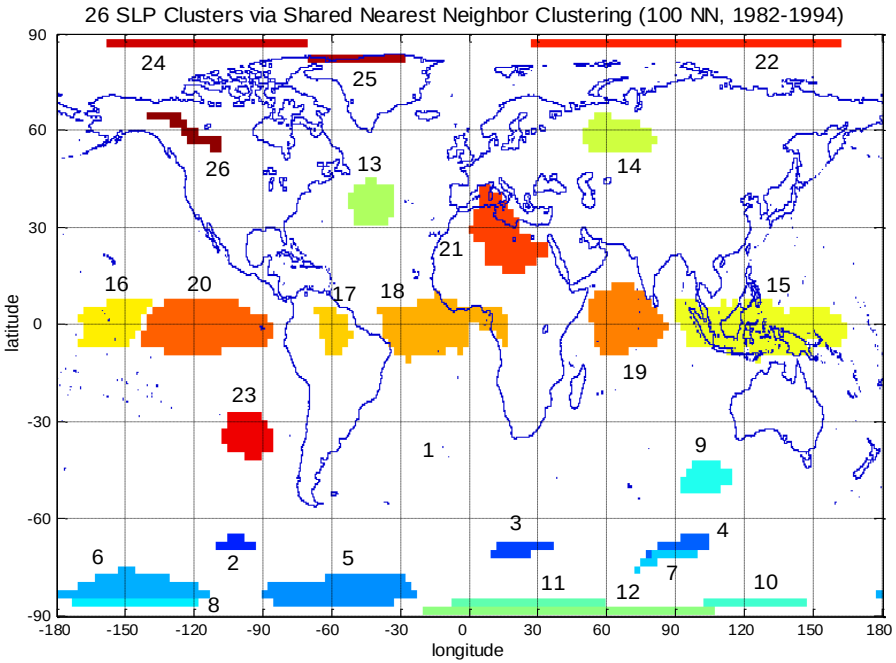
SNN klaszterezés

Az SNN klaszterezés képes kezelni a nehéz helyzeteket

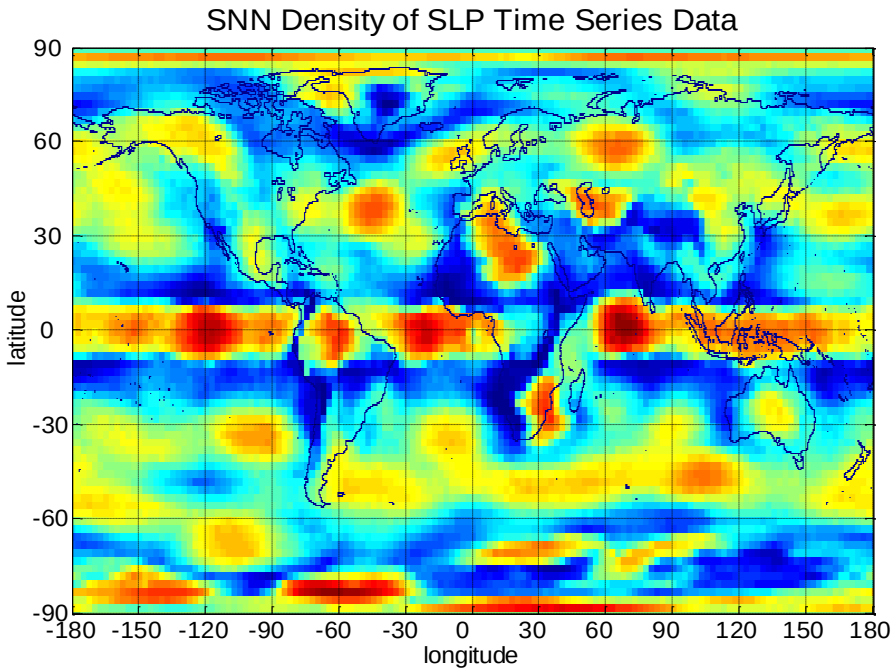




Idősorok klasztereinek keresése tér-időbeli adatokban



SNN klaszterek a tenger szinti légnyomásban (SLP).



SNN sűrűségek a Föld pontjain

Az SNN klaszterezés jellemzői és korlátai

- Nem klaszterezi az összes pontot

- Az SNN klaszterezés komplexitása nagy
  - O(n * *Eps* belüli szomszédok megkeresési ideje)
  - A legrosszabb esetben $O(n^2)$
  - Alacsony dimenzióban vannak hatékonyabb módszerek a legközelebbi társak megtalálására
    - R* fa
    - k-d fa