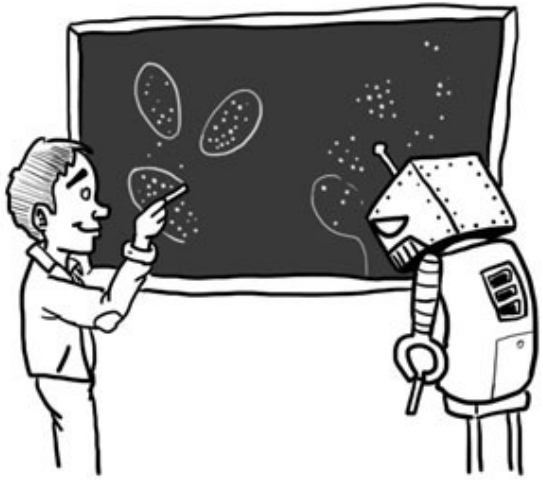
Data-Science 1

clusteranalyse



Karel de Grote Hogeschool

9/05/23

Inhoud

- n-dimensionale ruimten..
- afstanden tussen punten
- clusters zoeken
  - kmeans
  - hiërarchisch
- combinatie met beslissingsbomen

n-dimensionale ruimten

Voorbeeld

- gegeven: tabel met punten van studenten

vak1	vak2	vak3	vak4	vak5	vak6	vak7	vak8	vak9	vak10
10	14	11	16	15	13	9	18	14	13
16	15	18	19	16	16	15	14	17	18
...	...	...	...	...	...	...	...	...	...
8	6	9	10	14	5	0	5	1	0

- gevraagd: kunnen we de studenten indelen in bepaalde "types" (clusters)
- dit is een voorbeeld van “unsupervised learning”
- opmerking: welk meetniveau hebben de kolommen?

n-dimensionale ruimten

- wat als er maar 2 kolommen waren?
  - eerste kolom hernoem je "x"
  - tweede kolom hernoem je "y"
  - beide kolommen hebben minstens interval meetniveau
  - wat is iedere rij dan?
- wat als er 3 kolommen zijn?
- wat als er n kolommen zijn?

Clusters

- rij = punt in n-dimensionale ruimte
- een cluster is een aantal rijen van een tabel die "bij elkaar horen" of "gelijkaardig zijn"
- 2 rijen zijn gelijkaardig als de punten dicht bij elkaar liggen
  - de "afstand" moet klein zijn

Afstanden

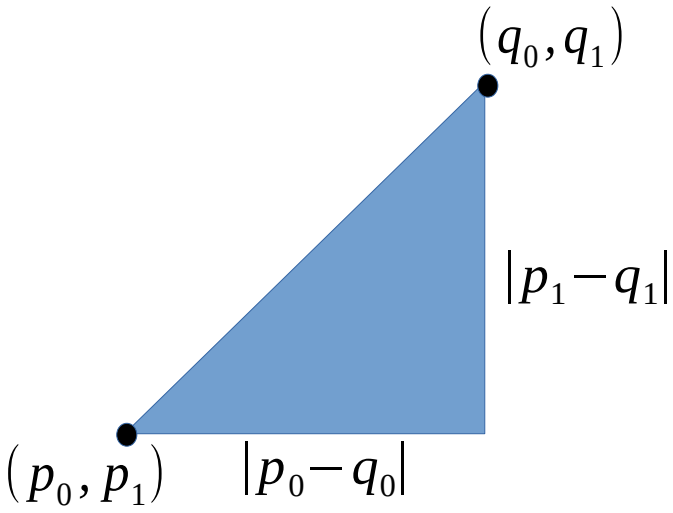
2

Hoe meet je de afstand?

- gebruik een "metriek":
  - Euclidisch
  - Manhattan (taxi)
  - Chebychev
  - Minkowski
  - Mahalanobis
  - ...
- notatie: ieder punt heeft coördinaten

$$p=(p_0, p_1, p_2, \cdots, p_{n-1})$$

Euclidische afstand



- stelling van pythagoras

- 2D: $d(p,q) = \sqrt{(p_0 - q_0)^2 + (p_1 - q_1)^2}$
- 3D: $d(p,q) = \sqrt{(p_0 - q_0)^2 + (p_1 - q_1)^2 + (p_2 - q_2)^2}$
- n-D: $d(p,q) = \sqrt{\sum_{i=0}^{n-1} (p_i - q_i)^2}$

Euclidische afstand

- voorbeeld

10	12	15	13	9
18	14	13	15	17

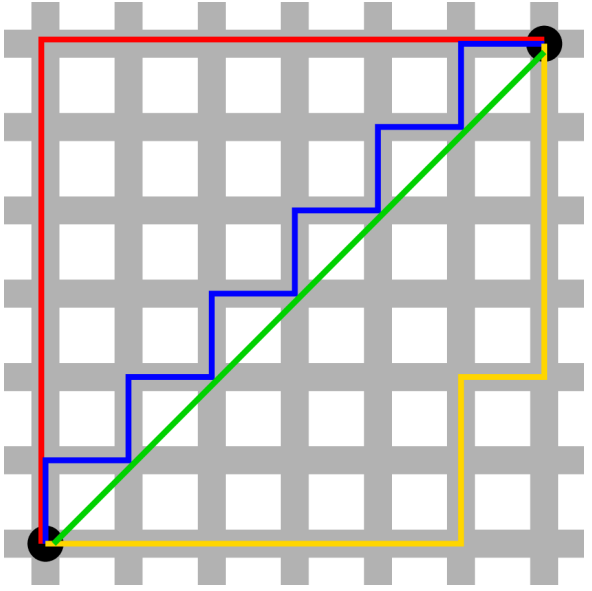
- wat is de afstand?

$$d(p,q)=\sqrt{(10-18)^2+(12-14)^2+(15-13)^2+(13-15)^2+(9-17)^2}$$
$$=\sqrt{8^2+2^2+2^2+2^2+8^2}=\sqrt{140}=11{,}832$$

- in Python: zie code

Manhattan (taxi) afstand

- je afstanden ook anders meten
- voorbeeld: kortste afstand in Manhattan
- $d(p,q)=\sum_{i=1}^{n}|p_i-q_i|$



Manhattan (taxi) afstand

- voorbeeld

10	12	15	13	9
18	14	13	15	17

- wat is de afstand?

$$d(p,q) = |10-18| + |12-14| + |15-13| + |13-15| + |9-17|$$
$$= 8+2+2+2+8 = 22$$

- in Python: zie code

Gestandardiseerde afstand

- probleem: waarden in kolommen hebben soms compleet andere grootorde
- voorbeeld: kolommen "leeftijd" en "km per jaar gereden"
  - "km per jaar" zal een grotere invloed hebben
- oplossing: schalen
  - zet iedere kolom eerst om naar Z-scores
- zie Python

Meetniveau’s

- normaal: minstens interval meetniveau nodig
- wat als dit niet is?
  - ordinaal
    - gebruik de volgnummers (niet helemaal correct, maar kan helpen)

      ```
      tabel.kolom.cat.codes
      ```

    - als er klassen zijn: vervang ze door de klassenmiddens

      ```
      tabel.kolom = tabel.kolom.apply(lambda i :
      (i.left+i.right)/2)
      ```

Meetniveau’s

- nominaal
  - verwijderen...
  - binaire variabele: gebruik 1 en 2 of -1 en 1
  - woorden: word2vec
  - adressen: longitude/latitude
  - producten: plaats in winkel (rij, rek, hoogte)
  - kleur: R,G,B
  - pd.to_dummies() -> weinig zin hier
  - ...

Clusters zoeken: k-means

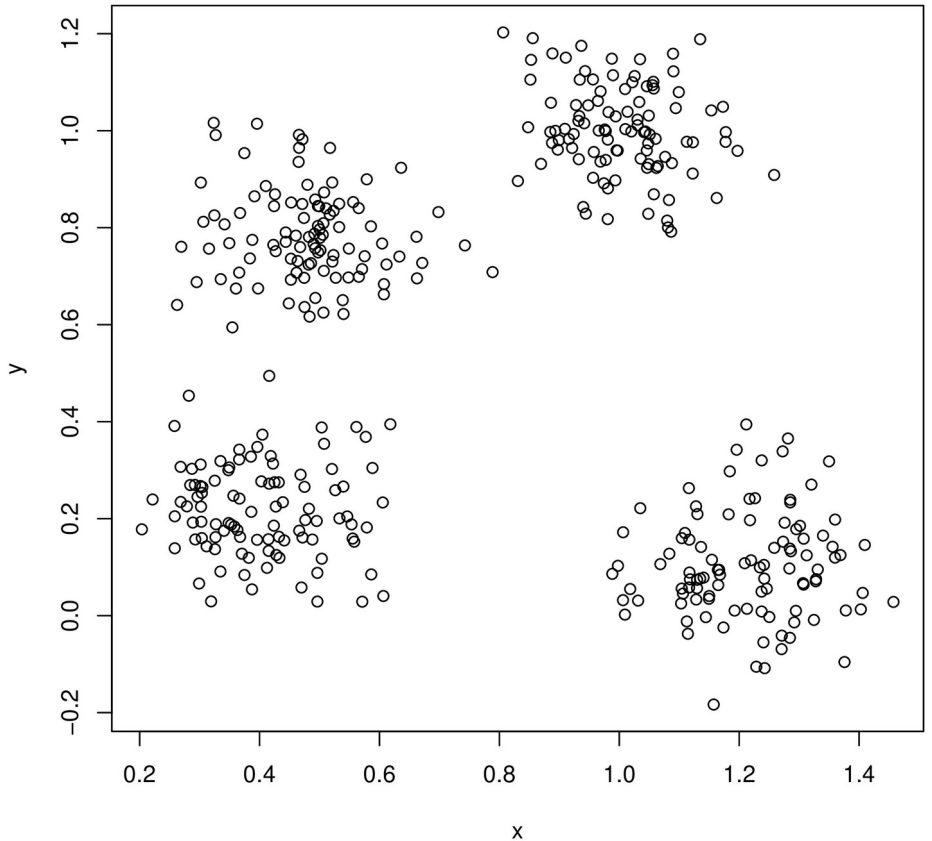
K-means

- we zoeken n clusters (n is gegeven)
- is gebaseerd op Kohonen netwerk
  - neuraal netwerk (1982)
  - geïnspireerd door de werking van onze hersenen
- meestal (gestandardiseerde) euclidische afstand

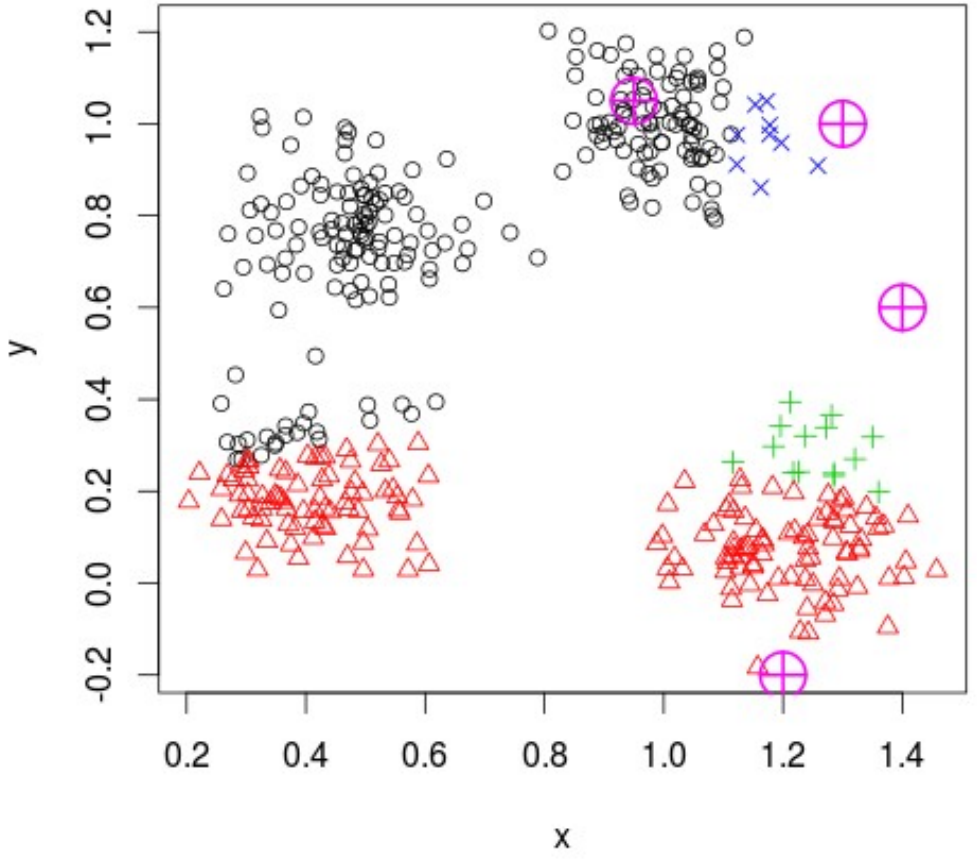
Algoritme

- selecteer n willekeurige punten ("centroids")
- herhaal
  - associeer ieder punt van de dataset met de centroid die het dichtste bij ligt (zo maak je n clusters)
  - bereken per cluster het "midden" en vervang de centroid door deze nieuwe waarde
  - totdat de centroids niet meer veranderen

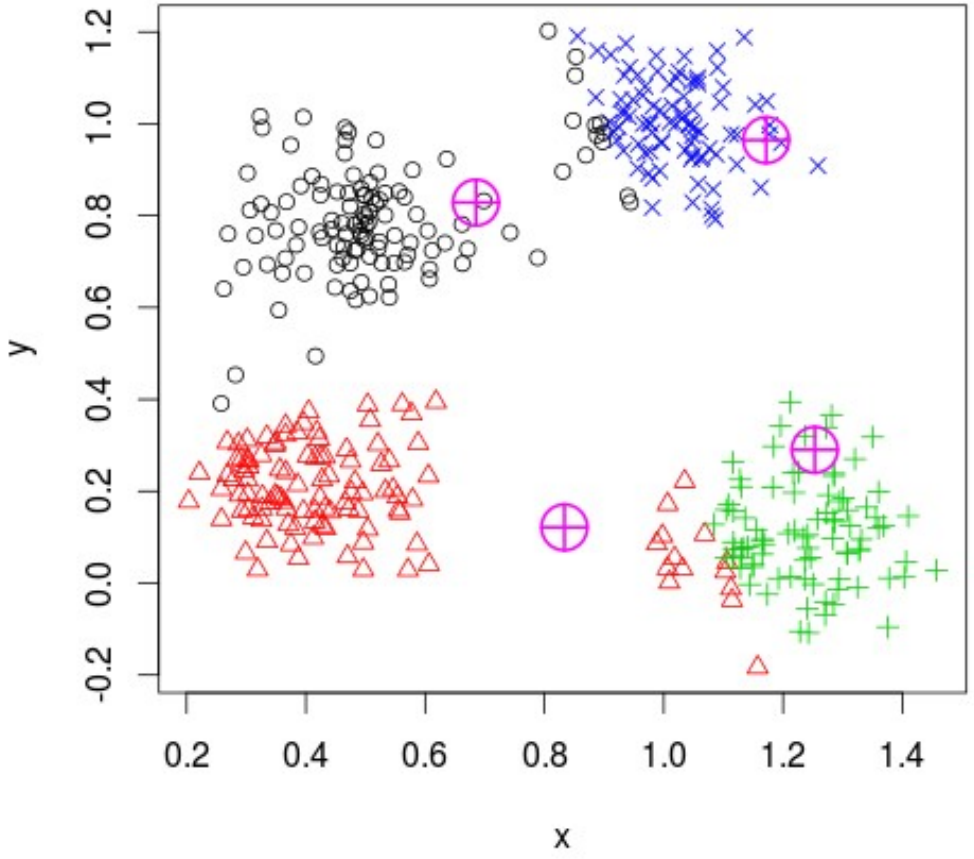
Voorbeeld k-means



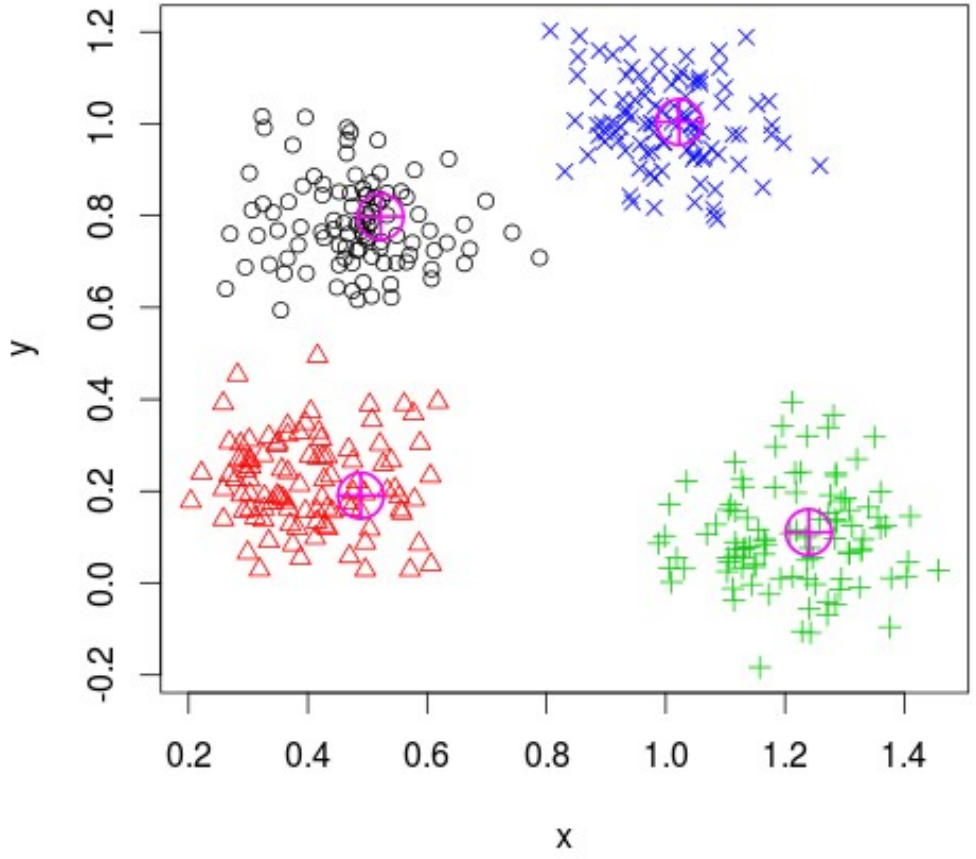
Voorbeeld k-means



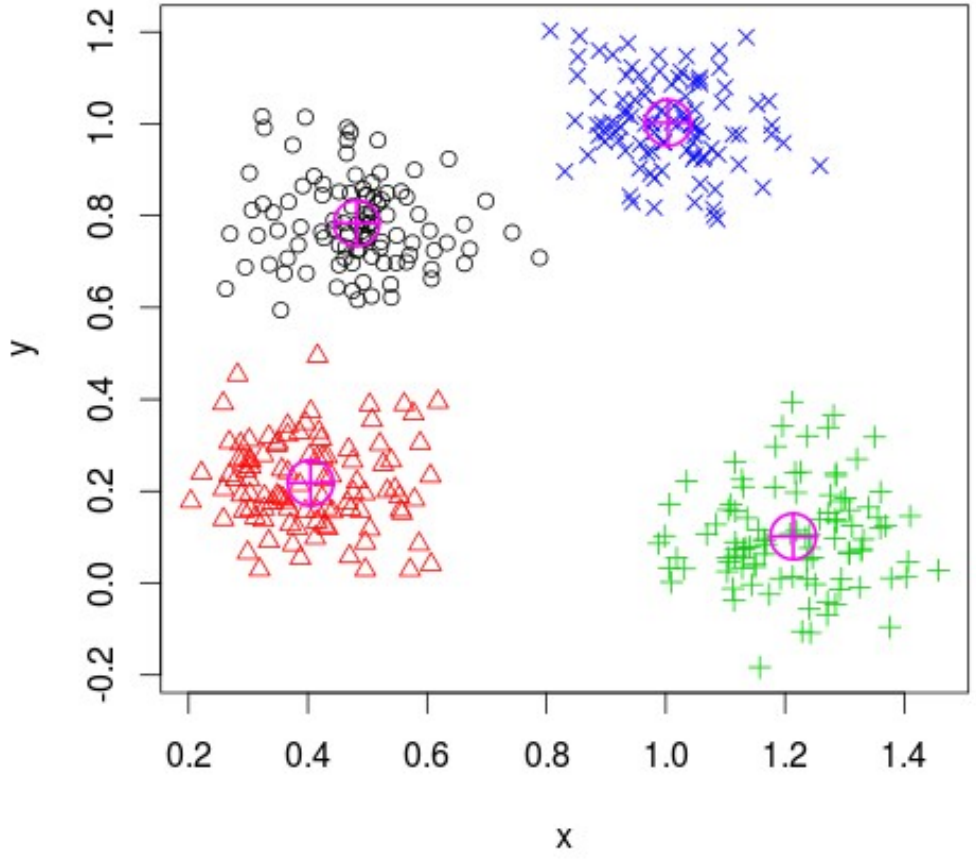
Voorbeeld k-means



Voorbeeld k-means



Voorbeeld k-means



In Python

- zie code

K-means

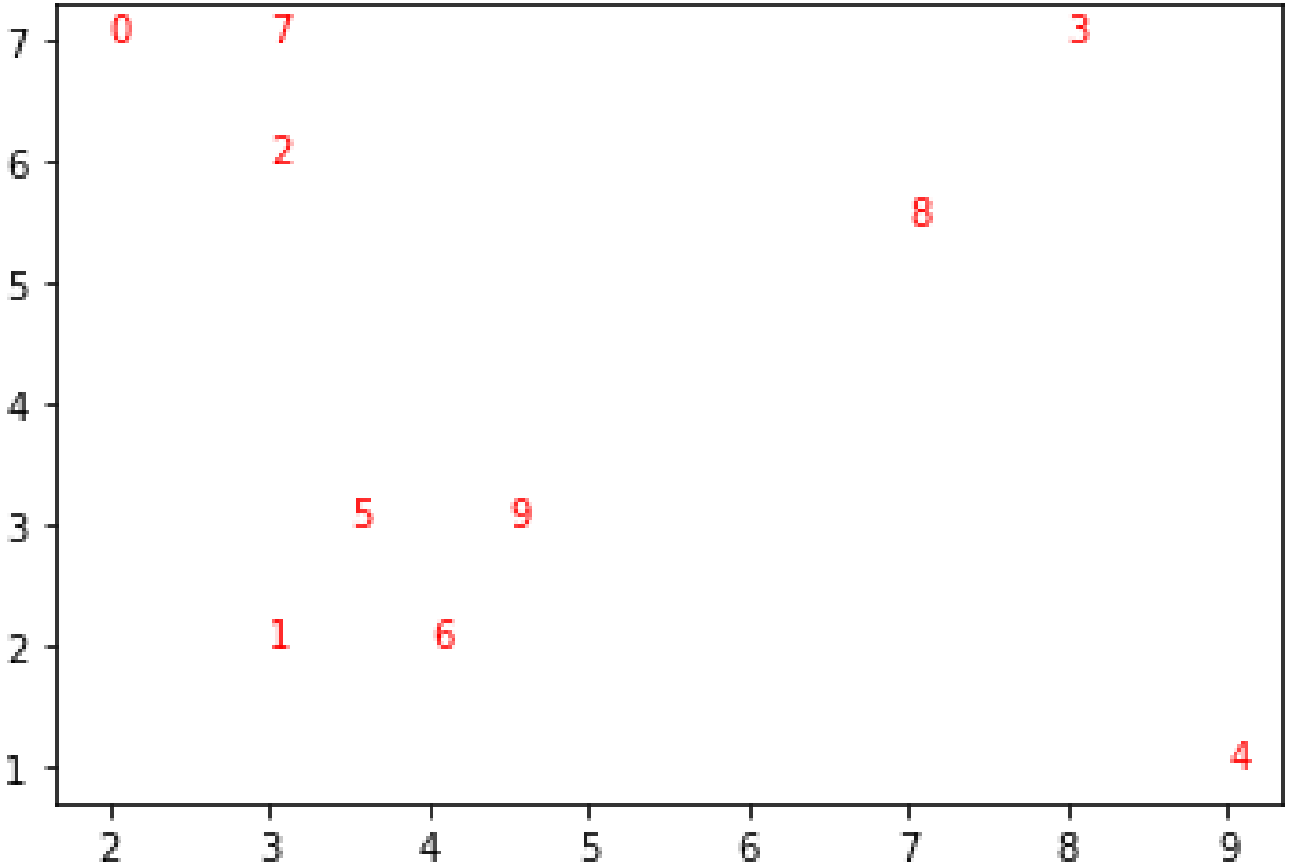
- je moet op voorhand weten hoeveel clusters je zoekt
- plaats van initiële centroids kan belangrijk zijn (Python doet dit op een ‘intelligente’ manier)

Clusters zoeken: hiërarchisch

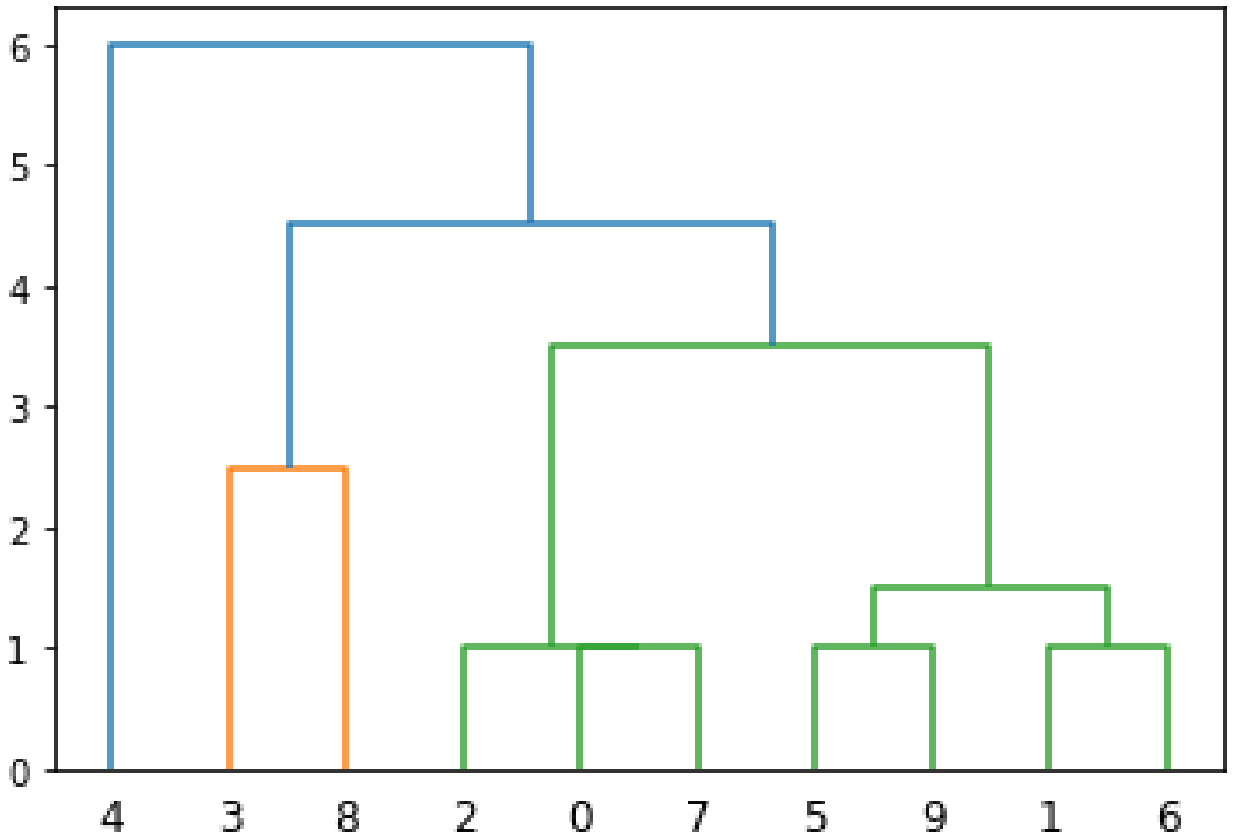
Hiërarchische clustering

- als je op voorhand niet weet hoeveel clusters je zoekt
- deterministisch: levert steeds hetzelfde resultaat
- maakt een boomstructuur: "dendrogram"

Voorbeeld



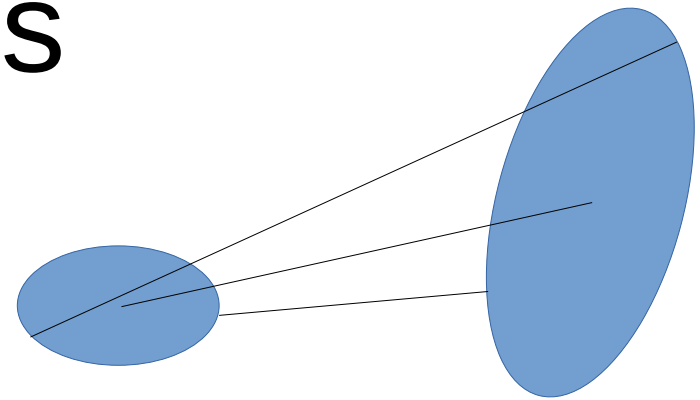
Dendrogram



Algoritme

- begin "onderaan": ieder punt is een cluster met 1 element erin
- herhaal
  - zoek de 2 clusters die het dichtst bij elkaar liggen
  - voeg deze clusters bij elkaar in een hoger niveau
  - tot er maar 1 cluster overblijft

Afstand tussen clusters



- je moet nu de afstand tussen 2 clusters kunnen berekenen
  - afstand tussen middelpunten (centroid linkage)
  - min afstand tussen de punten (single linkage)
  - max afstand tussen de punten (complete linkage)
  - gemiddelde afstand tussen de punten (average linkage)
  - mediaan afstand tussen de punten (median linkage)
  - ...

In Python

zie code

4

Combinatie met beslissingsbomen

Clustering en beslissingsbomen

- clustering geeft een "label" aan iedere rij
- voeg dit label als kolom toe
- een beslissingboom kan nu bepalen waarom een rij in een cluster hoort!

Voorbeeld

zie code

Oefeningen

Oefeningen

- Simpsons revisited
- Studenten
- Extraterrestrial life