

# 732G12 Data Mining

## Föreläsning 10

---

Josef Wilzén

IDA, Linköping University, Sweden

- Projekt
- K-medoid klustring
- Densitetsbaserade metoder
- Faktorer som påverkar klusteranalys
- Utvärdera klusteranalys
- Sammanfattning av kursen

- Kursvecka 7 och framåt:
  - Arbeta med projektet
  - Datorlabbar: hjälp med projektet → utnyttja tiden!
  - Förbereda inför tentan
- Datum: [länk](#)

# K-medoid klustering

Använder **medoider** som center/prototyp vid klustering.

- En **medoid** är en representativ observation inom ett dataset/kluster.
- Medoid är **inte** samma som centroid, median, geometrisk median etc.
- Medoider är **lätta att tolka**
  - centroider kan vara punkter som inte liknar någon av observationerna i data.
- k-medoids:
  - minimerar summan av parvisa avstånd.
  - kan använda godtyckligt avståndsmått.
  - mer robust med brus och extremvärden.
- k-means: använder oftast euklidiskt avstånd.
- k-medoid klustering kallas också Partitioning Around Medoids (PAM)

---

**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:

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

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

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

$$C(i) = \operatorname{argmin}_{1 \leq k \leq K} D(x_i, m_k). \quad (14.36)$$

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

# Densitetsbaserade metoder

Kluster kan formas baserat på hur densiteten på punkter varierar över variablerna: Täta områden kan defineras som ett kluster.

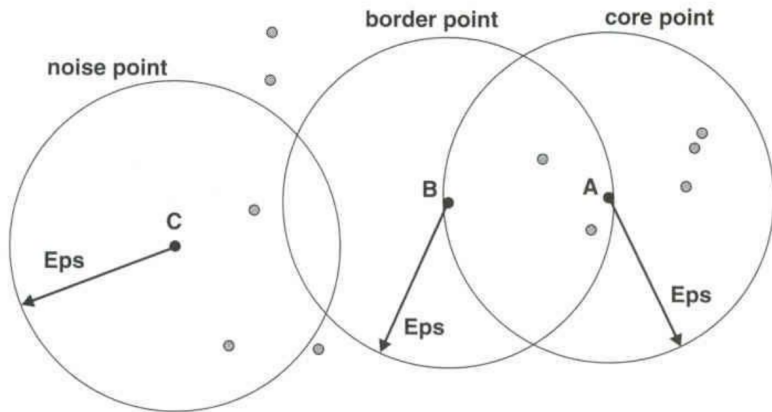


- Algoritm för att skapa kluster baserat på punkternas täther.
- Använder två begrepp:
  - $\epsilon$ , en sökradie där vi letar efter punkter.
  - minPts, minsta antal punkter/grannar.

Från detta kan vi klassificera observationerna till någon av följande kategorier:

- Kärnpunkt** Punkter med minst minPts punkter inom sökradien  $\epsilon$ .
- Gränspunkt** Inte en kärnpunkt men hamnar inom sökradien från en kärnpunkt.
- Bruspunkt** Varken kärnpunkt eller gränspunkt.

# Illustration





---

**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.
-

Hyperparametrar som vi måste välja.

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

$k$ -värdet vi valt i steg 1 påverkar **inte** eps-värdet mycket om  $k$  inte är extremt (för litet eller för stort).

# DBSCAN - Exempel



Figure 8.22. Sample data.

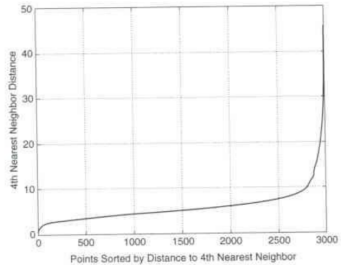
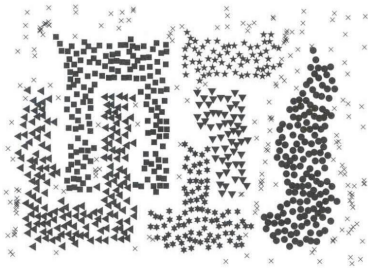
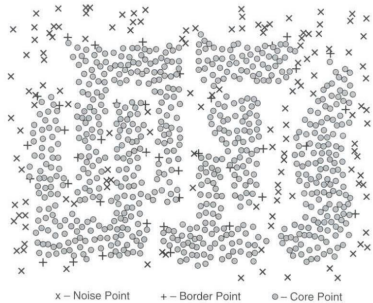


Figure 8.23. K-dist plot for sample data.

# DBSCAN - Exempel



(a) Clusters found by DBSCAN.



(b) Core, border, and noise points.

- Brusbeständig.
- Behandlar kluster av olika former och storlekar.
- Problem med kluster som har tydligt varierande tätheter.
  - Svårt att välja ett bra eps.
- Problem i stora dimensioner.

## Förbättring: HDBscan

- Variant som är en kombination av täthetsbaserad och hierarkisk klustring.
- Hyperparameter: Antal grannar, (eps bestäms automatiskt)
- Skalbar till stora dataset

# 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 $minPts$
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.
  - problem att välja närhetsmått för datamängder med blandade attribut.
- Standardisering av variabler.

- Fördelningar - Olika metoder passar bättre på vissa fördelningar.
- Form - Godtyckliga former är svårare att klustra.
- Storlek - K-means, problem med olika storlekar.
- Täthet - Olika täthet problem för K-means, DBSCAN.
- Dåligt separerade kluster - Vissa metoder slår ihop överlappande kluster eller kluster som ligger nära varandra

Ingen klustermetod passar för alla dataset!



- Cluster tendency: Finns det kluster i data? Eller har observationerna bara slumpmässiga värden?
  - Notera: det kan finnas bra ett "naturligt" kluster i data
- 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: avgör om vi har en bra klustring.

Interna mått.

**Cohesion:** Hur tight eller sammanhållet ett kluster är med sig själv.

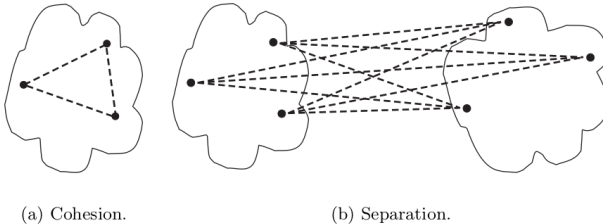
**Separation:** Hur väl separerat ett kluster är från övriga kluster.

När vi har beräknat dessa mått för ett kluster kan vi väga samman alla dessa mått till ett mått för hela klustringen.

# Cohesion och Separation

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

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



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

$\text{proximity}(x, y)$  kan vara både närhetsmått eller avståndsmått.

# The Silhouette Coefficient

Använder både cohesion och separation för att beräkna ett mått.

1. Beräkna medelavståndet från observation<sub>*i*</sub> till alla andra observationer i dess kluster, kalla det  $a_i$ .
2. Beräkna nu medelavståndet från observation<sub>*i*</sub> till alla kluster som inte innehåller denna observation.
3. Hitta det minsta av dessa avstånd kalla det  $b_i$ .
4. Silhouette coefficient för observation<sub>*i*</sub> defineras som,

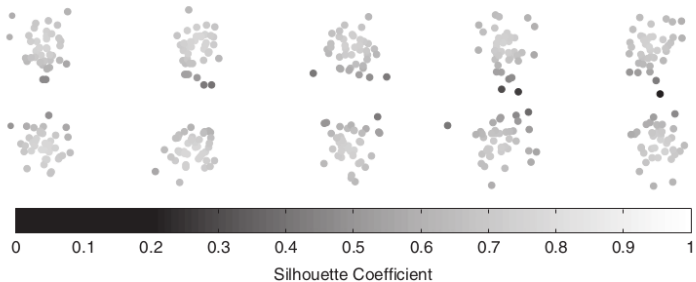
$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

# The Silhouette Coefficient

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

- $s_i$  kan ta värden mellan  $-1$  och  $1$ .
- $1$  är bästa möjliga värde.
  - Vill ha  $a_i < b_i$  och att  $a_i$  ska vara nära noll.
- Average silhouette coefficient.
  - Ta medelvärdet över alla  $s_i$
  - Ger ett mått på hur bra klustringen är.

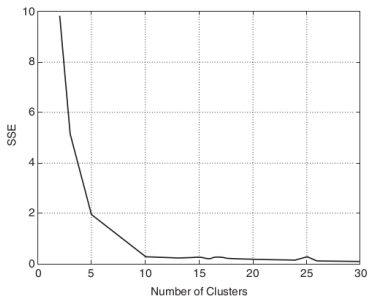
## The Silhouette Coefficient - Exempel



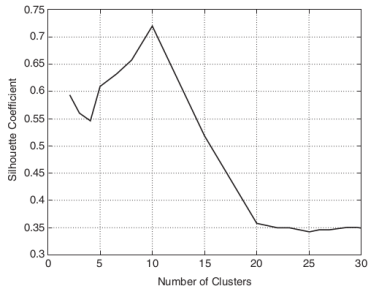
**Figure 7.29.** Silhouette coefficients for points in ten clusters.

- K-means: vi kan använda total SSE och average silhouette coefficient.
- Plotta dessa mot antal kluster.
  - Kolla 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 - Exempel



**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.



# Calinski-Harabasz Index

## Inter-cluster dispersion

$$\text{BCSS} = \sum_{k=1}^K n_k \|C_k - C\|^2.$$

## Intra-cluster dispersion

$$\text{WCSS}_k = \sum_{i=1}^{n_k} \|x_{i,k} - C_k\|^2, \quad \text{WCSS} = \sum_{k=1}^K \text{WCSS}_k.$$

## Calinski-Harabasz Index

$$\text{CH} = \frac{\text{BCSS}}{\text{WCSS}} \cdot \frac{N - K}{K - 1}.$$

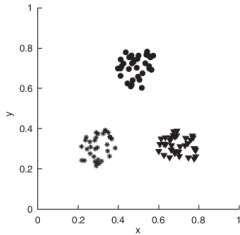
Höga värden är bra för CH.

Davies-Bouldin Index är ett liknande mått. Där ska man ha låga värden.

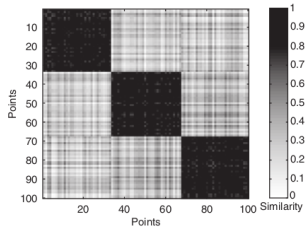
# Välja antal kluster

- Vi kan beräkna närhetsmatrisen eller avståndsmatrisen för alla datapunkter.
  - Matris med alla parvisa närheter/avstånd mellan observationer.
- Notera att detta är dyrt!
  - Kostar  $O(n^2)$
  - Svårt att plotta med många observationer.
  - En lösning är att ta ett slumpmässigt urval av data.
- Sortera närhetsmatrisen baserat på kluster.
  - Först kommer kluster 1, sen kluster 2, osv.
- Om vi har väl separerade kluster och valt ett bra antal kluster kommer den sorterade matrisen vara ungefär blockdiagonal.

# Välja antal kluster - Exempel

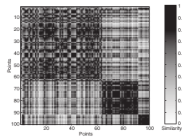


(a) Well-separated clusters.

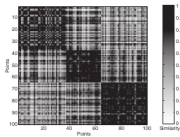


(b) Similarity matrix sorted by K-means cluster labels.

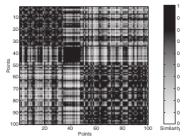
**Figure 7.30.** Similarity matrix for well-separated clusters.



(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.

# Cluster Tendency

- Har vi slumpmässig data eller finns det något mönster? (kluster)
- Sampla två grupper om  $p$  datapunkter
  - Uniformt fördelade från datarymden.
  - Från datasetet utan återläggning.
- Beräkna avståndet till närmaste granne i datasetet.
  - $u_i$  är minsta avståndet från en uniform datapunkt till en observation.
  - $w_i$  är minsta avståndet från en samplad datapunkt till en icke-samplad datapunkt.

- Hopkins statistic:

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

- Nollhypotesen är att datasetet följer en uniform fördelning.  $H_0$  kommer då vara Beta( $p, p$ ) fördelat.
- Värden nära 1 indikerar att data inte är uniformt fördelat.

- Jämför med sanna klasser/kluster.
- Vi kan ta resultatet från vår klusteranalys som våra "predikterade värden"
- Kan då jämföra med sanna klasserna.
  - Vi kan då 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.

# Användning and utvärdering av klusteranalys

- Börja med lämplig datahantering: välja variabler, skalning, mm
- Välj lämpligt avståndsmått/närhetsmått
- Överväg om någon variabelreduktion kan behövas (väja mindre delmängd av variabler, PCA, autoencoders, etc)
- Ofta är det bra att testa olika klusteringsmetoder och olika inställningar på hyperparametrar
- Vi kan använda klustering för att diskretisera variabler
  - Välj en eller ett mindre antal variabler
  - Kör en klustring
  - Använd klustertillhörighet som en ny kategorisk variabel

# Användning and utvärdering av klusteranalys

- Klusteranalys kan användas för att komprimera ett dataset. Här passar det bra med någon prototypbaserad klustering
  - Om vi har många obs så kör vi k-means eller PAM på data, där  $k$  är relativt stort
  - Vi använder sen centroiderna/medoiderna som data i vidare analyser: vi har nu  $k$  stycken obs
  - Vi kan använda den "nya datan" för klustering med någon annan klustringsmetod eller för annan dataanalys
- För att testa robusthet i resultatet:
  - Dra ett antal slumpmässiga urval från data (t.ex. med 70-90% av obs i ursprungliga data)
  - Klustra alla dessa data med samma metod med samma inställningar
  - Undersök om klustering blir lika mellan dataseten eller om det är tydliga skillnader

# Sammanfattning av kursen



Sammanfattning i en mening:

- Givet data, hitta den bästa (mest lämpade), modellen som beskriver eller predikterar detta dataset.

Till vår hjälp har vi gått igenom väldigt många olika modeller och algoritmer.

- Modellval
  - Felfunktioner
  - Utvärderingsmått
  - Dela upp data i träning, validering, test.
  - Korsvalidering
  - AIC, BIC...
  - Variabelselektion
- Regularisering
  - LASSO, Ridge
  - Vi vill ofta ha så enkla modeller som möjligt
- Vi vill ha bra generaliserbarhet!

- Icke-linjär regression/klassificering
  - Grundidé är att hitta en transformationer av förklarande variabler.
  - Gått igenom många olika transformationer.
- Basfunktioner
- Splines
- Kernelfunktioner
- Lokal regression
- Trädmodeller
- Neurala nätverk
  - Olika typer av lager för olika problem
  - Olika aktiveringsfunktioner
  - Global approximation theorem
  - Bra för bilder, video, text, ...

- Trädmodeller
  - Dela upp variabelrummet i rektanglar.
  - Varje rektangel får ett värde.
  - Olika regler för uppdelning beroende på problem.
- Beskärning av träd
  - Förbeskärning
  - Efterbeskärning
- Ensemblemetoder
  - Bagging - Använd bootstrap för att skapa många "oberoende" träd.
  - Random forest - Gör slumpmässiga ändringar i träden.
  - Boosting - Skapa många (små) träd, men modifiera datan mellan varje träd.

- K-närmaste grannar
  - Skattar värdet med hjälp av närmaste datapunkterna
  - Kan förbättras genom att vikta med avståndet
- Naive bayes
  - Skatta klassificering med hjälp av bayes sats
  - För full sannolikhetsfördelning krävs mycket data
  - Görs ofta förenklingen att varje variabel är oberoende

- Klusteranalys
  - Ööversvakad inläärning.
  - K-means klustring
  - K-medoid klustring
  - Hierarkisk klustring
  - DBSCAN

# Tack för att ni har lyssnat!

Nu är det bara projektet och tentan kvar.