Föreläsning 7 - Klusteranalys

Josef Wilzen

2020-09-15

Outline

Intro

- Oövervakad inlärning
  - Klusteranalys
  - Associationsanalys och sekventiella mönster
- Multivariata metoder:
  - PCA
  - Hierarkisk klustring

Intro

- Målet med klusteranalys är att dela upp datamaterialet i grupper (kluster) som är intressanta och/eller användbara
- Vi vet inte i förväg vilka grupper som kommer att blidas
- Tillämpningsområden
  - Biologi (taxonomi)
  - Informationssökning (sökmotorer)
  - Psykologi och medicin
  - Kunddata

Definition av kluster

Begreppet ”kluster” är inte entydigt definierat

Klassificering och klustering

- Klassificeringsmetoder som beskrevs tidigare är exempel på övervakad klassificering – markerar nya objekt, utgår från originaldata (data, etiketter)
- Klusteranalys är ett exempel på **oövervakad** klassificering – härleder en etikett för objekt, utgår endast från data

Klustringstyper

- **Partitionell** (data är indelad i ett antal oöverlappande kluster), eller
- **Hierarkisk** (delkluster är tillåtna, kluster är representerade som ett träd)

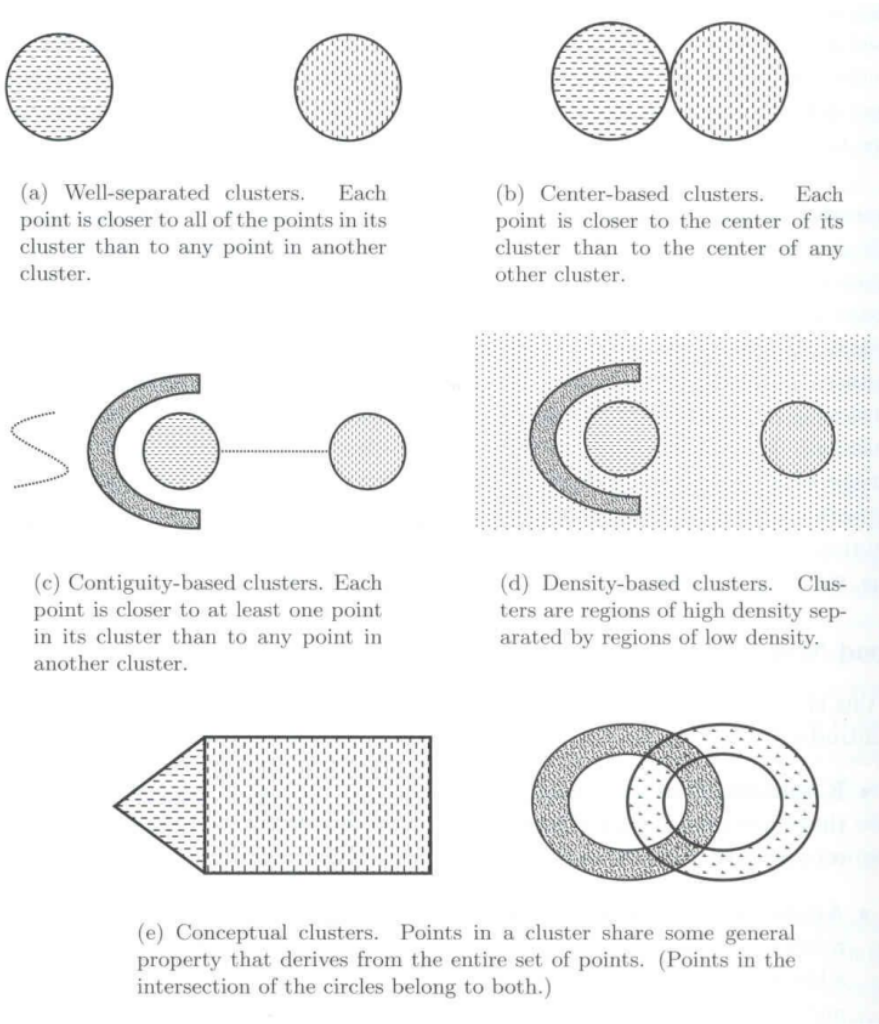
- **Uteslutande** (ett objekt tillhör ett kluster) eller
- **Överlappande** (ett objekt hör till några kluster) eller
- **Fuzzy** (Ett objekt hör till olika kluster med en specifik sannolikhet)

Klustringstyper

- **Fullständig** (varje objekt tillskrivet ett kluster)
  eller
- **Ofullständig** (somliga objekt är inte tillskrivna något kluster)

Klustertyper

- Separerade
- Angränsande/intilliggande
- Centroid- eller prototypbaserade
- Densitet- eller täthetsbaserade
- Konceptuella



(a) Well-separated clusters. Each point is closer to all of the points in its cluster than to any point in another cluster.

(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.

(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

K-means klustring

- Centroid-baserad och partitionell klustringsmetod
  - Centroid = en punkt som ska representera/sammanfatta alla obs i ett kluster
- Enkel och ofta effektiv metod
- K: hyperparameter, antalet kluster

Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.

K-means klustring

Algorithm 10.1 *K-Means Clustering*

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

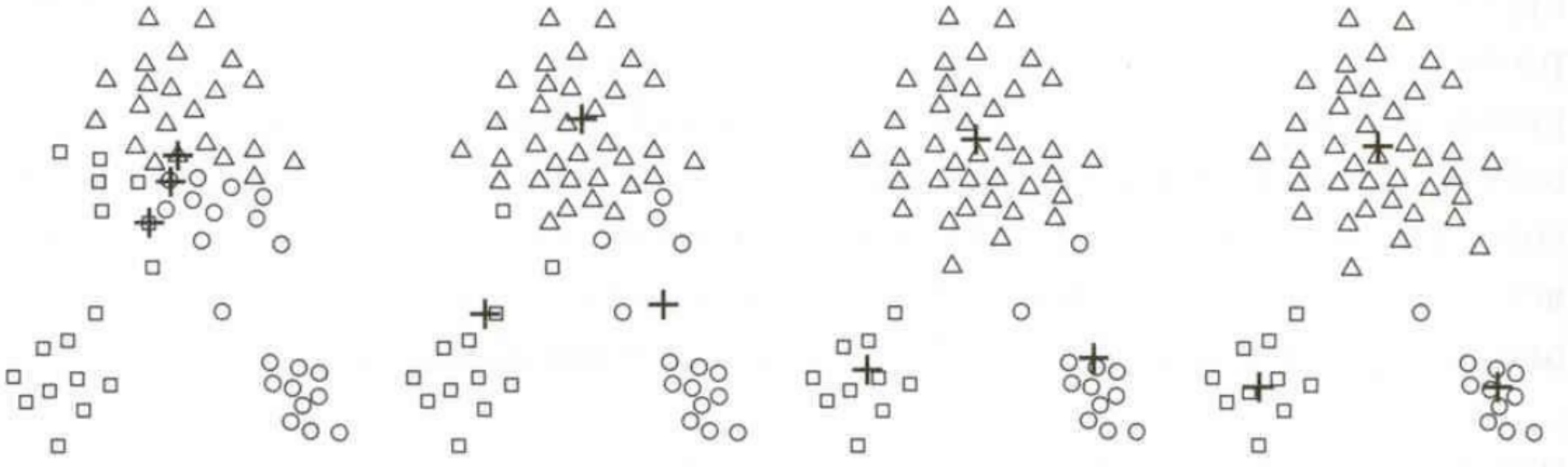
2. Iterate until the cluster assignments stop changing:

   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

Från "An Introduction to Statistical Learning with Applications in R" av Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Exempel



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

K-means

- Låt c_i vara centroider för kluster i, låt C_i vara en mängd med alla obs i kluster i
- Vi behöver definera ett avståndsmått
  - Används för att mäta avstånd mellan c_i och övriga obs
  - Vanligast är euklidiskt avstånd: låt p och q vara två vektorer

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

  - Notera: $d(p,p) = \sqrt{\sum_{i=1}^{n} p_i^2} = \sqrt{p^T p} = ||p||$ är den euklidiska normen. Ridge använder $||p||^2 = \sum_{i=1}^{n} p_i^2$

K-means

- K-means minimerar *SSE*
- SSE i ett kluster:

$$E_{c_i} = \sum_{x \in C_i} d(x, c_i)^2$$

- Totala SSE för alla kluster

$$SSE = \sum_{i=1}^{K} E_{c_i} = \sum_{i=1}^{K} \sum_{x \in C_i} d(x, c_i)^2$$

- I det euklidiska rummet beräknas centroider som

$$c_i = \frac{1}{n_i} \sum_{x \in C_i} x$$

- K-means hittar en ett lokalt minima till *SSE*

Exempel

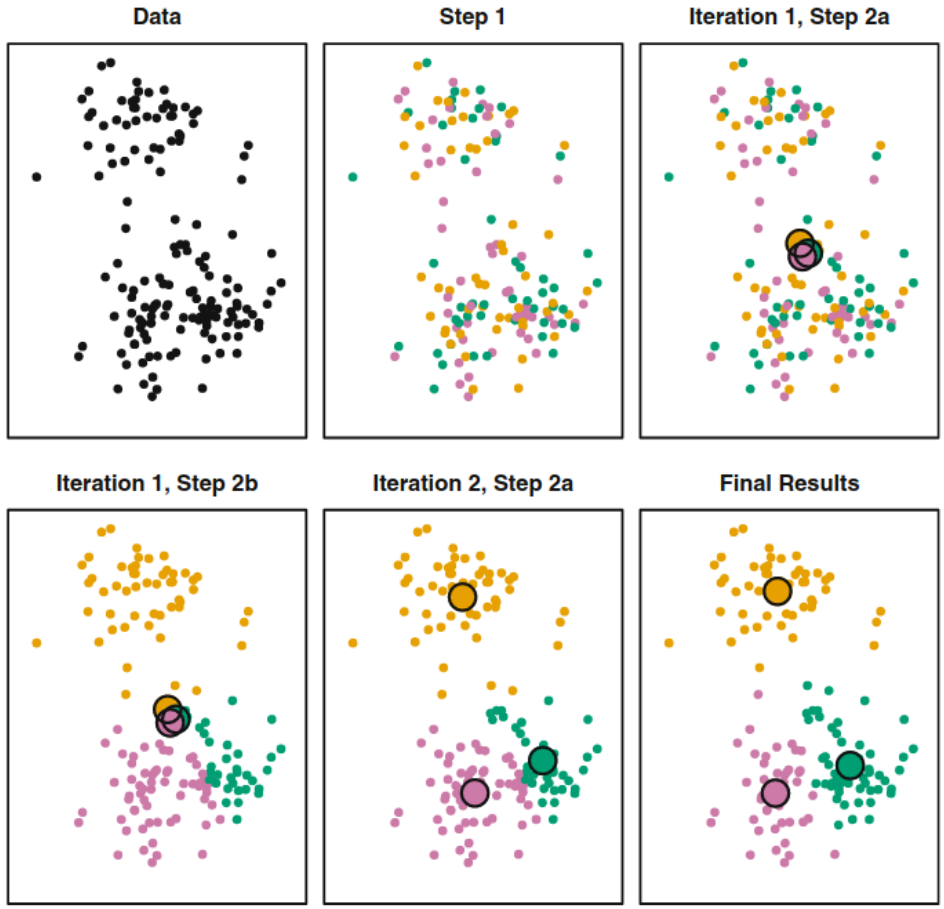
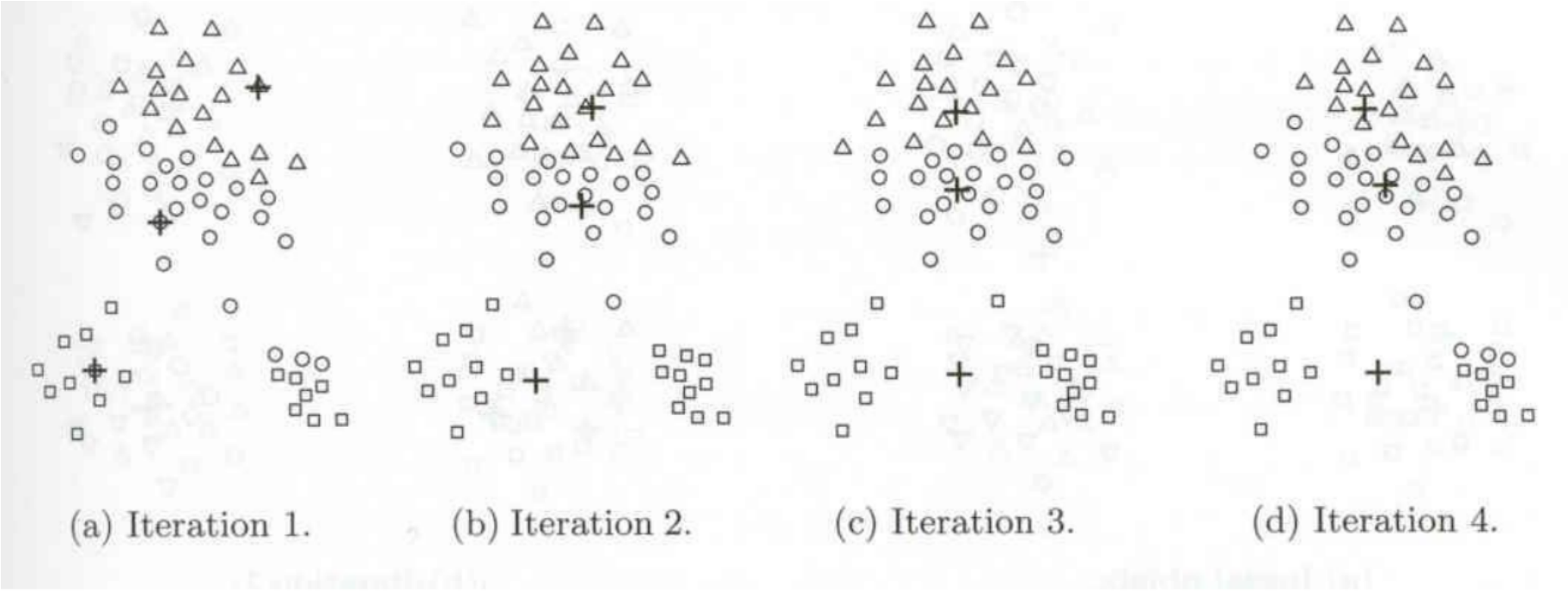


FIGURE 10.6. *The progress of the K-means algorithm on the example of Figure 10.5 with K=3.* Top left: *the observations are shown.* Top center: *in Step 1 of the algorithm, each observation is randomly assigned to a cluster.* Top right: *in Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.* Bottom left: *in Step 2(b), each observation is assigned to the nearest centroid.* Bottom center: *Step 2(a) is once again performed, leading to new cluster centroids.* Bottom right: *the results obtained after ten iterations.*

Från "An Introduction to Statistical Learning with Applications in R" av Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Startvärden

- Vi måste välja startvärden för centroiderna, valet påverkar starkt utgången av algoritmen
- Exempelvis, om man väljer dåliga startpunkter



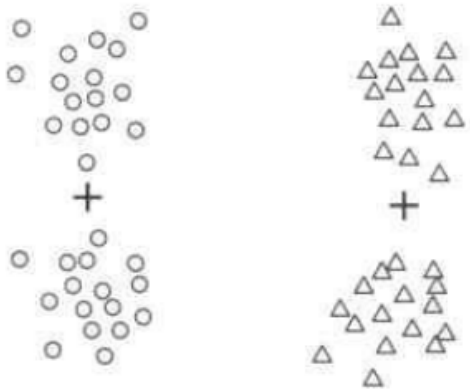
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

- Vanlig metod är att köra algoritmen många gånger med olika slumpmässiga startvärden.

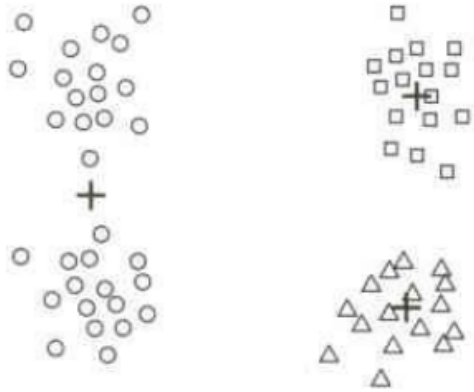
Halverande K-means

- Algoritm som motverkar problemet med val av start-centroider
- Dela upp datamängden i två kluster, välj ett och dela upp i två, välj ett (av de nuvarande 3) och dela upp osv...
  - Valet av kluster kan göras med avseende på flest observationer, störst SSE eller annat kriterie
- Uppdelningen kan liknas vid ett binärt träd

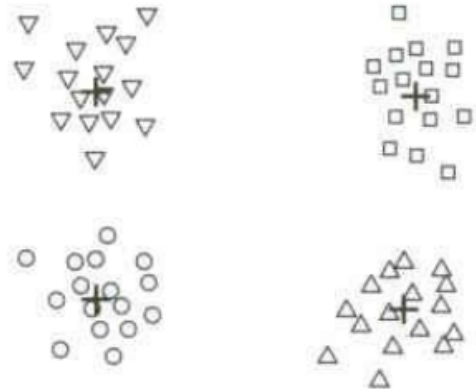
Halverande K-means



(a) Iteration 1.



(b) Iteration 2.



(c) Iteration 3.

Halverande K-means

Algorithm 8.2 Bisecting K-means algorithm.

1: Initialize the list of clusters to contain the cluster consisting of all points.
2: **repeat**
3: Remove a cluster from the list of clusters.
4: {Perform several "trial" bisections of the chosen cluster.}
5: **for** $i = 1$ to *number of trials* **do**
6: Bisect the selected cluster using basic K-means.
7: **end for**
8: Select the two clusters from the bisection with the lowest total SSE.
9: Add these two clusters to the list of clusters.
10: **until** Until the list of clusters contains K clusters.

K-means++

- Algoritm som motverkar problemet med val av start-centroider
- Kostnad för vanlig k-means: $O(n \cdot K \cdot d \cdot i)$, för många dataset $\approx O(n)$
  - n: antal obs, K: antal kluster, d: antalet variabler, i: antalet iterationer till konvergens
  - Ordo eller Big O notation, se här.
- SSE kan bli godtyckligt dåligt med k-means

K-means++

1. Välj en centroid uniformt slumpmässsigt från observationerna
2. För varje datapunkt x, beräkna avståndet $d(x, c_i)$ mellan x och den närmaste centroiden c_i som redan har valts.
3. Välj en datapunkt som centroid genom att:
    1. slumpa en punkt (som inte redan är en centroid), med hjälp av viktade sannolikheter, där vikterna är proportionella mot $d(x, c_i)^2$
4. Upprepa steg 2) och 3) tills K centroider har valts
5. Givet de valda centroiderna: kör vanlig k-means klustering

K-means++

- Generellt: k-means++ förbättrar slutgiltiga SSE mycket
- Steg 1)-4) tar extra tid att beräkna, men sen krävs det ofta mycket mindre iterationer innan vanliga k-means konvergerar i 5)
  - Vanligt att k-means++ är dubbelt så snabb som k-means med avseende på total beräkningstid

K-means: kommentarer

- Enkel och ganska effektiv
- Känslig mot initialiseringsproblem
  - Halverande K-means, k-means++
- Skapar kluster som är klotformade i $\mathbb{R}^d$ och är är linjärt spearerade
  - Andra former fungerar sämre
- Ger en centroid/prototyp för varje kluster: kan användas för att beskriva klustren
- Har svårt att identifiera kluster av olika storlekar eller med olika tätheter
- Känslig mot extremvärden

K-means: utökningar

- Kernel k-means: kan forma kluster av olika former, med icke-linjära speareringsgränser
- Gaussian mixture models/clustering:
  - Varje kluster beskrivs med en multivariat normalfördelning
  - Skattas med expectation–maximization (EM) algorithm
- k-medoids/Partitioning Around Medoids (PAM): använder medioder som center
- k-medians clustering: använder medianer istället

Densitetsbaserade metoder

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

DBSCAN

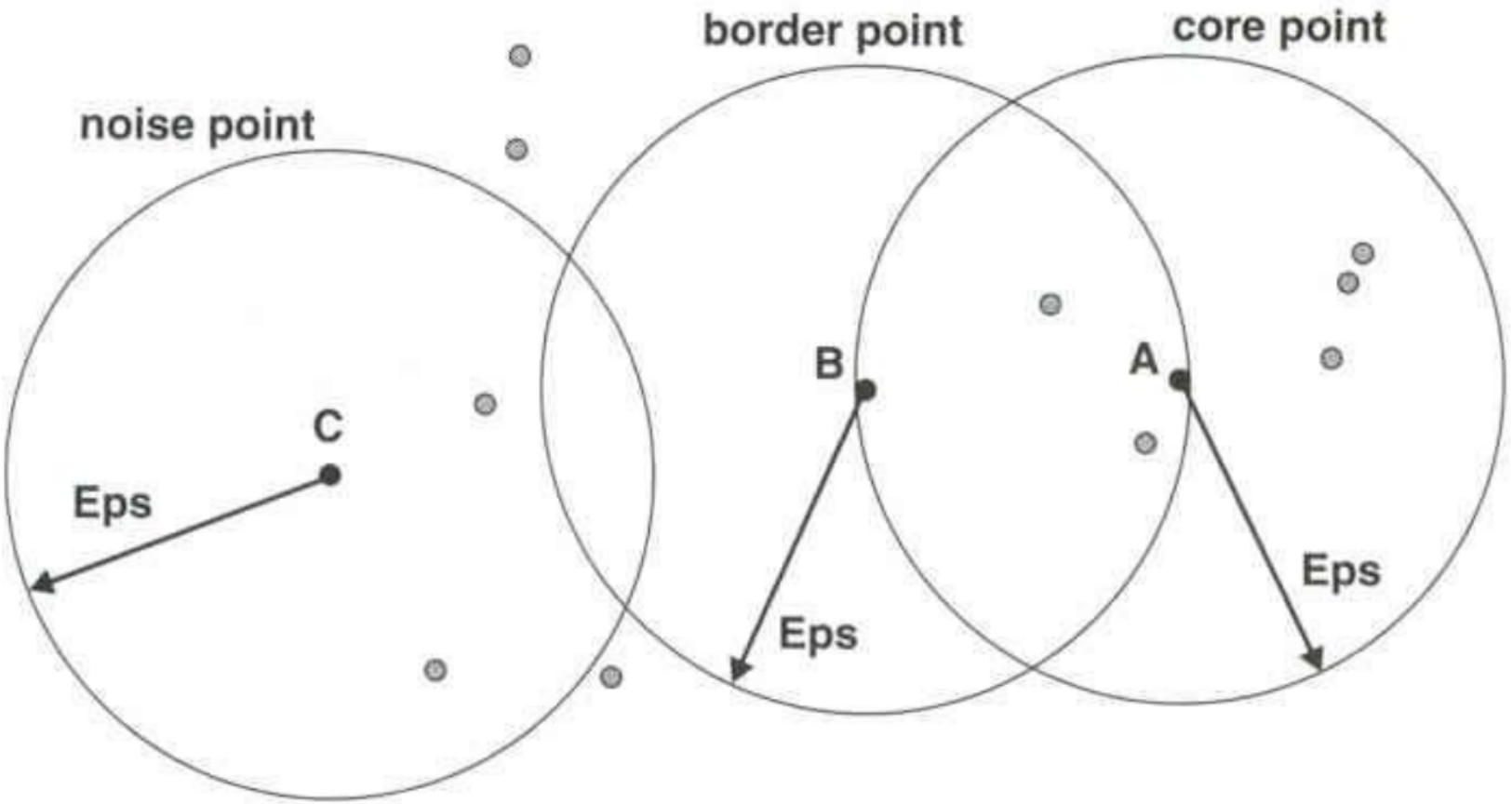
- Skapa kluster baserat på punkternas täthet
- Definitioner
  - *eps*, motsvarar en sökradie
  - *minPts*, anger minsta gräns för antalet punkter

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 avståndets ökning
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.

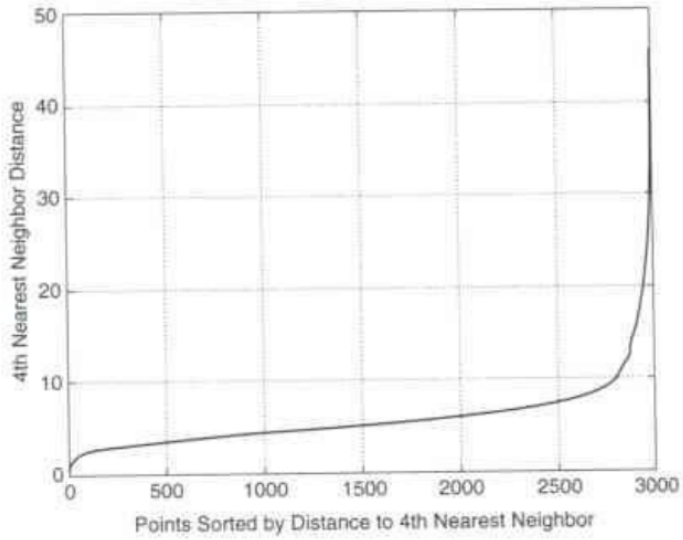
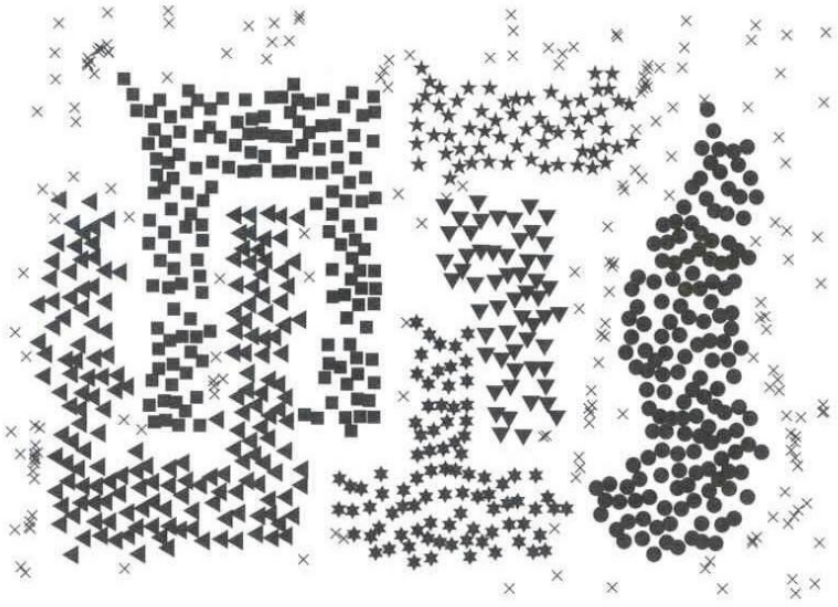
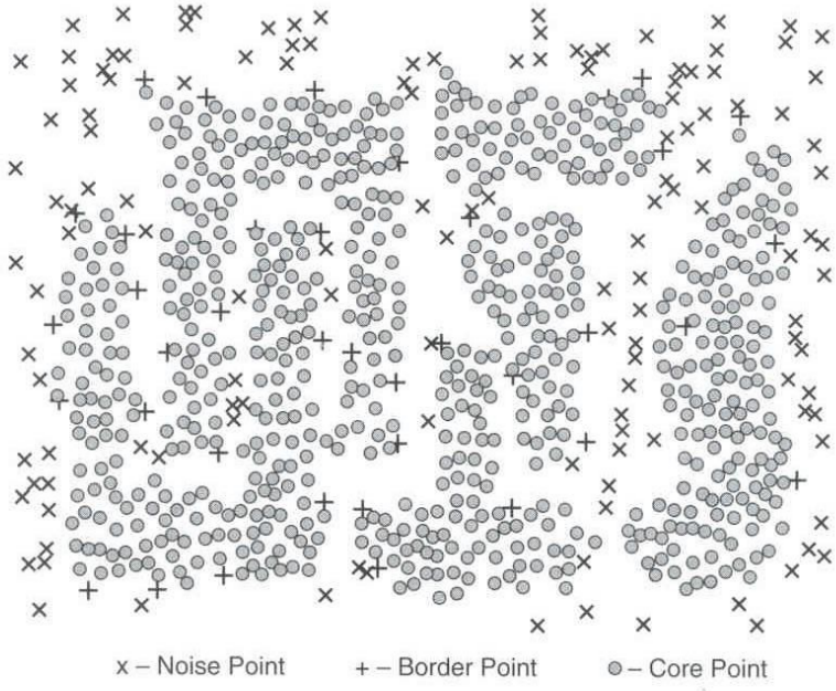


Figure 8.23. K-dist plot for sample data.

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

- 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!**

Avslut

- Kurshemsidan
- Labben