Föreläsning 8 - Klusteranalys

Josef Wilzen

2022-09-20

Outline

k-medoid klustering

k-medoids/Partitioning Around Medoids (**PAM**): använder medioder som center/prototyp

- mediod: är ett representativ observation inom ett dataset/kluster, som har minimalt avstånd med övriga observationer i datasetet/klustret.
- mediod $\neq$ centroid, median, geometrisk median
- Medioder är lätta att tolka
  - centroider kan vara punkter som inte liknar någon av observationerna i data
- k-medoids
  - minimimerar summan av parvisa avstånd
  - kan använda godtyckliga avståndsmått
  - mer robust brus och extremvärden
- k-means: använder oftast euklidiskt avstånd

k-medoids

Algorithm 14.2 *K-medoids Clustering.*

1. For a given cluster assignment C find the observation in the cluster minimizing total distance to other points in that cluster:

$$i_k^* = \underset{\{i:C(i)=k\}}{\operatorname{argmin}} \sum_{C(i')=k} D(x_i, x_{i'}). \tag{14.35}$$

   Then $m_k = x_{i_k^*}$, $k = 1, 2, \ldots, K$ are the current estimates of the cluster centers.

2. Given a current set of cluster centers $\{m_1, \ldots, m_K\}$, minimize the total error by assigning each observation to the closest (current) cluster center:

$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} D(x_i, m_k). \tag{14.36}$$

3. Iterate steps 1 and 2 until the assignments do not change.

Källa: The Elements of Statistical Learning

Densitetsbaserade metoder

Densitetsbaserade metoder

Kluster kan formas baserat på hur densiteten på punkterna varierar över variablerna: täta områden kan defineras som ett kluster

DBSCAN

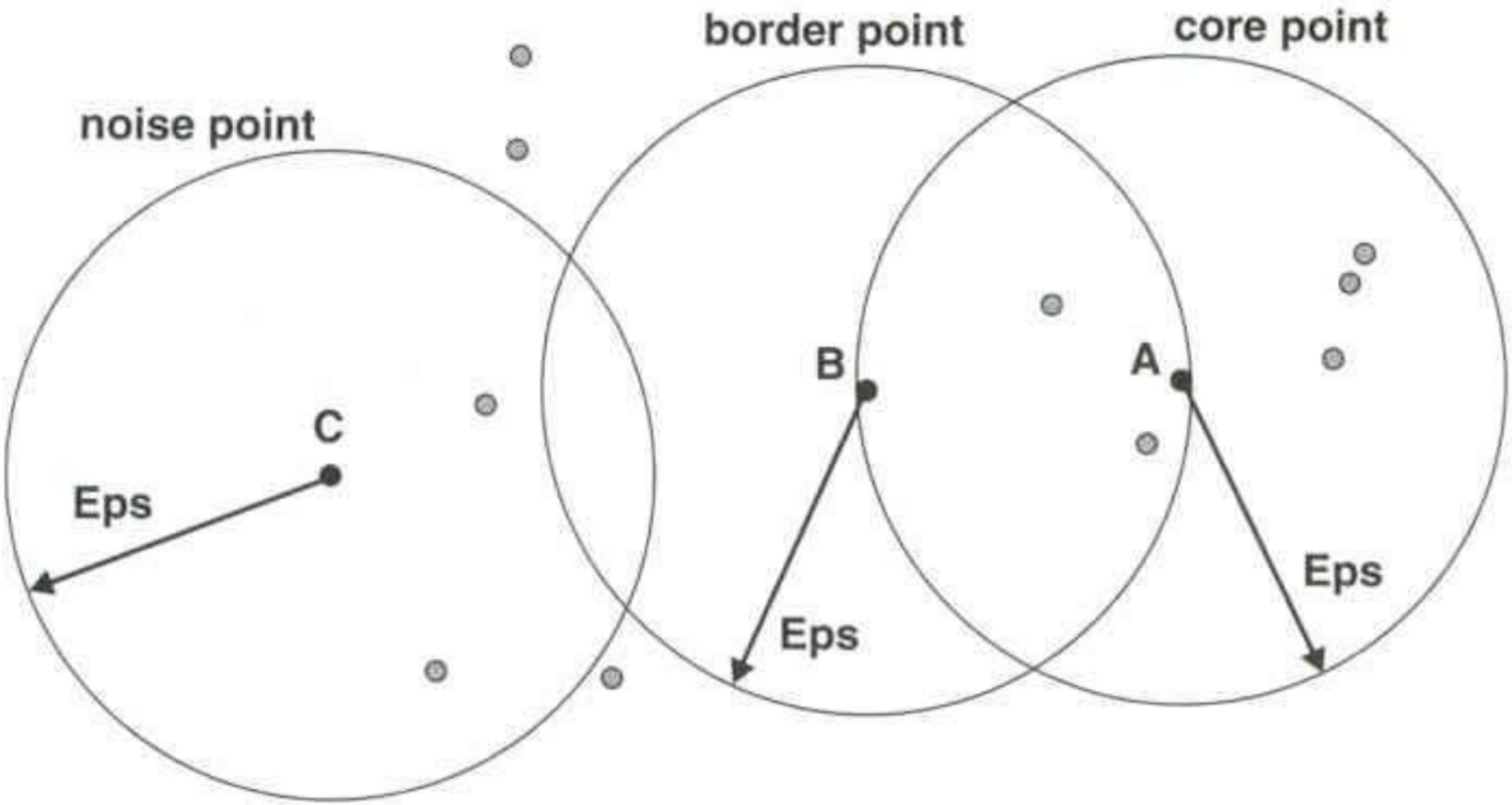
- Skapa kluster baserat på punkternas täthet
- Definitioner
  - *eps*, motsvarar en sökradie
  - *minPts*, anger minsta gräns för antalet punkter

Klassning av observationer

Tre olika klassningar av observationer

- Kärnpunkt: Antalet punkter inom sökradien *eps* överstiger *minPts*
- Gränspunkt: Inte en kärnpunkt, men hamnar inom *eps*-radien av en kärnpunkt
- Bruspunkt: Varken kärnpunkt eller gränspunkt

Illustration



Algoritmen

Algorithm 8.4 DBSCAN algorithm.

1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

Val av eps och minPts

1. Definiera ett nummer k
2. Beräkna avståndet mellan varje punkt och dess k-närmaste granne och sortera punkterna enligt ökande avstånd
3. Definiera *eps* som värdet där skarp förändring märks (armbågsmetoden)
4. $minPts = k$

k-värdet valt på steg 1 påverkar inte *eps*-värdet mycket om k inte är för stort eller för litet

Exempel

Figure 8.22. Sample data.

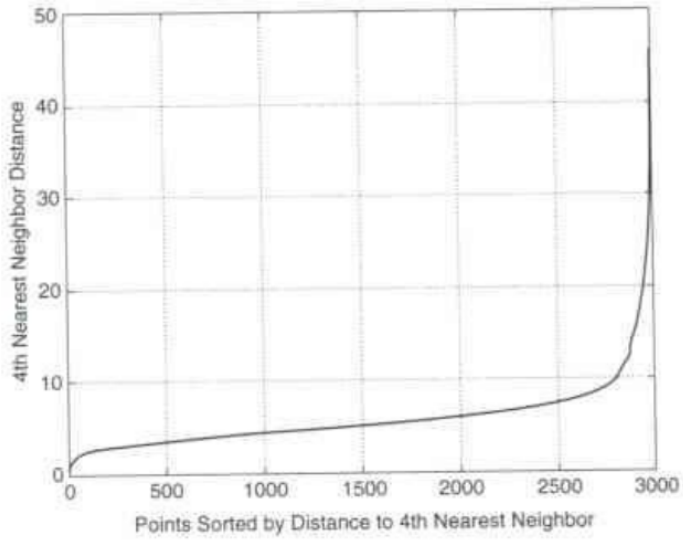
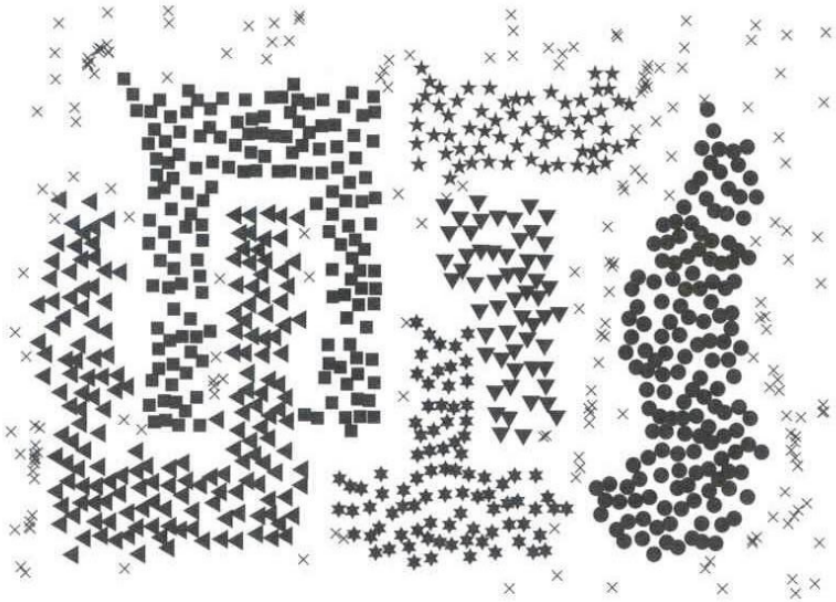
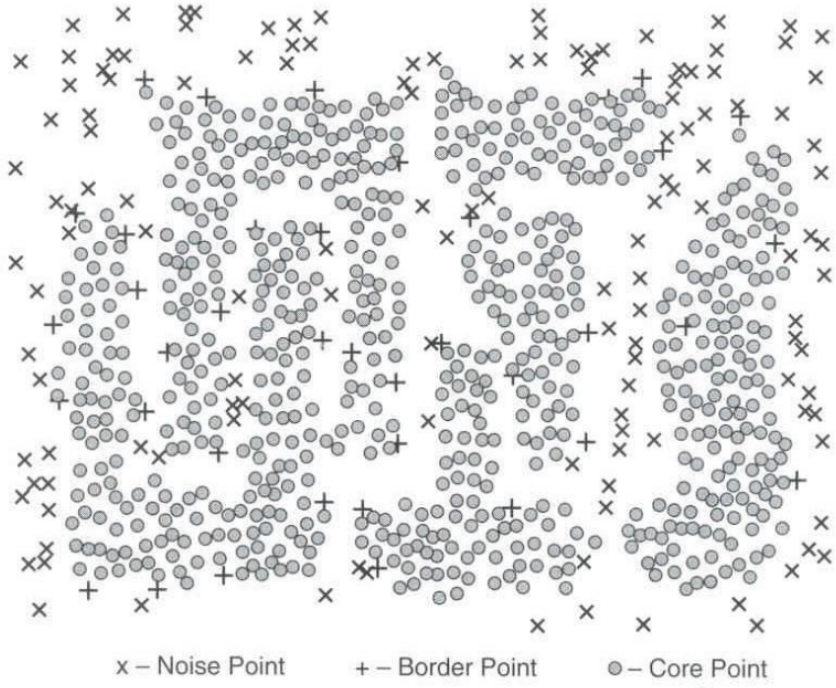


Figure 8.23. K-dist plot for sample data.

Exempel



(a) Clusters found by DBSCAN.



(b) Core, border, and noise points.

För- och nackdelar

- Brusbeständig
- Behandlar kluster av olika former och storlekar
- Problemet med kluster som har betydligt varierande tätheter
  - Svårt att välja ett bra *eps*
- Problem i stora dimensioner (curse of dimensionality)

K-means och DBSCAN

Egenskap	K-means	DBSCAN
Typ A	Partitionell	Partitionell
Typ C	Fullständig	Ofullständig
Klustertyp	Prototyp	Densitet
Klusterform	Klot	Olika
Närhetsmått	Olika	Olika
Användande av attribut	Alla	Alla
Upprepade körningar	Kluster beror på start-centroider	Samma kluster bildas
Algoritmbehov	k för antal kluster	eps och $\mathrm{m}inPts$
Optimeringsmodell	Ja	Nej
Tidskomplexitet	$O(m)$	$O(m^2)$

Faktorer som påverkar klusteranalys

Faktorer som påverkar klusteranalys

- Dimensionalitet (problem för täthetsbaserade metoder)
- Datamängdens storlek (stora datamängder är svåra att skala upp)
- Brus och extremvärden
- Skalan på data: numerisk, kategorisk mm
  - problem att välja närhetsmått för datamängder med blandade attribut
- Standardisering av variabler

Egenskaper

- Fördelningar – Olika metoder passar bättre på vissa fördelningar
- Form – Godtyckliga former är svårare att klustra
- Storlek – K-means, problem med olika storlekar
- Täthet – Olika tätheter problem för K-means, DBSCAN
- Dåligt separerade kluster – Vissa metoder slår ihop överlappande kluster
- **Ingen klustermetod passar för alla dataset!**

Utvärdera klusteranalys

Utvärdera klusteranalys

- Cluster tendency: Finns det kluster i data? Eller har obs bara slumpmässiga värden?
- Avgöra rätt antal kluster
- Interna mått på hur bra klusteranalysen är
- Externa mått på hur bra klusteranalysen är: om vi har tillgång till sanna klasser/grupper
- Jämföra olika metoder för klusteranalys på samma dataset
- Kontext och problembeskrivning $\rightarrow$ avgör om vi har en bra klustring!

Cohesion and Separation

- Interna mått
- Cohesion: hur tight eller sammanhållet ett kluster är med sig själv
- Separation: hur väl separerad ett kluster är från övriga kluster

Vi kan väga samma mått för alla kluster

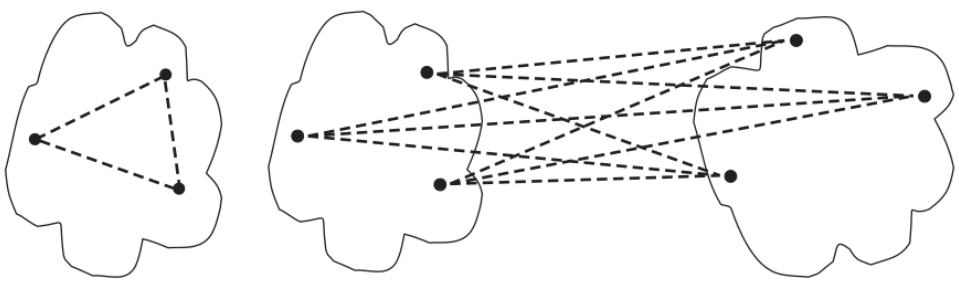
$$\textit{overall validity} = \sum_{i=1}^{K} w_i \cdot \textit{validity}\left(C_i\right)$$

Cohesion and Separation

$$cohesion(C_i) = \sum_{x_i \in C_i, y \in C_i} proximity(x, y)$$

$$separation(C_i, C_j) = \sum_{x_i \in C_i, y \in C_j} proximity(x, y)$$

- *proximity* () kan vara både närhetsmått eller avståndsmått
  - Närhetsmått: höga värden är bra för cohesion och låga värden är bra för separation



(a) Cohesion. (b) Separation.

Figure 7.27. Graph-based view of cluster cohesion and separation.

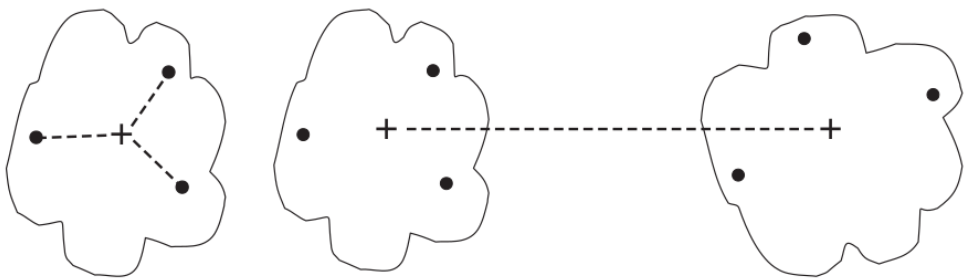
Källa: Introduction to Data Mining

Cohesion and Separation

$$cohesion(C_i) = \sum_{x_i \in C_i} proximity(x, c_i)$$

$$separation(C_i, C_j) = proximity(c_i, c_j)$$

$$separation(C_i) = proximity(c_i, c)$$



(a) Cohesion. (b) Separation.

Figure 7.28. Prototype-based view of cluster cohesion and separation.

Källa: Introduction to Data Mining

The Silhouette Coefficient

- Använder både cohesion och separation
- Metod:
  1. beräkna medelavståndet från obs_i till alla andra obs i dess kluster, kalla det a_i
  2. För alla kluster som inte innehåller obs_i, iterera över kluster:
     1. Beräkna medelavståndet från obs_i till alla andra obs i det aktuella klusteret
  3. Hitta det minsta sådana avståndet i steg 2), kalla det b_i
  4. Silhouette coefficient för obs_i defineras som

$$s_i = \frac{(b_i - a_i)}{max(a_i, b_i)}$$

The Silhouette Coefficient

$$s_i = \frac{(b_i - a_i)}{max(a_i, b_i)}$$

- s_i: ligger mellan -1 och 1
- 1 är bästa möjliga
  - Vi vill ha $a_i < b_i$, och att a_i ska ligga nära 0.
- average silhouette coefficient:
  - ta medelvärdet över alla s_i
  - ger ett mått på hur bra klusteringen är

Exempel

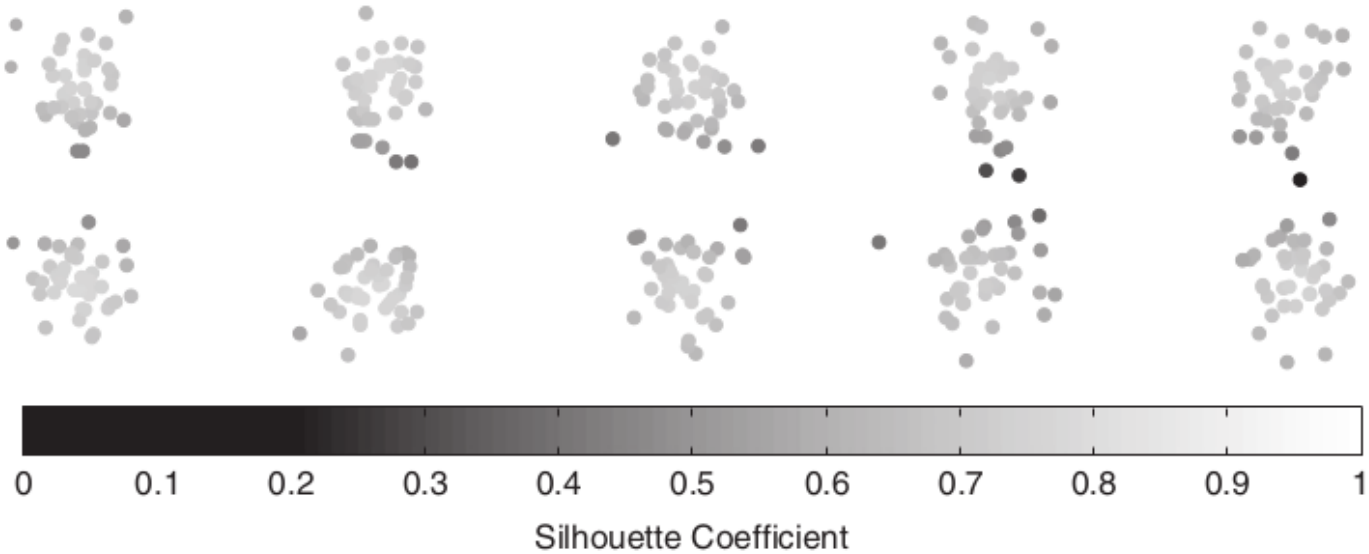


Figure 7.29. Silhouette coefficients for points in ten clusters.

Källa: Introduction to Data Mining

Välja antal kluster

- K-means: vi kan använda total SSE och average silhouette coefficient
- Plotta dessa mot antal kluster. Vi kollar efter böjar och toppar.
  - SSE planar ut efter en böj: ta antal kluster vid böjen
  - Average silhouette coefficient: kolla om det finns en eller flera toppar

Välja antal kluster

Här finns det 10 naturliga kluster i data.

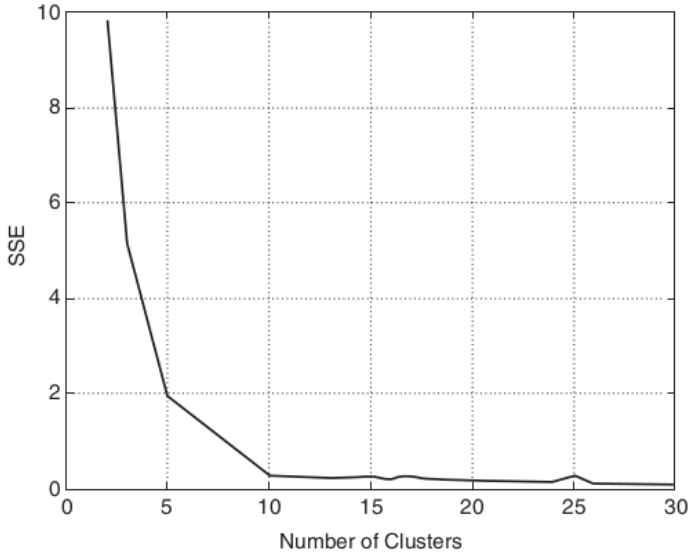


Figure 7.32. SSE versus number of clusters for the data of Figure 7.29 on page 582.

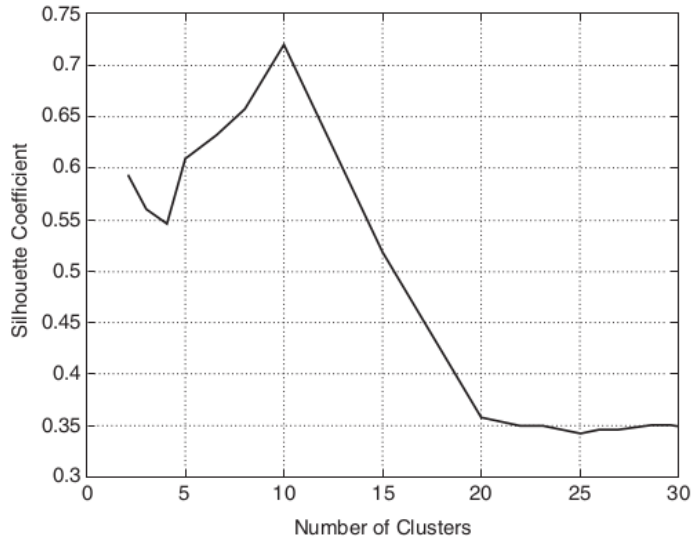


Figure 7.33. Average silhouette coefficient versus number of clusters for the data of Figure 7.29.

Källa: Introduction to Data Mining

Calinski-Harabasz Index

n_k = antal obs i kluster k, K = antal kluster, C_k= centroid för kluster k, C = centroid för hela datasetet, N = antal obs i data

- **Inter-cluster dispersion**

$$BGSS = \sum_{k=1}^{K} n_k \cdot ||C_k - C||^2$$

- **Intra-cluster dispersion**

$$WGSS_k = \sum_{i=1}^{n_k} ||X_{i,k} - C_k||^2 \qquad WGSS = \sum_{k=1}^{K} WGSS_k$$

- **Calinski-Harabasz Index**

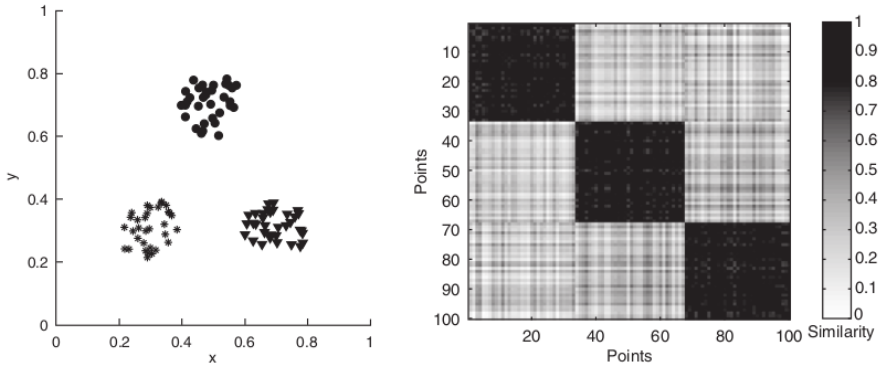
$$CH = \frac{BGSS}{WGSS} \cdot \frac{N-K}{K-1}$$

- Höga värden är bra för CH
- Annat liknande mått: Davies-Bouldin index $\rightarrow$ låga värden är bra

Välja antal kluster

- Vi kan beräkna närhetsmatrisen eller avståndsmatrisen för alla datapunkter
  - Matris med alla parvisa närheter/avstånd mellan obs.
- Notera att detta är dyrt!
  - Kostar: $O\left(n^2\right)$
  - svårt att plotta med många obs
  - en lösning är att ta slumpmässigt urval av data
- Sortera närhetsmatrisen baserat på klustertillhörighet:
  - Först kommer alla obs i kluster 1, sen alla obs i kluster 2, ...
- Om vi har väl separaerade kluster och valt ett bra antal kluster:
  - Då kommer den sorterade närhetsmatrisen vara ungefärligt blockdiagonal.

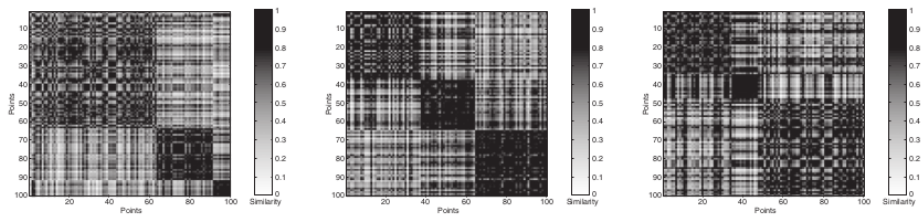
Välja antal kluster



(a) Well-separated clusters.

(b) Similarity matrix sorted by K-means cluster labels.

Figure 7.30. Similarity matrix for well-separated clusters.



(a) Similarity matrix sorted by DBSCAN cluster labels.

(b) Similarity matrix sorted by K-means cluster labels.

(c) Similarity matrix sorted by complete link cluster labels.

Figure 7.31. Similarity matrices for clusters from random data.

Källa: Introduction to Data Mining

Cluster tendancy

- Har vi slumpmässig data eller finns det något mönster (kluster)?
- Hopkins statstic:

$$H = \frac{\sum_{i=1}^{p} w_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

- Sampla två grupper om p punkter:
  - uniformt fördelat från datarymden
  - från datasetet utan återläggning
- Beräkna avstånd till närmaste granne för varje punkt i båda grupperna.
- Nollhypotesen är att datasetet följer en uniform fördelning
- Värden nära 1 indikerar på att data inte är uniformt fördelat

Extern validering

- Jämföra med sanna klasser/kluster
- Varför vill vi göra det?
- Vi kan ta resultatet från vår klusteranalys som våra "predikterade värden"
- Vi kan då jämföra med de sanna klasserna.
  - Vi kan beräkna förväxlingsmatris och liknande mått.
- Notera:
  - vi har inte de "rätta namnen" på våra kluster
  - vi vill ofta att klustren ska vara så rena som möjligt, dvs domineras av en klass

Avslut

- Kurshemsidan
- Labben