Data Mining and Machine Learning

Part 1: Data Mining, Clustering, and Bayesian Learning

Melih Kandemir

University of Southern Denmark

DM566, Spring 2022

Outline

Outline

Introduction
- Data Science
- Knowledge Discovery from Data
- Data Mining Methods
- Preprocessing
- Transformation
- Evaluation
- Material
- Summary

Frequent Pattern Mining

Feature Spaces

Clustering – Basics and k-means

Classification – Basics and a Basic Classifier

Basic Probability Theory, Bayes' Rule, and Bayesian Learning

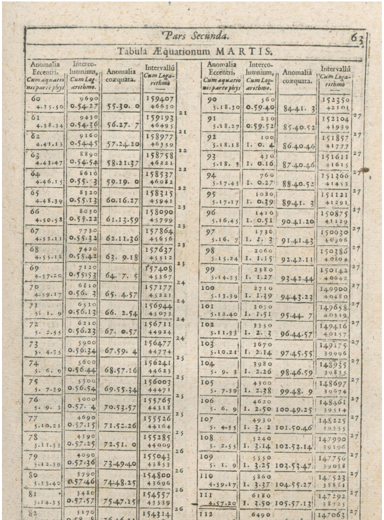
Distributions and Learning with Distributions

Outline

Learning from Data

Tabulæ Rudolphinæ

(observations by Tycho Brahe and Johannes Kepler)



Pars Secunda. 63

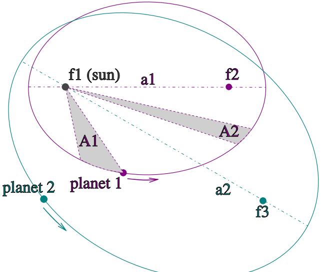
Tabula Æquationum MARTIS.

Anomalia Eccentri, Cum æquationis parte phys	Intercolumnium, Cum Logarithmo.	Anomalia coæquata.	Intervallũ Cum Logarithmo	
60 4.35.50	9690 0.54.27	55.30. 0	159407 46630	21
61 4.38.34	9430 0.54.36	56.27. 7	159193 46495	21
62 4.41.13	9160 0.54.45	57.24.20	158977 46359	22
63 4.43.47	8890 0.54.54	58.21.37	158758 46221	22
64 4.46.15	8610 0.55. 3	59.19. 0	158537 46082	22
65 4.48.39	8320 0.55.13	60.16.27	158315 45941	23
66 4.50.58	8030 0.55.22	61.13.59	158090 45799	23
67 4.53.11	7730 0.55.32	62.11.36	157864 45656	23
68 4.55.18	7420 0.55.42	63. 9.18	157637 45512	23
69 4.57.20	7120 0.55.53	64. 7. 5	157408 45367	24
70 4.59.17	6810 0.56. 3	65. 4.57	157177 45221	24
71 5. 1. 9	6510 0.56.13	66. 2.54	156944 45073	24
72 5. 2.55	6210 0.56.23	67. 0.57	156711 44924	24
73 5. 4.35	5900 0.56.34	67.59. 4	156477 44774	25
74 5. 6. 9	5600 0.56.44	68.57.16	156241 44623	25
75 5. 7.39	5300 0.56.54	69.55.34	156003 44471	25
76 5. 9. 3	5000 0.57. 4	70.53.57	155765 44318	25
77 5.10.21	4690 0.57.15	71.52.26	155526 44164	25
78 5.11.33	4390 0.57.25	72.51. 0	155285 44009	26
79 5.12.39	4090 0.57.36	73.49.40	155043 43853	26
80 5.13.40	3790 0.57.46	74.48.25	154800 43696	26
81 5.14.35	3480 0.57.57	75.47.15	154557 43539	26
82	3170	[illegible]	154314	

Anomalia Eccentri, Cum æquationis parte phys	Intercolumnium, Cum Logarithmo.	Anomalia coæquata.	Intervallũ Cum Logarithmo	
90 5.18.30	560 0.59.40	84.41. 3	152350 42101	27
91 5.18.27	230 0.59.52	85.40.52	152104 41939	27
92 5.18.18	100 1. 0. 4	86.40.46	151857 41777	27
93 5.18. 3	430 1. 0.16	87.40.46	151611 41615	27
94 5.17.43	760 1. 0.27	88.40.52	151366 41453	27
95 5.17.17	1080 1. 0.39	89.41. 3	151121 41291	27
96 5.16.45	1410 1. 0.51	90.41.20	150875 41129	27
97 5.16. 7	1730 1. 1. 3	91.41.43	150630 40966	27
98 5.15.24	2060 1. 1.15	92.42.11	150386 40804	27
99 5.14.35	2380 1. 1.27	93.42.44	150142 40642	27
100 5.13.39	2710 1. 1.39	94.43.23	149900 40480	27
101 5.12.40	3030 1. 1.51	95.44. 7	149658 40319	27
102 5.11.33	3350 1. 2. 3	96.44.57	149416 40157	27
103 5.10.21	3670 1. 2.14	97.45.55	149175 39996	27
104 5. 9. 3	3980 1. 2.26	98.46.59	148935 39835	27
105 5. 7.39	4300 1. 2.38	99.48. 9	148697 39674	27
106 5. 6. 9	4620 1. 2.50	100.49.25	148461 39514	27
107 5. 4.35	4930 1. 3. 2	101.50.46	148225 39355	27
108 5. 2.55	5240 1. 3.14	102.52.14	147990 39196	27
109 5. 1. 9	5550 1. 3.25	103.53.47	147756 39038	27
110 4.59.17	5860 1. 3.37	104.55.27	147523 38881	27
111 4.57.20	6180 1. 3.50	105.57.13	147292 38725	27
112	6490		147063	27

Kepler's laws
of planetary motion

$\Longrightarrow$



Data Science

https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

Data Science

https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

- "data science" is a vaguely defined concept

Data Science

https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

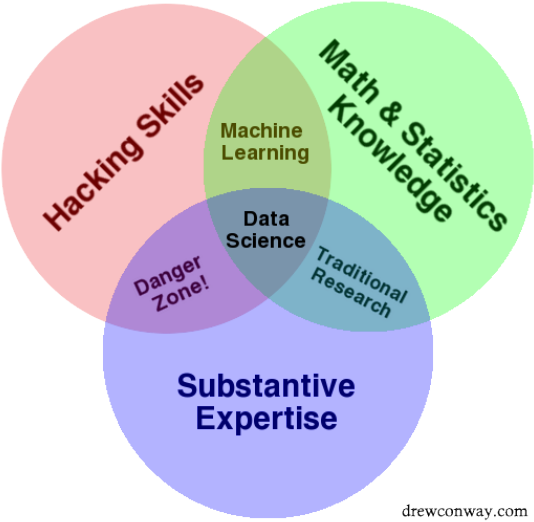
- “data science” is a vaguely defined concept
  - meaning A: doing science based on (big) data

Data Science

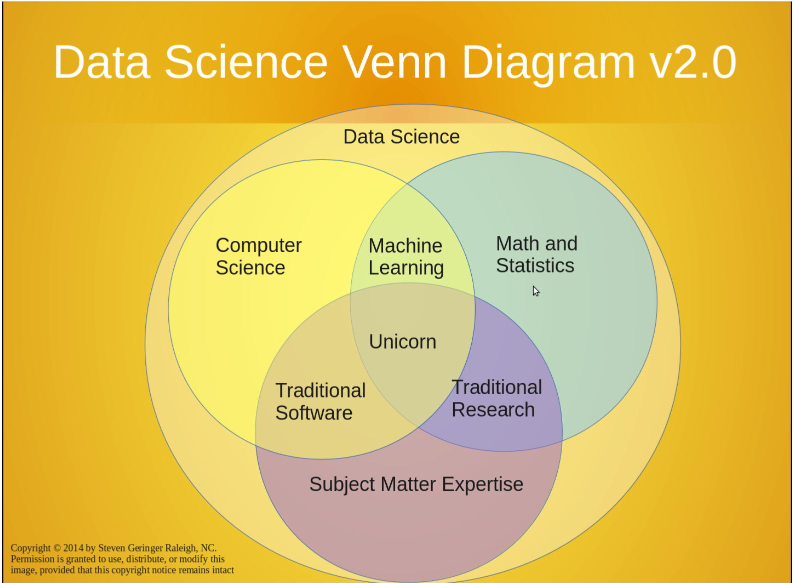
`https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century`

- "data science" is a vaguely defined concept
  - meaning A: doing science based on (big) data
  - meaning B: the science of working with data

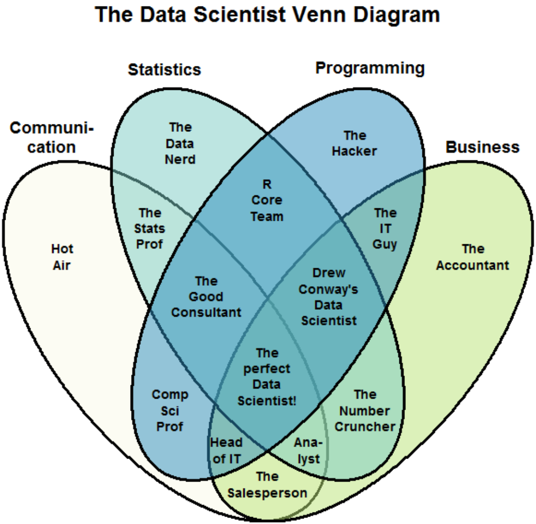
Data Science in Venn Diagrams



Data Science in Venn Diagrams



Data Science in Venn Diagrams



Data Science in Venn Diagrams

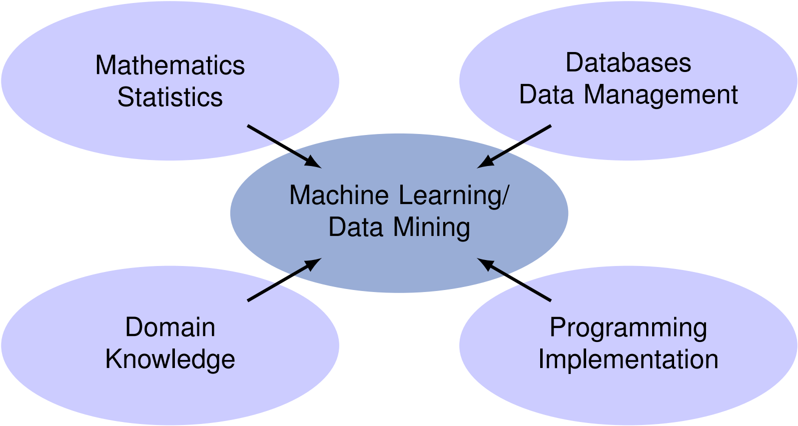
Recommended Reading:

Entertaining post on attempts to describe data science by Venn diagrams:

- `http://www.kdnuggets.com/2016/10/battle-data-science-venn-diagrams.html`

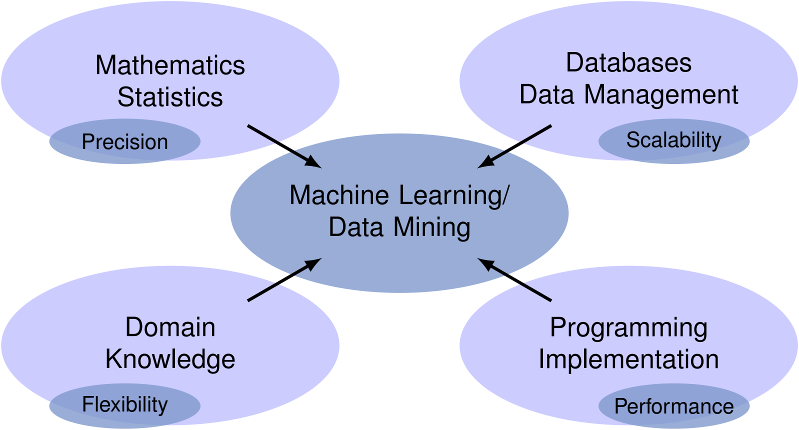
(The diagrams shown on the previous slide were taken from there.)

Data Mining and Related Areas



Data Mining tries to combine the strengths of many domains.

Data Mining and Related Areas



Data Mining tries to combine the strengths of many domains.

Understand Methods for Analyzing Data

Illustration by David Parkins (detail).

Source: Silberzahn and Uhlmann [2015].

- data science: learning from data, finding patterns in data, understanding databases

Understand Methods for Analyzing Data

Illustration by David Parkins (detail).

Source: Silberzahn and Uhlmann [2015].

- data science: learning from data, finding patterns in data, understanding databases
- data mining/machine learning: computational *methods* for learning from data

Understand Methods for Analyzing Data

Illustration by David Parkins.

Source: Silberzahn and Uhlmann [2015].

- data science: learning from data, finding patterns in data, understanding databases
- data mining/machine learning: computational *methods* for learning from data
- different methods deliver different pictures of the data

Understand Methods for Analyzing Data

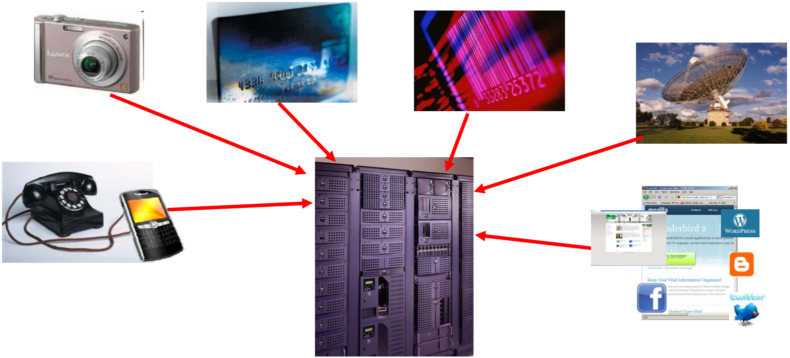
Illustration by David Parkins.

Source: Silberzahn and Uhlmann [2015].

- data science: learning from data, finding patterns in data, understanding databases
- data mining/machine learning: computational *methods* for learning from data
- different methods deliver different pictures of the data
- this course should help:
  - to learn about data mining/ machine learning methods
  - to understand their characteristics
  - to apply them correctly
  - to derive meaningful results

Outline

Motivation: Knowledge Discovery from Data



- huge amount of data collected in various application domains
- manual analysis?

Definition: Knowledge Discovery from Data

"KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data." Fayyad et al. [1996]

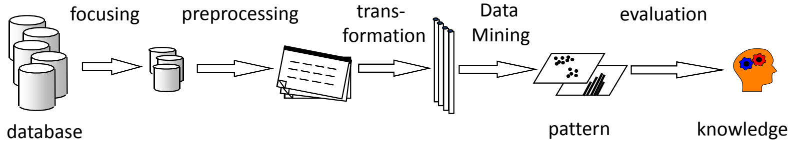
- *data*: set of facts (e.g., entries in a database)
- *pattern*: expression in some language to describe a data subset (e.g., mathematical model)
- *process*: can involve several steps or iterations
- *nontrivial*: more complex than search, inference, simple aggregations
- *valid*: applicable to new data with a certain degree of reliability
- *novel*: for the system, better: for the user
- *potentially useful*: beneficial for user of application
- *ultimately understandable*: if not immediately then given some post processing

understandability $\Leftrightarrow$ simplicity?

(validity, novelty, usefulness, simplicity) $\Leftrightarrow$ "interestingness"

The KDD process model

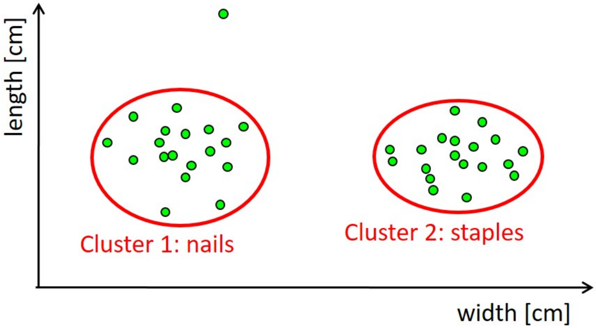
KDD process model (cf. Fayyad et al. [1996])



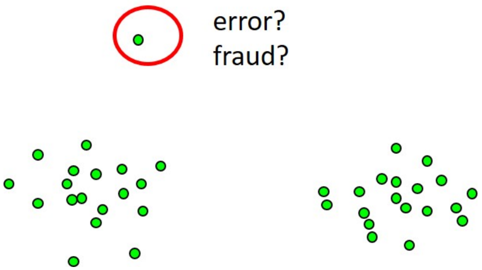
- *focusing*
  - get the data
  - organize data (file/database)
  - select relevant data
- *preprocessing*
  - integrate heterogeneous data
  - check for completeness
  - check for consistency
- *transformation*
  - discretize numeric attributes
  - infer new attributes
  - select relevant attributes
- *data mining*
  - generate patterns or modells
- *evaluation*
  - assess “interestingness” for the user
  - validate models statistically

Outline

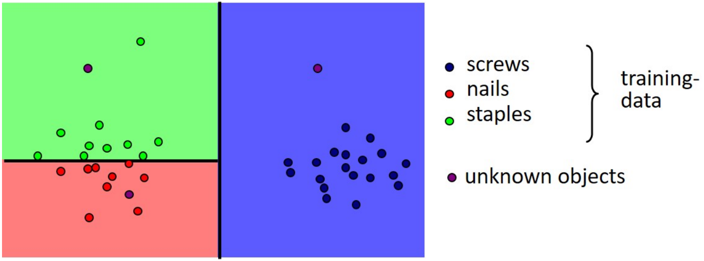
Example: Clustering



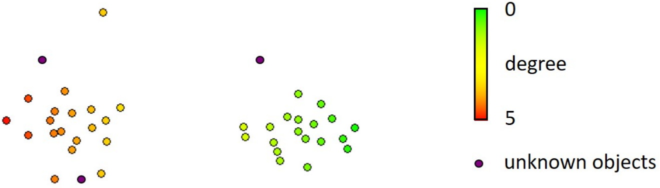
Example: Outlier



Example: Classification



Example: Regression



Example: Frequent Patterns

a,**b,c,d**,e
b,c,d
a,**b,c,d**
c,d,f
a,**b,c,d**,e
a,c,d
a,c,e,f
c,d,e,f
a,**b,c,d**,f
a,b,e,f

→

In 5 out of 10 (50%) cases,

b,c,d

occur together.

→

In 5 cases we have **b,c**, and in all those 5 cases we also have **d**:

Rule with 100% confidence:

If **b,c** *are in the set, then also* **d** *is in the set.*

Data Mining Techniques: From Data to Knowledge

Data	Method	Knowledge
	clustering	customer profile
	outlier detection	fraud
	frequent pattern mining	product groups
	classification	star type

Patterns and Models

- *predictive models*:
  A predictive model should describe (known) data in a way suitable to make predictions on unknown data.
- *descriptive models*:
  A model should provide insight in the data, help to understand the underlying structure, correlations, properties etc.
- A data mining model can be both, descriptive and predictive.
- Based on both could be a *prescriptive model*:
  Learn from the data for the future: a model to tell you what to change, or how to implement details (typically related to optimization problems).

Data Mining Techniques: Categories of Learning

- *supervised* (e.g., classification, regression, outlier detection):
  An in general unknown attribute is learned, based on examples (training data) where the attribute is known. Emphasis on predictive modeling.
- *unsupervised* (e.g., clustering, outlier detection, association rules)
  The data are distinguished/organized in different groups without previously known examples. Emphasis on descriptive modeling.
- *semi-supervised* (e.g., clustering, outlier detection)
  Semi-supervised techniques are guided by using *some* information (e.g., only one class is known, or some constraints restrict clustering results)

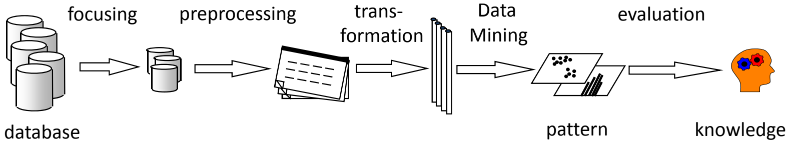
Note that:

KDD is targeted from very different perspectives, such as researchers in statistics, in machine learning, in databases, and in applied areas such as bioinformatics or economy. Thus, terminology is not always used in the same way.

Outline

The KDD process model

KDD process model (cf. Fayyad et al. [1996])



- *focusing*
  - get the data
  - organize data (file/database)
  - select relevant data
- *preprocessing*
  - integrate heterogeneous data
  - check for completeness
  - check for consistency
- *transformation*
  - discretize numeric attributes
  - infer new attributes
  - select relevant attributes
- *data mining*
  - generate patterns or modells
- *evaluation*
  - assess “interestingness” for the user
  - validate models statistically

Preprocessing

Data may be

- noisy in the sense of containing errors, outliers
- noisy in the sense of containing lots of irrelevant information
- incomplete (e.g., missing values, missing attributes, that would have been particularly interesting for some given task)
- inconsistent (e.g., different scaling in student evaluation sheets for different cohorts, different questions in questionaires for different universities)

Typical Preprocessing Tasks

- data cleaning: impute missing values, smoothing of noisy values, identify or remove outliers, resolve inconsistencies
- data integration: combination from different data sources: entity identification, value resolution
- data reduction: elimination of duplicates

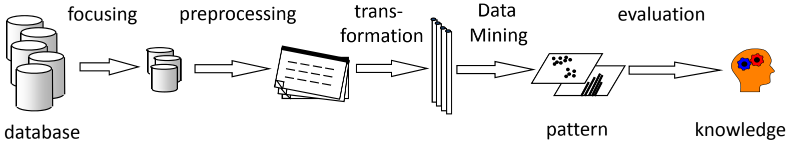
Note that:

Many of these operations do actually change the data, based on some assumptions – handle with care if these assumptions are not explicit!

Outline

The KDD process model

KDD process model (cf. Fayyad et al. [1996])



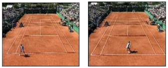
- *focusing*
  - get the data
  - organize data (file/database)
  - select relevant data
- *preprocessing*
  - integrate heterogeneous data
  - check for completeness
  - check for consistency
- *transformation*
  - discretize numeric attributes
  - infer new attributes
  - select relevant attributes
- *data mining*
  - generate patterns or modells
- *evaluation*
  - assess “interestingness” for the user
  - validate models statistically

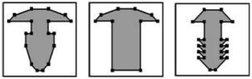
Data Descriptions

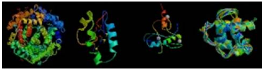
- Many techniques work on feature attributes (feature vectors).
- Other techniques work directly on complex data such as text, sets, graphs.
- If we are to perform data mining on some complex objects, it is an important preprocessing step to derive meaningful features to describe these objects.

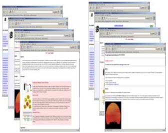
Similarity

- Similarity (as given by some distance measure) is a central concept in data mining, e.g.:
  - clustering: group similar objects in the same cluster, separate dissimilar objects to different clusters
  - outlier detection: identify objects that are dissimilar (by some characteristic) from most other objects
- definition of a suitable distance measure is often crucial for deriving a meaningful solution in the data mining task
  - images
  - CAD objects
  - proteins
  - texts
  - ...



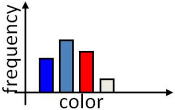




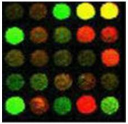


Deriving Features from Complex Objects

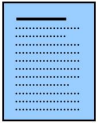
- image database
  ↓
  color histograms



- gene database
  ↓
  expression levels



- text database
  ↓
  word counts



Data	25
Mining	15
Feature	12
Object	7
...	

Note that:

Data mining methods work on the derived feature space no matter the original nature of the object – thus the mapping to a meaningful feature space is of paramount importance.

Typical Transformation Tasks

- scale, normalize, generalize (e.g., by concept hierarchy)
- data reduction: aggregation, feature combination, dimensionality reduction
- derive new features

Note that:

Many of these operations do actually change the data, based on some assumptions – handle with care if these assumptions are not explicit!

Representation Matters

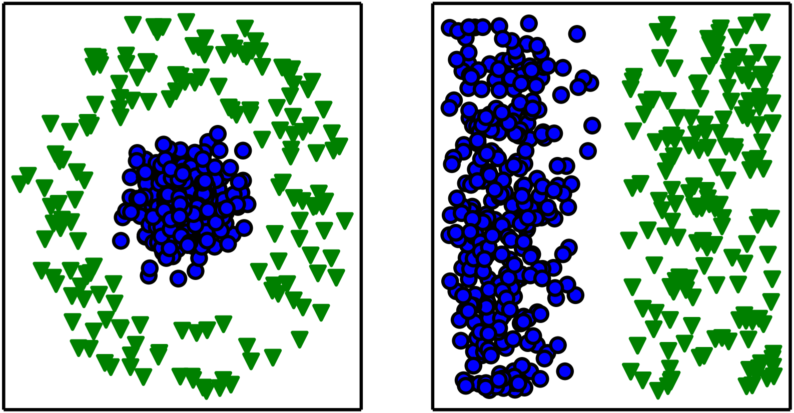
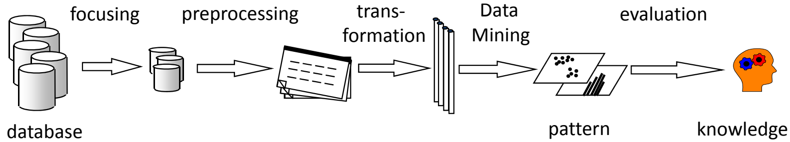


Figure adapted from Figure 1.1, Goodfellow et al. [2016].

Outline

The KDD process model

KDD process model (cf. Fayyad et al. [1996])



- *focusing*
  - get the data
  - organize data (file/database)
  - select relevant data
- *preprocessing*
  - integrate heterogeneous data
  - check for completeness
  - check for consistency
- *transformation*
  - discretize numeric attributes
  - infer new attributes
  - select relevant attributes
- *data mining*
  - generate patterns or modells
- *evaluation*
  - assess “interestingness” for the user
  - validate models statistically

Evaluation

"KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data." [Fayyad et al., 1996]

- The challenge of evaluation: to quantify how *valid, novel, useful and understandable* the derived patterns and models are.
  - Evaluation of descriptive models (clustering)?
  - Evaluation of predictive models (classification)?

Note that:

There is something like a 'Heisenbergian uncertainty relation' between validity and novelty.

Outline

Resources

- overview on free and commercial tools:
  `http://www.kdnuggets.com/software/suites.html`
- `http://www.kdnuggets.com/2015/03/machine-learning-table-elements.html`
- commercial tools offered by database companies (such as Oracle, IBM, Microsoft)
- some free / open source tools:

Environment for DeveLoping KDD-Applications Supported by Index-Structures

WEKA The University of Waikato

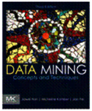
SciPy.org

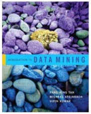
Books (General Textbooks)

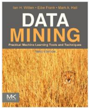
- Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. 3rd edition, Morgan Kaufmann, 2012
- Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, 2006
- Mitchell, T. M.: Machine Learning. McGraw-Hill, 1997
- Witten, I. H., Frank, E., Hall, M. A.: Data Mining: Practical Machine Learning Tools and Techniques. 3rd edition, Morgan Kaufmann, 2011
- Zaki, M. J ., Meira Jr., W.: Data Mining and Analysis. Cambridge, 2014

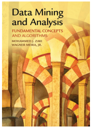
None is required, all might be helpful.

See also information on itslearning (syllabus) or find them on the semester shelf.









Course Material

On itslearning you find furthermore:

- details about the exam
- these lecture slides (updated over time)
- weekly exercises (to be prepared before the tutorial!)
- data sets we use in the lecture and in exercises
- later on: projects

Outline

Summary

You learned in this section:

- *steps for analyzing data (KDD process model)*
- *“data science”, “data mining”, “machine learning”*
- *descriptive and predictive models*
- *supervised, unsupervised, semi-supervised*
- *paradigmatic data mining methods: clustering, outlier detection, classification, regression, frequent pattern mining and association rules*
- *typical tasks of preprocessing*
- *data transformation: data representation and similarity measures (motivating examples)*
- *evaluation: the fundamental problem of measuring validity and novelty at the same time*

Outline

SDU SYDDANSK UNIVERSITET

DM566

Melih Kandemir

Introduction
Freq. Pattern Mining
Sets and Relations
Item Sets
Definitions and Problem-Statements
Monotonicity-Property of Frequency
The Apriori Algorithm
Example
Efficiency
Association Rule Mining
Summary
Feature Spaces
Clustering – Basics
Classification – Basics
Bayesian Learning
Learning with Distributions
References

Recommended Reading

Recommended Reading:

- *Tan et al. [2006], Chapter 6; Tan et al. [2020], Ch. 4.*
- *Han et al. [2011], Chapter 6.*
- *Zaki and Meira Jr. [2014], Chapters 8+9.*
- *Witten et al. [2011], Chapter 4.5.*
- *Advanced topics: Aggarwal and Han [2014].*

Outline

Definition of Sets

A set is a collection of objects (e.g., $S = \{1, 4, 9\}$).

- The objects are said to be elements of the set (e.g., $1 \in S, 4 \in S, 9 \in S$). Each element is *unique*.

We can define sets

extensionally: by enumerating the elements that define the set (e.g., $S = \{1, 4, 9\}$).

intensionally: by characterizing the elements of the set,

- describing what condition (the "*characteristic function*" of the set) holds for all the elements and only for the elements of the set (e.g. $S = \{x | \sqrt{x} \in \mathbb{N}$ *AND* $x < 15\}$ — read '|' as 'for which holds' or 'such that').
- The intensional definition typically resorts to a domain over which the set is defined (here: $\mathbb{N}$).

Properties of Sets

- A set can be finite (e.g., $S = \{1, 2, 3, \ldots, n\}$: $1 \in S, 2 \in S, 3 \in S, \ldots, n \in S$). In this case, the set has a finite size (*cardinality*), that is the number of elements of the set (e.g., $|S| = n$).
- A set can be countably infinite. Example:

$$\begin{aligned} S &= \{1, -1, 3, -3, 5, -5, \ldots\} \\ &= \{x | x = 2k + 1 \textit{ OR } x = -2k + 1, k \in \mathbb{N}_0\} \end{aligned}$$

- A set can be uncountable (e.g., $\mathbb{R}$).
- A set can be empty: $S = \{\} = \emptyset$. $|S| = 0$.

Notations

- logical operators:
  - $\vee$: or
  - $\wedge$: and
  - $\neg$: not
  - $\not x$: negation of x (e.g., $\neq$, $\notin$)
  - $A \Rightarrow B$: if A, then B (short for $\neg(A \wedge \neg B)$)
  - $A \Leftarrow B$: if B, then A (short for $\neg(B \wedge \neg A)$)
  - $A \iff B$: $A \Rightarrow B \wedge A \Leftarrow B$ ("iff", read: "if and only if", also: $\equiv$, read: "is equivalent to")
- $\exists x : p(x)$, means: there exists some x such that $p(x)$
- $\forall x : p(x)$, means: for all x holds $p(x)$
- subset: $T \subseteq S \equiv \forall x \in T : x \in S$
- proper subset: $T \subset S \equiv (\forall x \in T : x \in S) \wedge (\exists x \in S : x \notin T)$

Algebra of Sets

An algebra is defined over a base set Ω, all sets involved in the algebra are subsets of Ω.

basic operations for $S, T \subseteq \Omega$:

- **union** $S \cup T \equiv \{x | x \in S \vee x \in T\}$
- **intersection** $S \cap T \equiv \{x | x \in S \wedge x \in T\}$
- **complement** $\bar{S} \equiv S^C \equiv \{x | x \notin S\}$
- **difference** $S \setminus T \equiv \{x | x \in S \wedge x \notin T\}$
- **product** $S \times T \equiv \{(x, y) | x \in S \wedge y \in T\}$

Powerset $\mathcal{P}(S) \equiv \wp(S) \equiv 2^S \equiv \{T | T \subseteq S\}$

example Let $\Omega = \mathbb{N}$, $S = \{1, 2, 3\}$ and $T = \{2, 3, 4\}$ – what are the values of all these expressions?

Tupel and Relations

n-tupel result of the (Cartesian) product of n sets:

$$S_1 \times S_2 \times S_3 \times \ldots \times S_n =$$
$$\{(a_1, a_2, a_3, \ldots, a_n) | a_1 \in S_1, a_2 \in S_2, a_3 \in S_3, \ldots, a_n \in S_n\}$$

If $S_1 = S_2 = S_3 = \ldots = S_n$, we write:

$$S^n \equiv S_1 \times S_2 \times S_3 \times \ldots \times S_n$$

n-ary relation R is a set of n-tupels:

$$R(x_1, \ldots, x_n) \equiv (x_1, \ldots, x_n) \in R$$

characteristic function of $R \subseteq S_1 \times \ldots \times S_n$ (can also be seen as “predicate”: “R is true/false”)

$$S_1 \times \ldots \times S_n \rightarrow \{\text{true}, \text{false}\}$$

$$t \mapsto \begin{cases} \text{true} & \text{if } t \in R \\ \text{false} & \text{otherwise} \end{cases}$$

Properties of Homogeneous Relations

Relation $R \subseteq S_1 \times S_2 \times S_3 \times \ldots \times S_n$ is homogeneous iff $S_1 = S_2 = S_3 = \ldots = S_n$.

A binary homogeneous relation $R \subseteq S \times S$ is

reflexive iff $\forall x \in S : (x, x) \in R$

symmetric iff $\forall x, y \in S : (x, y) \in R \Rightarrow (y, x) \in R$

antisymmetric iff $\forall x, y \in S : (x, y) \in R \wedge (y, x) \in R \Rightarrow x = y$

transitive iff $\forall x, y, z \in S : (x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$

total iff $\forall x, y \in S : (x, y) \in R \vee (y, x) \in R$

Examples:

- '$<$' $\subseteq \mathbb{N} \times \mathbb{N}$: $(1, 2) \in$ '$<$'
  (more customary is the infix notation $1 < 2$)
- '$=$' $\subseteq \mathbb{N} \times \mathbb{N}$: $(361, 361) \in$ '$=$' or $361 = 361$
- '$\subset$' $\subseteq \wp(S)^2$

Partial and Total Order

A relation $R \in S \times S$ can be an order:

R is a partial order if R is antisymmetric and transitive.

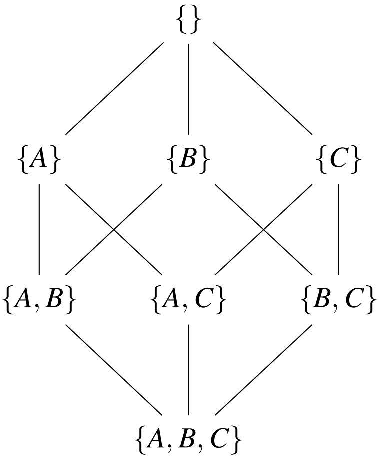
R is a total order if R is a partial order and is total.

R is a strict order if R is a partial order and is not reflexive

Are these relations orders? Which kind of order?

- alphanumeric sorting of strings?
- '$<$' $\subseteq \mathbb{N} \times \mathbb{N}$
- '$\leq$' $\subseteq \mathbb{N} \times \mathbb{N}$
- '$\subset$' $\subseteq \wp(S)^2$
- '$\subseteq$' $\subseteq \wp(S)^2$

Lattice of Subsets



Functions

- A *function* is a mapping from some set (the domain) to some set (the image).
- We can see functions as relations with particular properties: a *univalent* (or *right-unique*) relation over domain and image.
- Formally, a function f is a binary relation over $D \times I$: $f \subseteq D \times I$, for which holds:

$$(x, y) \in f \wedge (x, z) \in f \Rightarrow y = z$$

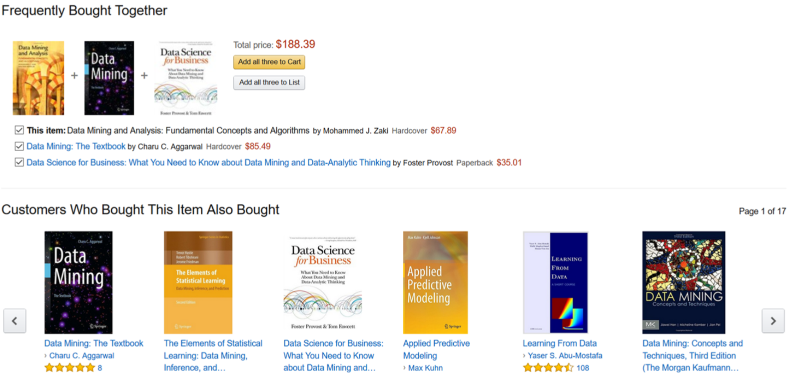
  i.e., for each $d \in D$, f maps to at most one $i \in I$.
- Notation:

$$(x, y) \in f \iff y = f(x) \iff f(x) = y \iff f : x \mapsto y$$

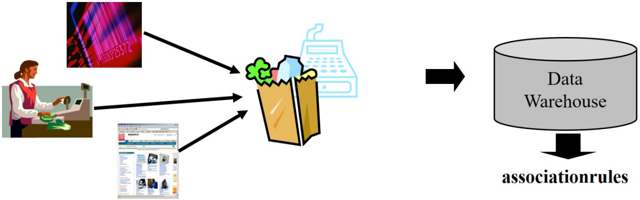
Outline

Motivation

on the webpage of some online bookseller:



Market Basket Analysis



application examples:

- layout of supermarket: optimize the arrangement of items bought together
- online seller: recommend related items
- cross-marketing, add-on sales, targeted attached mailings

read about the infamous example of beer and diapers:

`http://www.theregister.co.uk/2006/08/15/beer_diapers/`

Frequent Patterns

Transaction database:

```
T1:  {bread, butter, milk, sugar}
T2:  {butter, flour, milk, sugar}
T3:  {butter, eggs, milk, salt}
T4:  {eggs}
T5:  {butter, flour, milk, salt, sugar}
⋮
⋮
```

If we observe patterns, can we conclude on associations between items?

Associations

a,**b,c,d**,e
b,c,d
a,**b,c,d**
c,d,f
a,**b,c,d**,e
a,c,d
a,c,e,f
c,d,e,f
a,**b,c,d**,f
a,b,e,f

In 5 out of 10 (50%) cases,

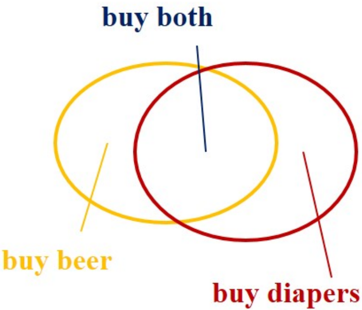
b,c,d

occur together.

In 5 cases we have **b,c**, and in all those 5 cases we also have **d**:

Rule with 100% confidence:

If ***b,c*** *are in the set, then also* ***d*** *is in the set.*



Outline

Association Rules


```
T1:  {bread, butter, milk, sugar}
T2:  {butter, flour, milk, sugar}
T3:  {butter, eggs, milk, salt}
T4:  {eggs}
T5:  {butter, flour, milk, salt, sugar}
```

items bread, butter, eggs, milk etc.

transaction a set of items

rule $L \Rightarrow R$, $L, R \subseteq$ items, $L \cap R = \emptyset$

- L *left-hand-side* or *antecedent*
- R *right-hand-side* or *consequent*

- {butter, flour} $\Rightarrow$ {milk}
- {sugar} $\Rightarrow$ {butter}

Definition: Frequent Itemsets

items: $I = \{i_1, \dots, i_m\}$ a set of literals (e.g., items in a shop)

itemset: $X \subseteq I$ (e.g., the items in a basket)

transaction: $T = (tid, X_{tid})$ designates a specific itemset

transaction database $\mathcal{D}$: a set of transactions

order: items in an itemset are ordered by some strict total order (e.g., alphabetical order of the literals), i.e.:
$X = (x_1, x_2, \dots, x_k) \Rightarrow x_1 < x_2 < \dots < x_k$

length of an itemset: number of elements contained in the itemset

k-itemset: an itemset of length k (e.g., T1 is a 4-itemset, T4 is a 1-itemset)

Definition: Cover, Support, Frequency

cover of an itemset: set of all transactions that contain the itemset: $\mathrm{cover}(X) = \{(tid, X_{tid}) | (tid, X_{tid}) \in \mathcal{D} \wedge X \subseteq X_{tid}\}$

support of an itemset: the support s of an itemset X ($s(X)$) is the number of transactions containing X (i.e., the size of the cover set): $s(X) = |\mathrm{cover}(X)|$

frequency of an itemset: the frequency of an itemset X is its support relative to the database size $f(X) = \frac{s(X)}{|\mathcal{D}|}$

frequent itemset: given some support threshold σ, an itemset X is frequent (w.r.t. σ) iff: $s(X) \geq \sigma$ or equivalently $f(X) \geq \frac{\sigma}{|\mathcal{D}|}$

Problem 1: Frequent Itemset Mining

Given:

- a set of items I
- a transaction database $\mathcal{D}$ over I
- a support threshold σ

Find all frequent itemsets in $\mathcal{D}$, i.e., $\{X | X \subseteq I \wedge s(X) \geq \sigma\}$

example: which itemsets are frequent with $\sigma = 3$ in $\mathcal{D}$:

```
T1:  {bread, butter, milk, sugar}
T2:  {butter, flour, milk, sugar}
T3:  {butter, eggs, milk, salt}
T4:  {eggs}
T5:  {butter, flour, milk, salt, sugar}
```

Definition: Association Rule

association rule: expresses an implication of the form $X \Rightarrow Y$, where X and Y are itemsets, $X \cap Y = \emptyset$

implication: describes a co-occurrence, not a causality

An association rule does not necessarily need to hold in all cases. We can describe its strength (or weakness), based on the observed cases:

support: The support of an association rule in $\mathcal{D}$ is the support of the union of its components:
$s(X \Rightarrow Y) = s(X \cup Y)$

frequency: Analogously, $f(X \Rightarrow Y) = f(X \cup Y)$

confidence: $\mathrm{conf}(X \Rightarrow Y) = \frac{s(X \cup Y)}{s(X)}$

Problem 2: Association Rule Mining

Given:

- a set of items I
- a transaction database $\mathcal{D}$ over I
- a support threshold σ and a confidence threshold c

Find all association rules $X \Rightarrow Y$ in $\mathcal{D}$ with a support of at least σ and a confidence of at least c, i.e.:

$$\{X \Rightarrow Y | s(X \Rightarrow Y) \geq \sigma \wedge \mathrm{conf}(X \Rightarrow Y) \geq c\}$$

```
T1:  {bread, butter, milk, sugar}
T2:  {butter, flour, milk, sugar}
T3:  {butter, eggs, milk, salt}
T4:  {eggs}
T5:  {butter, flour, milk, salt, sugar}
```

Problem 1 $\subset$ Problem 2

Problem 1 is part of Problem 2:

- Itemset A is frequent w.r.t. σ
- Given
  - $A = X \cup Y$
  - $X \cap Y = \emptyset$
  - $X = A \setminus Y$
  - $Y = A \setminus X$
- '$X \Rightarrow Y$' is frequent w.r.t. σ

Two-step approach:

1. find all frequent itemsets w.r.t. σ
2. generate rules with confidence $\geq c$ from each frequent itemset, where each rule is a binary partition of the itemset

Find Frequent Itemsets: Naïve Algorithm (Brute Force)

- Each possible itemset is a *candidate* for being a *frequent* itemset.
- We need to check the database to count if the itemset is actually frequent.
- Complexity roughly amounts to:
  number of candidates $\times$ number of transactions
- How many candidates do we have, given n items?

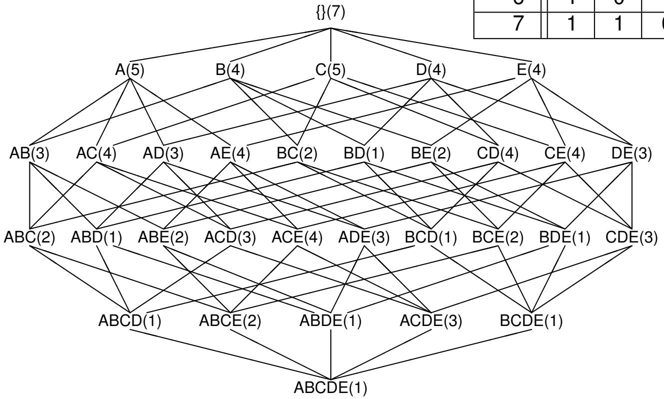
```
T1:  {bread, butter, milk, sugar}
T2:  {butter, flour, milk, sugar}
T3:  {butter, eggs, milk, salt}
T4:  {eggs}
T5:  {butter, flour, milk, salt, sugar}
```

SDU SYDDANSK UNIVERSITET

Outline

Lattice of Itemsets

TID	A	B	C	D	E
1	0	1	0	0	0
2	1	0	1	1	1
3	1	1	1	0	1
4	0	0	1	1	0
5	1	1	1	1	1
6	1	0	1	1	1
7	1	1	0	0	0

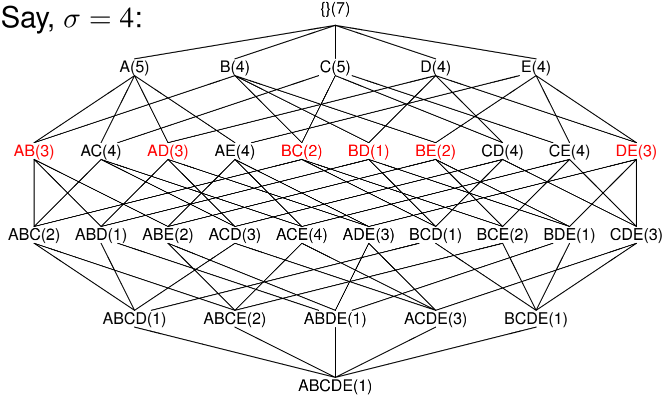


Monotonicity and Anti-Monotonicity

Observation:

- If X is frequent, all subsets $X' \subseteq X$ are frequent as well.
- If X is not frequent, neither any superset $X' \supseteq X$ can be frequent.

Say, $\sigma = 4$:

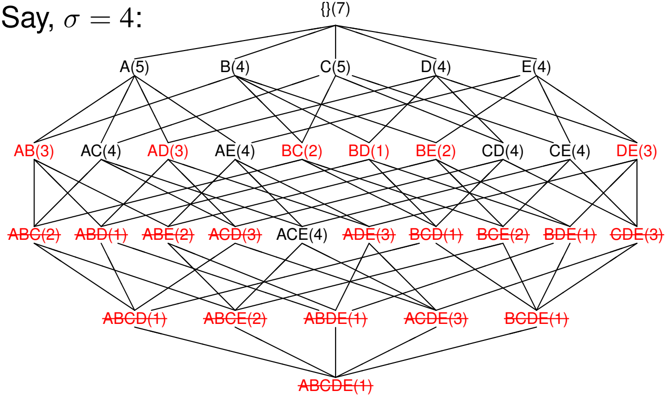


Monotonicity and Anti-Monotonicity

Observation:

- If X is frequent, all subsets $X' \subseteq X$ are frequent as well.
- If X is not frequent, neither any superset $X' \supseteq X$ can be frequent.

Say, $\sigma = 4$:

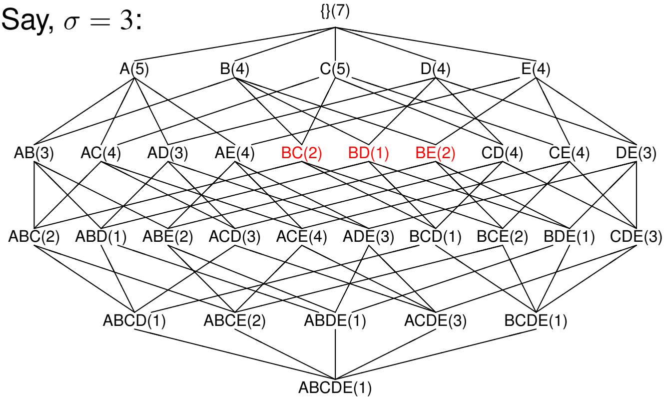


Monotonicity and Anti-Monotonicity

Observation:

- If X is frequent, all subsets $X' \subseteq X$ are frequent as well.
- If X is not frequent, neither any superset $X' \supseteq X$ can be frequent.

Say, $\sigma = 3$:

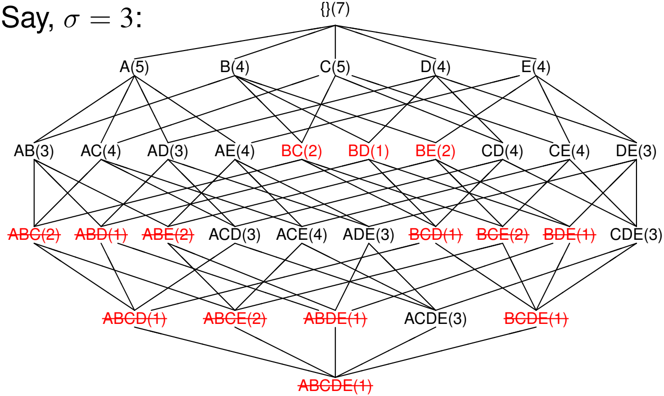


Monotonicity and Anti-Monotonicity

Observation:

- If X is frequent, all subsets $X' \subseteq X$ are frequent as well.
- If X is not frequent, neither any superset $X' \supseteq X$ can be frequent.

Say, $\sigma = 3$:



Positive and Negative Border

Due to the anti-monotonicity, we can summarize solutions by their border in the lattice:

- An itemset X belongs to the border (w.r.t. some σ), if:
  - $\forall Y \subset X : Y$ is frequent (w.r.t. σ)
  - $\forall Z \supset X : Z$ is not frequent (w.r.t. σ)
- positive border: X itself is frequent (also: “X is a maximal frequent itemset”)
- negative border: X itself is not frequent

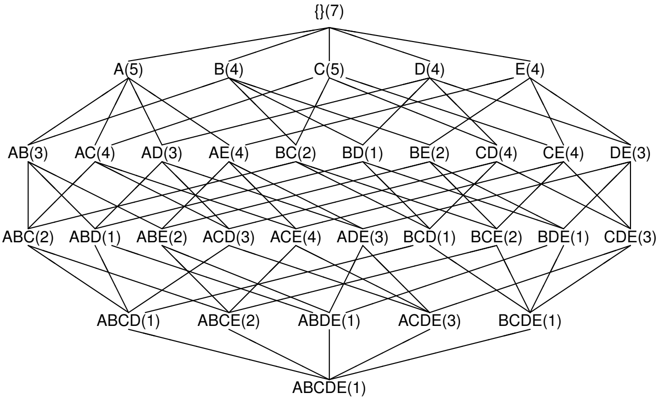
Maximal frequent itemsets can be used as a condensed representation of a solution, as all frequent itemsets can be derived from the maximal frequent itemsets.

Note that:

The anti-monotonicity property of support is also called downward-closure property.

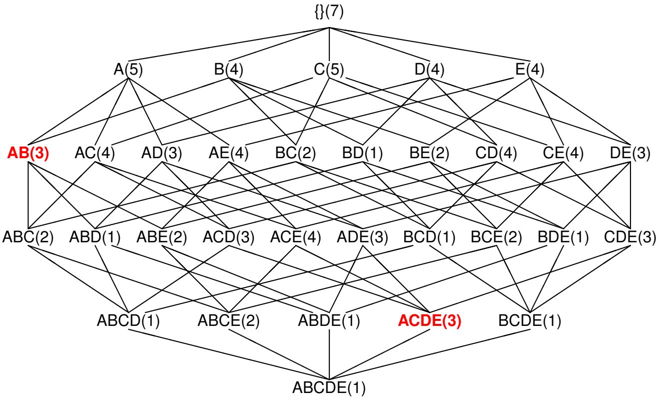
Maximal Frequent Itemsets

Example: find the positive border (maximal frequent itemsets) for $\sigma = 3$



Maximal Frequent Itemsets

Example: find the positive border (maximal frequent itemsets) for $\sigma = 3$

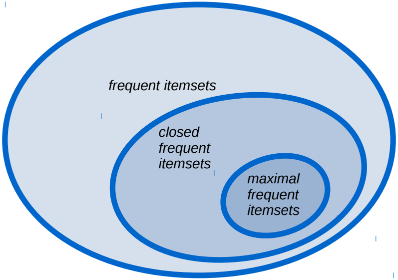


Closed (Frequent) Itemsets

closed itemset: An itemset X is *closed* if none of its immediate supersets has exactly the same support as X.

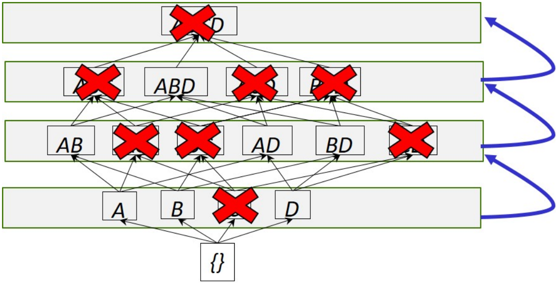
closed frequent itemset: An itemset is a closed frequent itemset (w.r.t. some σ) if it is a closed itemset and is frequent (w.r.t. σ).

Closed frequent itemsets represent a solution (all frequent itemsets) *and* their support.



Outline

Apriori Algorithm [Srikant and Agrawal, 1996]: Idea



1. find frequent 1-itemsets first, then 2-itemsets, 3-itemsets etc. (breadth-first search in the lattice)
2. for finding $(k+1)$-itemsets C_{k+1}: consider only those as candidates, where *all* k-itemsets $C_k \subset C_{k+1}$ are frequent
3. count frequency of all k-itemset candidates in a single database scan (hashing of the candidate itemsets)

Apriori Algorithm [Srikant and Agrawal, 1996]: Pseudo Code

C_k: k-itemset candidates, S_k: frequent k-itemsets (solution)

Algorithm 2.1 (Apriori [Srikant and Agrawal, 1996])

Apriori(I, $\mathcal{D}$, σ)
  $S_1 =$ {frequent 1-itemsets};
  $k = 2$;
  while $S_{k-1} \neq \emptyset$ do
    $C_k =$ AprioriGenerateCandidates(S_{k-1});
    for each transaction $T \in \mathcal{D}$ do
      $C_T = \{c \in C_k | c \subseteq T\}$;
      for each $c \in C_T$ do
        c.count++;
    $S_k = \{c \in C_k | c.count \geq \sigma\}$;
    k++;
  return $\cup_k S_k$;

Apriori Candidate Generation

Algorithm 2.2 (AprioriGenerateCandidates(S_{k-1}))

1. *join:*
   *two frequent $(k-1)$-itemsets $p, q \in S_{k-1}$ are joined if they are identical in the **first** (order!) $k-2$ items:*

$$\begin{array}{rl} p \in S_{k-1}\text{:} & (\underline{A, B,} C) \\ q \in S_{k-1}\text{:} & (\underline{A, B,} D) \\ \hline \Rightarrow & (A, B, C, D) \quad \in C_k \end{array}$$

2. *pruning:*
   remove all k-itemsets from C_k that contain any $(k-1)$-itemset $\notin S_{k-1}$

example: $S_3 = \{(1,2,3),(1,2,4),(1,3,4),(1,3,5),(2,3,4)\}$

1. join: $C_4 = \{(1,2,3,4),(1,3,4,5)\}$
2. pruning: remove $(1,3,4,5)$

result: $C_4 = \{(1,2,3,4)\}$

Outline

Apriori Example: Find Frequent Itemsets X with $f(X) \geq 0.3$

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	
B	1
C	
D	
E	1
F	
G	1
H	1
I	
J	
K	
L	

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	1
B	2
C	1
D	
E	2
F	
G	2
H	2
I	
J	
K	
L	

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	2
B	3
C	2
D	
E	3
F	1
G	2
H	3
I	
J	
K	
L	

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	2
B	4
C	3
D	1
E	4
F	2
G	3
H	4
I	
J	
K	
L	1

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	3
B	5
C	3
D	1
E	5
F	2
G	3
H	5
I	
J	
K	1
L	1

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	3
B	6
C	3
D	1
E	6
F	3
G	4
H	6
I	1
J	
K	2
L	1

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	4
B	7
C	3
D	2
E	6
F	3
G	5
H	7
I	1
J	
K	2
L	1

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	5
B	8
C	3
D	3
E	6
F	3
G	6
H	7
I	1
J	
K	2
L	1

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	5
B	9
C	3
D	4
E	6
F	4
G	7
H	7
I	1
J	
K	2
L	1

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	5
B	9
C	4
D	4
E	7
F	5
G	7
H	7
I	1
J	
K	2
L	1

SDU SYDDANSK UNIVERSITET

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C1

A	6
B	9
C	5
D	4
E	8
F	6
G	7
H	8
I	1
J	
K	2
L	1

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
→ 12	A B E G

C1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8
I	1
J	
K	2
L	1

C1 and S1

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8
I	~~1~~
J	~~0~~
K	~~2~~
L	~~1~~

$\sigma = 30\% \Leftrightarrow$ support ≥ 4

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB		CE	
AC		CF	
AD		CG	
AE		CH	
AF		DE	
AG		DF	
AH		DG	
BC		DH	
BD		EF	
BE		EG	
BF		EH	
BG		FG	
BH		FH	
CD		GH	

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB		CE	
AC		CF	
AD		CG	
AE		CH	
AF		DE	
AG		DF	
AH		DG	
BC		DH	
BD		EF	
BE	1	EG	1
BF		EH	1
BG	1	FG	
BH	1	FH	
CD		GH	1

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	1	CE	1
AC	1	CF	
AD		CG	1
AE	1	CH	1
AF		DE	
AG	1	DF	
AH	1	DG	
BC	1	DH	
BD		EF	
BE	2	EG	2
BF		EH	2
BG	2	FG	
BH	2	FH	
CD		GH	2

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	2	CE	2
AC	2	CF	1
AD		CG	1
AE	2	CH	2
AF	1	DE	
AG	1	DF	
AH	2	DG	
BC	2	DH	
BD		EF	1
BE	3	EG	2
BF	1	EH	3
BG	2	FG	
BH	3	FH	1
CD		GH	2

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	2	CE	3
AC	2	CF	2
AD		CG	2
AE	2	CH	3
AF	1	DE	1
AG	1	DF	1
AH	2	DG	1
BC	3	DH	1
BD	1	EF	2
BE	4	EG	3
BF	2	EH	4
BG	3	FG	1
BH	4	FH	2
CD	1	GH	3

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	3	CE	3
AC	2	CF	2
AD		CG	2
AE	3	CH	3
AF	1	DE	1
AG	1	DF	1
AH	3	DG	1
BC	3	DH	1
BD	1	EF	2
BE	5	EG	3
BF	2	EH	5
BG	3	FG	1
BH	5	FH	2
CD	1	GH	3

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	3	CE	3
AC	2	CF	2
AD		CG	2
AE	3	CH	3
AF	1	DE	1
AG	1	DF	1
AH	3	DG	1
BC	3	DH	1
BD	1	EF	3
BE	6	EG	4
BF	3	EH	6
BG	4	FG	2
BH	6	FH	3
CD	1	GH	4

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	4	CE	3
AC	2	CF	2
AD	1	CG	2
AE	3	CH	3
AF	1	DE	1
AG	2	DF	1
AH	4	DG	2
BC	3	DH	2
BD	2	EF	3
BE	6	EG	4
BF	3	EH	6
BG	5	FG	2
BH	7	FH	3
CD	1	GH	5

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	5	CE	3
AC	2	CF	2
AD	2	CG	2
AE	3	CH	3
AF	1	DE	1
AG	3	DF	1
AH	4	DG	3
BC	3	DH	2
BD	3	EF	3
BE	6	EG	4
BF	3	EH	6
BG	6	FG	2
BH	7	FH	3
CD	1	GH	5

SDU
SYDDANSK UNIVERSITET

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	5	CE	3
AC	2	CF	2
AD	2	CG	2
AE	3	CH	3
AF	1	DE	1
AG	3	DF	2
AH	4	DG	4
BC	3	DH	2
BD	4	EF	3
BE	6	EG	4
BF	4	EH	6
BG	7	FG	3
BH	7	FH	3
CD	1	GH	5

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	5	CE	4
AC	2	CF	3
AD	2	CG	2
AE	3	CH	3
AF	1	DE	1
AG	3	DF	2
AH	4	DG	4
BC	3	DH	2
BD	4	EF	4
BE	6	EG	4
BF	4	EH	6
BG	7	FG	3
BH	7	FH	3
CD	1	GH	5

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
→ 11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	5	CE	5
AC	3	CF	4
AD	2	CG	2
AE	4	CH	4
AF	2	DE	1
AG	3	DF	2
AH	5	DG	4
BC	3	DH	2
BD	4	EF	5
BE	6	EG	4
BF	4	EH	7
BG	7	FG	3
BH	7	FH	4
CD	1	GH	5

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
→ 12	A B E G

S1

A
B
C
D
E
F
G
H

C2

AB	6	CE	5
AC	3	CF	4
AD	2	CG	2
AE	5	CH	4
AF	2	DE	1
AG	4	DF	2
AH	5	DG	4
BC	3	DH	2
BD	4	EF	5
BE	7	EG	5
BF	4	EH	7
BG	8	FG	3
BH	7	FH	4
CD	1	GH	5

C2 and S2

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S1

A
B
C
D
E
F
G
H

S2

AB	6	CE	5
AC	~~3~~	CF	4
AD	~~2~~	CG	~~2~~
AE	5	CH	4
AF	~~2~~	DE	~~1~~
AG	4	DF	~~2~~
AH	5	DG	4
BC	~~3~~	DH	~~2~~
BD	4	EF	5
BE	7	EG	5
BF	4	EH	7
BG	8	FG	~~3~~
BH	7	FH	4
CD	~~1~~	GH	5

S2 to C3

S2

AB	CE
AE	CF
AG	CH
AH	DG
BD	EF
BE	EG
BF	EH
BG	FH
BH	GH

→

C3

S2 to C3

S2

AB	CE
AE	CF
AG	CH
AH	DG
BD	EF
BE	EG
BF	EH
BG	FH
BH	GH

$\longrightarrow$

C3

ABE	
ABG	
ABH	
AEG	
AEH	
AGH	

S2 to C3

S2

AB	CE
AE	CF
AG	CH
AH	DG
BD	EF
BE	EG
BF	EH
BG	FH
BH	GH

⟶

C3

ABE	BEG
ABG	BEH
ABH	BFG
AEG	BFH
AEH	BGH
AGH	
BDE	
BDF	
BDG	
BDH	
BEF	

S2 to C3

S2

AB	CE
AE	CF
AG	CH
AH	DG
BD	EF
BE	EG
BF	EH
BG	FH
BH	GH

⟶

C3

ABE	BEG
ABG	BEH
ABH	BFG
AEG	BFH
AEH	BGH
AGH	CEF
BDE	CEH
BDF	CFH
BDG	
BDH	
BEF	

S2 to C3

S2

AB	CE
AE	CF
AG	CH
AH	DG
BD	EF
BE	EG
BF	EH
BG	FH
BH	GH

⟶

C3

ABE	BEG
ABG	BEH
ABH	BFG
AEG	BFH
AEH	BGH
AGH	CEF
BDE	CEH
BDF	CFH
BDG	EFG
BDH	EFH
BEF	EGH

SDU
SYDDANSK UNIVERSITET

S2 to C3

S2

AB	CE
AE	CF
AG	CH
AH	DG
BD	EF
BE	EG
BF	EH
BG	FH
BH	GH

→

C3

ABE	BEG
ABG	BEH
ABH	~~BFG~~
AEG	BFH
AEH	BGH
AGH	CEF
~~BDE~~	CEH
~~BDF~~	CFH
BDG	~~EFG~~
~~BDH~~	EFH
BEF	EGH

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE		BEH	
ABG		BFH	
ABH		BGH	
AEG		CEF	
AEH		CEH	
AGH		CFH	
BDG		EFH	
BEF		EGH	
BEG			

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE		BEH	1
ABG		BFH	
ABH		BGH	1
AEG		CEF	
AEH		CEH	
AGH		CFH	
BDG		EFH	
BEF		EGH	1
BEG	1		

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	1	BEH	2
ABG	1	BFH	
ABH	1	BGH	2
AEG	1	CEF	
AEH	1	CEH	1
AGH	1	CFH	
BDG		EFH	
BEF		EGH	2
BEG	2		

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	2	BEH	3
ABG	1	BFH	1
ABH	2	BGH	2
AEG	1	CEF	1
AEH	2	CEH	2
AGH	1	CFH	1
BDG		EFH	1
BEF	1	EGH	2
BEG	2		

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	2	BEH	4
ABG	1	BFH	2
ABH	2	BGH	3
AEG	1	CEF	2
AEH	2	CEH	3
AGH	1	CFH	2
BDG	1	EFH	2
BEF	2	EGH	3
BEG	3		

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	3	BEH	5
ABG	1	BFH	2
ABH	3	BGH	3
AEG	1	CEF	2
AEH	3	CEH	3
AGH	1	CFH	2
BDG	1	EFH	2
BEF	2	EGH	3
BEG	3		

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	3	BEH	6
ABG	1	BFH	3
ABH	3	BGH	4
AEG	1	CEF	2
AEH	3	CEH	3
AGH	1	CFH	2
BDG	1	EFH	3
BEF	3	EGH	4
BEG	4		

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	3	BEH	6
ABG	2	BFH	3
ABH	4	BGH	5
AEG	1	CEF	2
AEH	3	CEH	3
AGH	2	CFH	2
BDG	2	EFH	3
BEF	3	EGH	4
BEG	4		

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	3	BEH	6
ABG	3	BFH	3
ABH	4	BGH	5
AEG	1	CEF	2
AEH	3	CEH	3
AGH	2	CFH	2
BDG	3	EFH	3
BEF	3	EGH	4
BEG	4		

SDU SYDDANSK UNIVERSITET

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	3	BEH	6
ABG	3	BFH	3
ABH	4	BGH	5
AEG	1	CEF	2
AEH	3	CEH	3
AGH	2	CFH	2
BDG	4	EFH	3
BEF	3	EGH	4
BEG	4		

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	3	BEH	6
ABG	3	BFH	3
ABH	4	BGH	5
AEG	1	CEF	3
AEH	3	CEH	3
AGH	2	CFH	2
BDG	4	EFH	3
BEF	3	EGH	4
BEG	4		

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	3	BEH	6
ABG	3	BFH	3
ABH	4	BGH	5
AEG	1	CEF	4
AEH	4	CEH	4
AGH	2	CFH	3
BDG	4	EFH	4
BEF	3	EGH	4
BEG	4		

C3 and S3

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C3

ABE	4	BEH	6
ABG	4	BFH	3
ABH	4	BGH	5
AEG	2	CEF	4
AEH	4	CEH	4
AGH	2	CFH	3
BDG	4	EFH	4
BEF	3	EGH	4
BEG	5		

C3 and S3

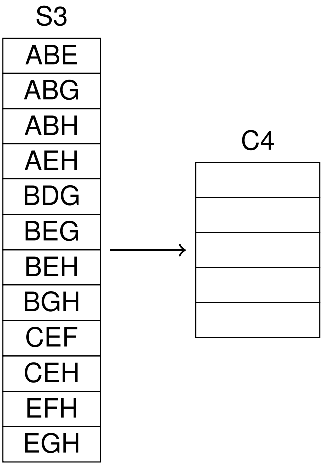
Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S3

ABE	4	BEH	6
ABG	4	BFH	~~3~~
ABH	4	BGH	5
AEG	~~2~~	CEF	4
AEH	4	CEH	4
AGH	~~2~~	CFH	~~3~~
BDG	4	EFH	4
BEF	~~3~~	EGH	4
BEG	5		

S3 to C4



S3 to C4

S3

ABE
ABG
ABH
AEH
BDG
BEG
BEH
BGH
CEF
CEH
EFH
EGH

→

C4

ABEG
ABEH
ABGH

S3 to C4

S3
ABE
ABG
ABH
AEH
BDG
BEG
BEH
BGH
CEF
CEH
EFH
EGH

→

C4
ABEG
ABEH
ABGH
BEGH

S3 to C4

DM566

Melih Kandemir

Introduction
Freq. Pattern Mining
Sets and Relations
Item Sets
Definitions and Problem-Statements
Monotonicity-Property of Frequency
The Apriori Algorithm
Example
Efficiency
Association Rule Mining
Summary
Feature Spaces
Clustering – Basics
Classification – Basics
Bayesian Learning
Learning with Distributions
References

S3
ABE
ABG
ABH
AEH
BDG
BEG
BEH
BGH
CEF
CEH
EFH
EGH

→

C4
ABEG
ABEH
ABGH
BEGH
CEFH

S3 to C4

S3
ABE
ABG
ABH
AEH
BDG
BEG
BEH
BGH
CEF
CEH
EFH
EGH

→

C4
~~ABEG~~
ABEH
~~ABGH~~
BEGH
~~CEFH~~

AEG, AGH and CFH not *frequent*!

C4 and S4

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C4

ABEH	
BEGH	

C4 and S4

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C4

ABEH	
BEGH	1

C4 and S4

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C4

ABEH	1
BEGH	2

C4 and S4

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C4

ABEH	2
BEGH	2

C4 and S4

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C4

ABEH	2
BEGH	3

C4 and S4

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C4

ABEH	3
BEGH	3

C4 and S4

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C4

ABEH	3
BEGH	4

C4 and S4

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

C4

ABEH	3
BEGH	4

C4 and S4

Database

1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

S4

ABEH	~~3~~
BEGH	4

Only one frequent 4-Itemset remaining.

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI
CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI
CFI only

Maximal Frequent Itemsets and Closed Frequent Itemsets

S1

A	7
B	10
C	5
D	4
E	9
F	6
G	8
H	8

S2

AB	6
AE	5
AG	4
AH	5
BD	4
BE	7
BF	4
BG	8
BH	7
CE	5
CF	4
CH	4
DG	4
EF	5
EG	5
EH	7
FH	4
GH	5

S3

ABE	4
ABG	4
ABH	4
AEH	4
BDG	4
BEG	5
BEH	6
BGH	5
CEF	4
CEH	4
EFH	4
EGH	4

S4

BEGH	4

MFI and CFI

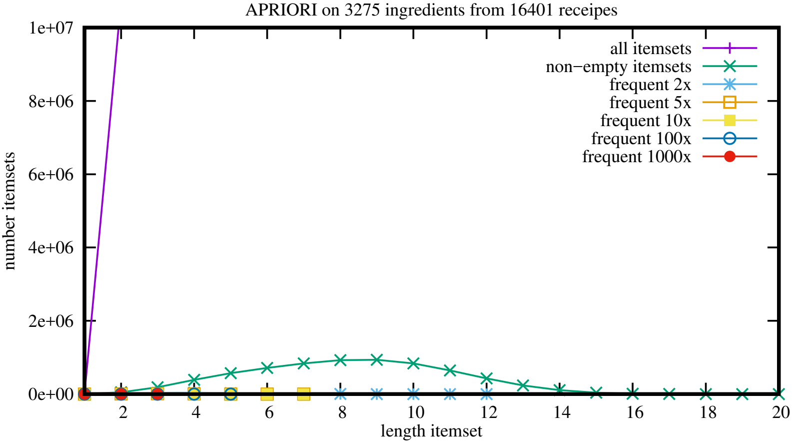
CFI only

Outline

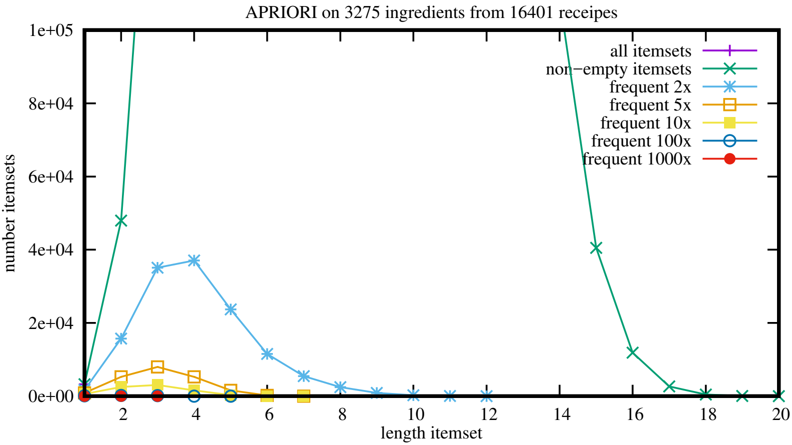
In Theory…

- naïve algorithm: count frequency of all k-itemsets, for all $\binom{I}{k}$ k-itemsets, for all k
- number of possible itemsets $0 \leq k \leq |I|$?
- Apriori: one database scan for all frequent k-itemset *candidates* of a given k
- reduction of number of candidates by the anti-monotonicity principle of frequency: generate only candidates that have a chance to be frequent (join of frequent $(k-1)$-itemsets and pruning)

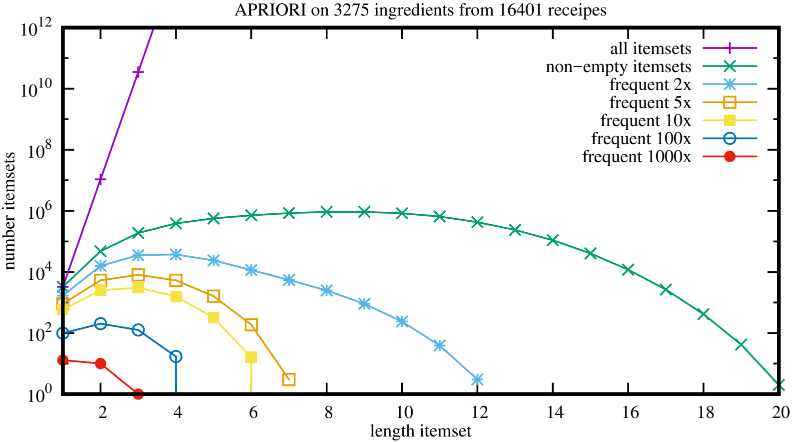
Experiments



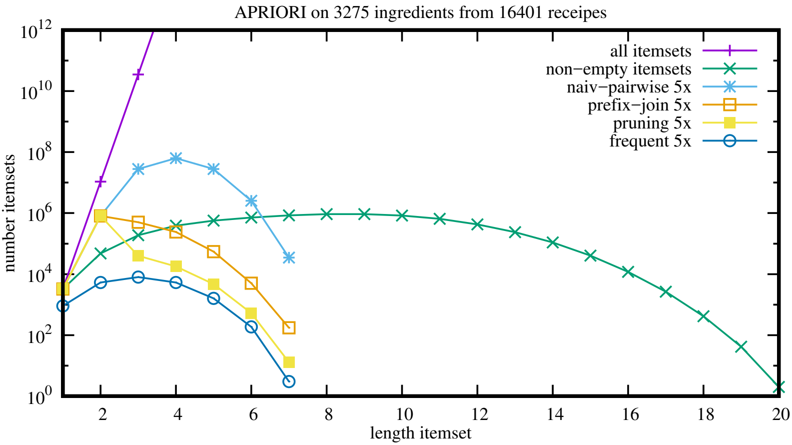
Experiments



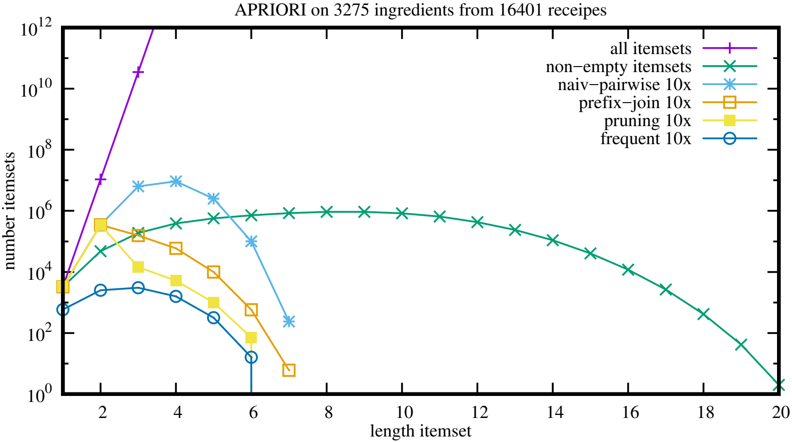
Experiments



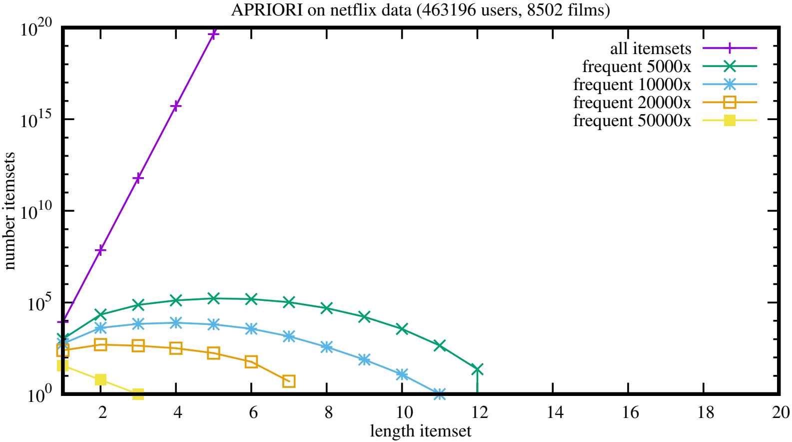
Experiments



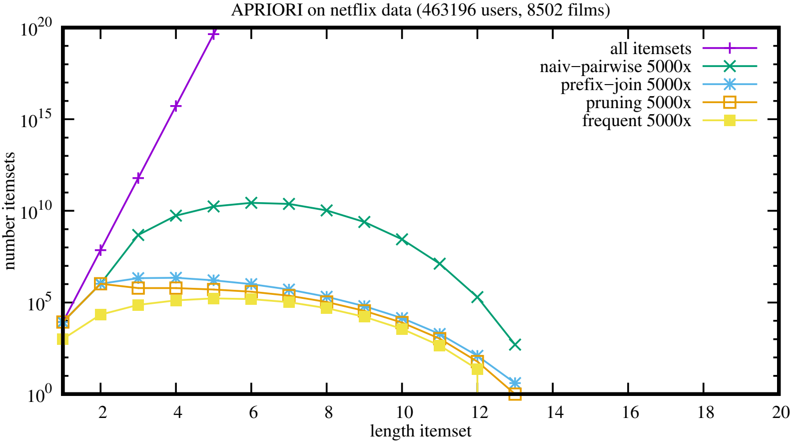
Experiments



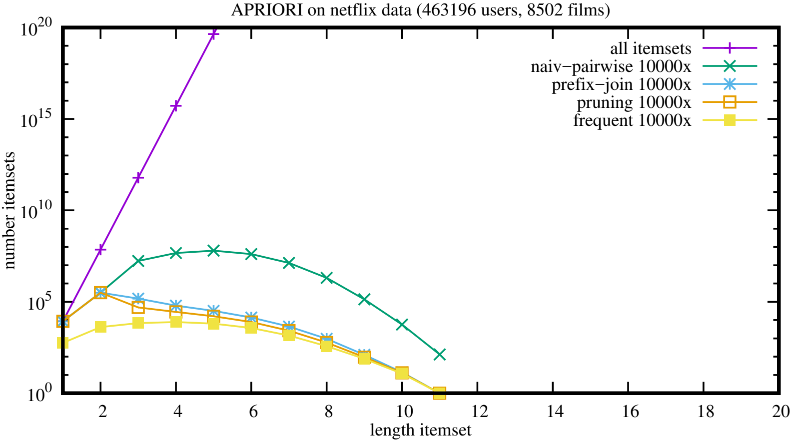
Experiments



Experiments



Experiments



Outline

Definition: Association Rule

association rule: expresses an implication of the form $X \Rightarrow Y$, where X and Y are itemsets, $X \cap Y = \emptyset$

implication: describes a co-occurrence, not a causality

An association rule does not necessarily need to hold in all cases. We can describe its strength (or weakness), based on the observed cases:

support: The support of an association rule in $\mathcal{D}$ is the support of the union of its components:
$s(X \Rightarrow Y) = s(X \cup Y)$

frequency: Analogously, $f(X \Rightarrow Y) = f(X \cup Y)$

confidence: $\mathrm{conf}(X \Rightarrow Y) = \frac{s(X \cup Y)}{s(X)}$

Problem 2: Association Rule Mining

Given:

- a set of items I
- a transaction database $\mathcal{D}$ over I
- a support threshold σ and a confidence threshold c

Find all association rules $X \Rightarrow Y$ in $\mathcal{D}$ with a support of at least σ and a confidence of at least c, i.e.:

$$\{X \Rightarrow Y | s(X \Rightarrow Y) \geq \sigma \wedge \text{conf}(X \Rightarrow Y) \geq c\}$$

```
T1:  {bread, butter, milk, sugar}
T2:  {butter, flour, milk, sugar}
T3:  {butter, eggs, milk, salt}
T4:  {eggs}
T5:  {butter, flour, milk, salt, sugar}
```

Find Association Rules

This is part of the Apriori algorithm [Srikant and Agrawal, 1996].

for frequent itemset X:

- for each $Y \subset X$, $Y \neq \emptyset$, build the rule $Y \Rightarrow (X \setminus Y)$
- $\mathrm{conf}(Y \Rightarrow (X \setminus Y)) = \frac{s(X)}{s(Y)}$
- delete rules with confidence below a given threshold c

Note that:

For all involved itemsets (X, Y, $(X \setminus Y)$), we have the support from the solution of Problem 1 (stored or reconstructable from closed frequent itemsets). Thus we don't need a single database scan here.

Monotonicity of Confidence

Theorem 2.1

Given:

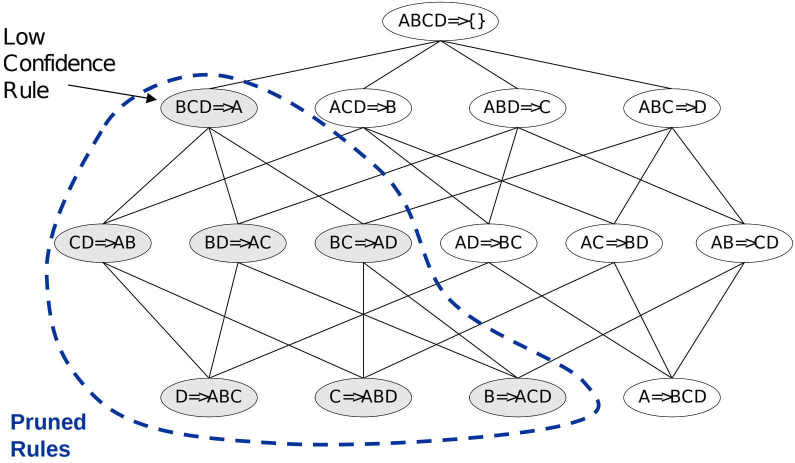
- *itemset* X
- $Y \subset X, Y \neq \emptyset$

If $\mathrm{conf}(Y \Rightarrow (X \setminus Y)) < c$, *then* $\forall Y' \subset Y$:

$$\mathrm{conf}(Y' \Rightarrow (X \setminus Y')) < c.$$

This property allows the construction of all association rules satisfying some confidence threshold from all frequent itemsets with a procedure similar to the Apriori construction of frequent itemsets, but without database scan. [Srikant and Agrawal, 1996]

Pruning of Association Rules



Adapted from Tan et al. [2006], Fig. 6.15.

Example: Association Rules

Association rules for BEGH (see slide 88), confidence $\geq 60\%$, support 4:

Antecedent	Consequent	Support	Confidence	Rule
BEGH	∅	4	1.000	
BEG	H	5	$4/5 =$ **0.800**	BEG ⇒ H
BE	GH	7	$4/7 \approx$ 0.571	-
BG	EH	8	$4/8 =$ 0.500	-
EG	BH	5	$4/5 =$ **0.800**	EG ⇒ BH
BEH	G	6	$4/6 \approx$ **0.667**	BEH ⇒ G
BH	EG	7	$4/7 \approx$ 0.571	-
EH	BG	7	$4/7 \approx$ 0.571	-
BGH	E	5	$4/5 =$ **0.800**	BGH ⇒ E
GH	BE	5	$4/5 =$ **0.800**	GH ⇒ BE
EGH	B	4	$4/4 =$ **1.000**	EGH ⇒ B

Interpretation of Support and Confidence

support: measures the frequency of the item set

- rules with very low support may occur simply by chance
- rules with low support are uninteresting from a business perspective

confidence: measures the reliability of the rule

- $X \Rightarrow Y$ – the higher the confidence, the more likely Y is present in transactions that contain X
- estimate of the conditional probability of Y given X

Limitations of Support and Confidence as Measures

	coffee	no coffee	
tea	150	50	200
no tea	650	150	800
	800	200	1000

$\{tea\} \Rightarrow \{coffee\}$

support?

confidence?

$\{\} \Rightarrow \{coffee\}$

support?

confidence?

Conclusion?

(Discussed by Tan et al. [2006], page 372f., example 6.3.)

Interestingness

Other measures to assess the interestingness of a rule include:

Lift: $\mathit{Lift}(A \Rightarrow B) = \frac{\mathrm{conf}(A \Rightarrow B)}{f(B)}$

Jaccard: $\mathit{Jaccard}(A \Rightarrow B) = \frac{s(A \cup B)}{s(A) + s(B) - s(A \cup B)}$

conviction: $\mathit{conviction}(A \Rightarrow B) = \frac{1 - f(B)}{1 - \mathrm{conf}(A \Rightarrow B)}$

Recommended Reading:

Advanced reading:

- *Vreeken and Tatti [2014]*

Outline

Outlook: Other Algorithms

- Frequent pattern mining is a large research field, for an overview see the collection of topics edited by Aggarwal and Han [2014].
- Another important algorithm for frequent pattern mining is FP-Growth [Han et al., 2000].
- Frequent patterns have been defined in other application scenarios such as, e.g., graphs, spatiotemporal data, sequential data (for example protein sequences, as in the work of Birzele and Kramer [2006]).
- The principle of anti-monotonicity for pruning has been applied in many other application areas (e.g., in subspace clustering [Zimek et al., 2014]).

Summary

You learned in this section:

- *mathematical basics on sets, relations, and orders*
- *the problem of frequent pattern mining*
- *the problem of association rule mining*
- *the "Apriori principle" (the anti-monotonicity property of frequent itemsets)*
- *the Apriori algorithm for mining of frequent itemsets and association rules*
- *evaluation aspects, pros and cons of support and confidence*

Outline

Recommended Reading

Recommended Reading:

- *Zaki and Meira Jr. [2014], Chapters 2+3*
- *Tan et al. [2006], Chapters 2+3*
- *Tan et al. [2020], Chapter 2*
- *Han et al. [2011], Chapter 2*

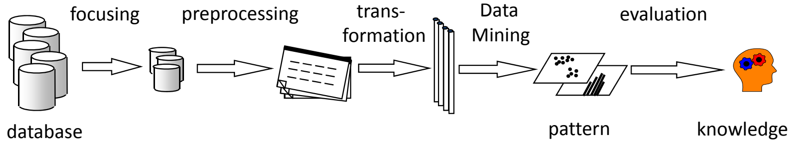
Outline

DM566

Melih Kandemir

The KDD process model

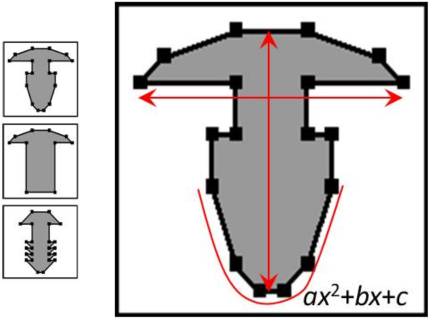
KDD process model (cf. Fayyad et al. [1996])



- *focusing*
  - get the data
  - organize data (file/database)
  - select relevant data
- *preprocessing*
  - integrate heterogeneous data
  - check for completeness
  - check for consistency
- *transformation*
  - discretize numeric attributes
  - infer new attributes
  - select relevant attributes
- *data mining*
  - generate patterns or modells
- *evaluation*
  - assess “interestingness” for the user
  - validate models statistically

Deriving Features from Complex Objects

Example: CAD-objects



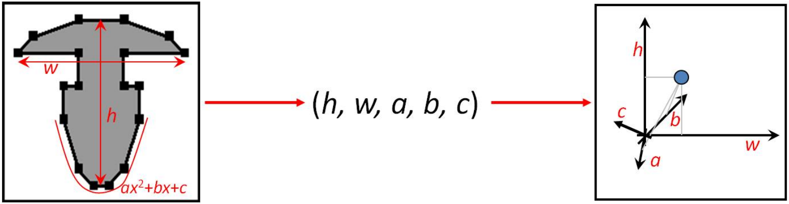
possible features:

- width
- height
- curvature parameters a, b, c

Deriving Features from Complex Objects

Transformation:

object space $\longrightarrow$ feature space



- features are combined to feature vectors
- often high-dimensional feature spaces (here only 5-dim.)
- statistical context: features are called ‘variables’

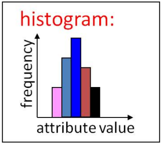
Scale Characteristics of Features

- nominal (categories):
  - It is possible to tell whether two values are equal or not – but no direction (better, more,. . .) or meaningful distance.
  - Examples: sex, eye color, healthy/sick, amino acids
- ordinal
  - there is an order relation (e.g., better/worse), but no uniform distance
  - Examples: grade, quality label, age class (e.g., 20-29, 30-39,. . .), color (?)
- metric
  - differences and relations between values are meaningful, values can be discrete or continuous
  - Examples: weight, selling counts, age

Aggregations of Features

For a feature X, we have a sample $x_1, x_2, \ldots, x_n$.

- absolute frequency: For each value a, we have $f(a)$ as the number of occurrences in the sample.
- relative frequency: $p(a) = f(a)/n$
- mode: the value with largest frequency



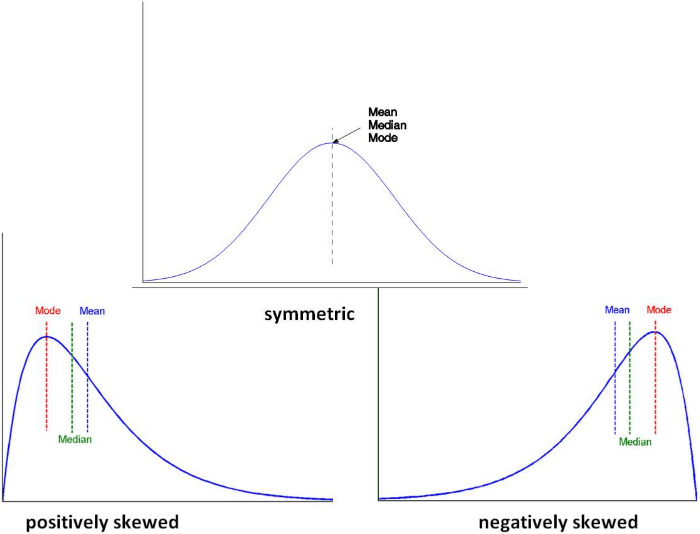
for at least ordinal features:

- median: the central element in the sample ordered by value

for metric features:

- arithmetic mean: $\mu = \bar{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$

Skewness


Mean
Median
Mode
symmetric
Mode
Mean
Median
positively skewed
Mean
Mode
Median
negatively skewed

Aggregations of Feature Vectors

For a given sample $\mathcal{D}$, we have:

- centroid: $\mu_{\mathcal{D}} = \frac{1}{|\mathcal{D}|} \cdot \sum_{o \in \mathcal{D}} o$

  In a general metric space (that is, not a vector space), where we only have pairwise distances, it might not be possible to compute a centroid.

- medoid: $\arg\min_{o \in \mathcal{D}} \frac{1}{I\mathcal{D}I} \sum_{o' \in \mathcal{D}} \mathrm{dist}(o, o')$

  In a general metric space, the medoid is the object with the smallest average distance to all other objects.

SDU SYDDANSK UNIVERSITET

Outline

Feature Space and Distance Function

- A feature space is a domain with a distance function:

$$F = (\mathrm{dom}, \mathrm{dist})$$

- dom is a sorted set of features
- $\mathrm{dist} : \mathrm{dom} \times \mathrm{dom} \to \mathbb{R}_0^+$ is a total (distance) function with the following properties:
  - strictness:

$$\forall p, q \in \mathrm{dom}, p \neq q : \mathrm{dist}(p, q) > 0$$

  - reflexivity:

$$\forall o \in \mathrm{dom} : \mathrm{dist}(o, o) = 0$$

  - symmetry:

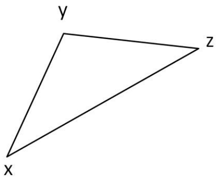
$$\forall p, q \in \mathrm{dom} : \mathrm{dist}(p, q) = \mathrm{dist}(q, p)$$

Metric Space

- $M = (\mathrm{dom}, \mathrm{dist})$ is a metric space, if the following properties are given:
  - M is a feature space
  - the triangle inequality holds:

$$\forall o, p, q \in \mathrm{dom} : \mathrm{dist}(o, p) \leq \mathrm{dist}(o, q) + \mathrm{dist}(q, p)$$

- most common example: Euclidean *vector* space:
  - $\mathrm{dom} = \mathbb{R}^d$ (d: number of features – dimensionality)
  - $\mathrm{dist} = (x, y) \mapsto \|x - y\|_2$

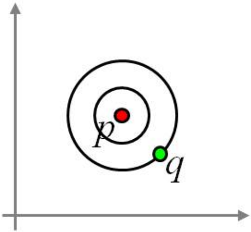


Spaces and Distance Functions

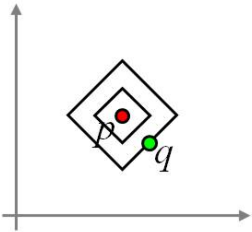
Common distance measure for (Euclidean) feature vectors: $L_{\mathcal{P}}$-norm

$$\mathrm{dist}_{\mathcal{P}}(p,q) = \left(|p_1 - q_1|^{\mathcal{P}} + |p_2 - q_2|^{\mathcal{P}} + \ldots + |p_n - q_n|^{\mathcal{P}}\right)^{\frac{1}{\mathcal{P}}}$$

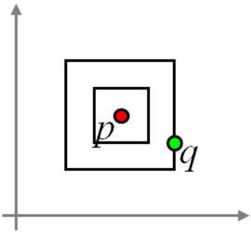
Euclidean norm (L_2):



Manhattan norm (L_1):



Maximum norm (L_∞, also: $L_{\max}$, supremum dist., Chebyshev dist.)



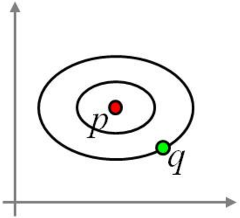
Spaces and Distance Functions

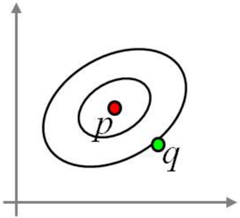
weighted Euclidean norm:

$$\mathrm{dist}(p,q) = \left(w_1|p_1 - q_1|^2 + w_2|p_2 - q_2|^2 + \ldots + w_n|p_n - q_n|^2\right)^{\frac{1}{2}}$$

quadratic form*:

$$\mathrm{dist}(p,q) = \left((p-q)M(p-q)^{\mathsf{T}}\right)^{\frac{1}{2}}$$





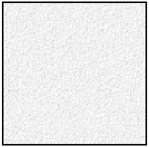
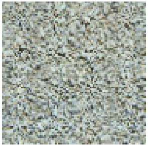
*note that we assume vectors to be row vectors here

Outline

Categories of Feature Descriptors for Images

- distribution of colors
- texture
- shapes (contoures)

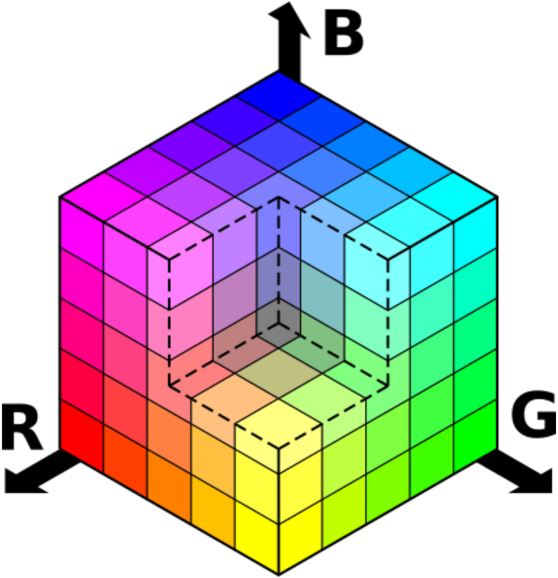


Color Histogram

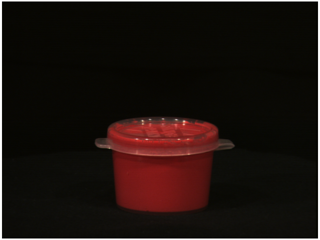
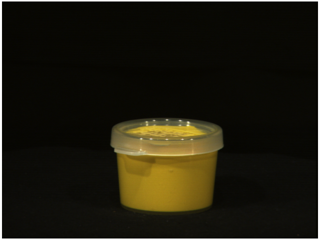
- a histogram represents the distribution of colors over the pixels of an image
- definition of an color histogram:
  - choose a color space (RGB, HSV, HLS, . . .)
  - choose number of representants (sample points) in the color space
  - possibly normalization (to account for different image sizes)

Color Space Example: RGB cube



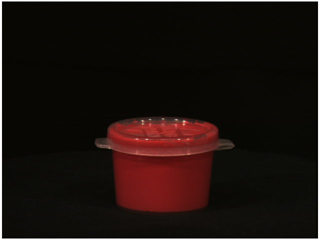
Impact of Number of Representants

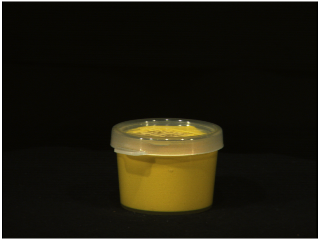
original images in full RGB space $(256^3 = 16,777,216)$


Impact of Number of Representants

original images in full RGB space ($256^3 = 16,777,216$)





Impact of Number of Representants

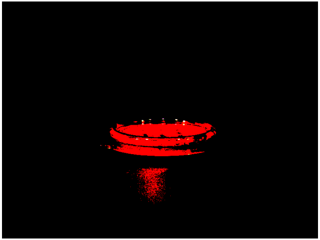
2^3

3^3

4^3

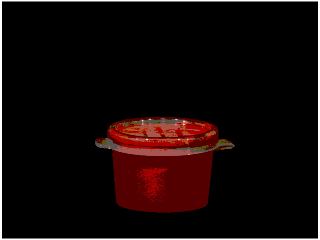
16^3

Impact of Number of Representants

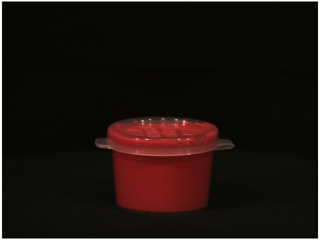


2^3

3^3

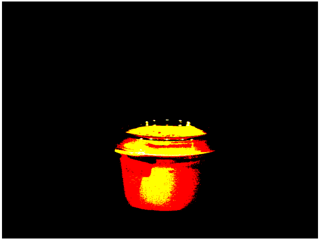


4^3

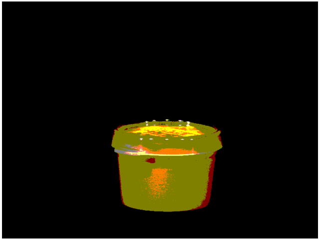


16^3

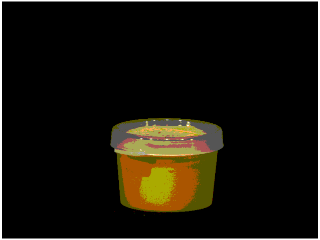
Impact of Number of Representants



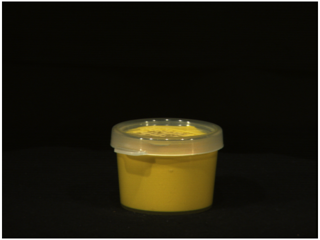
2^3



3^3



4^3



16^3

Impact of Number of Representants

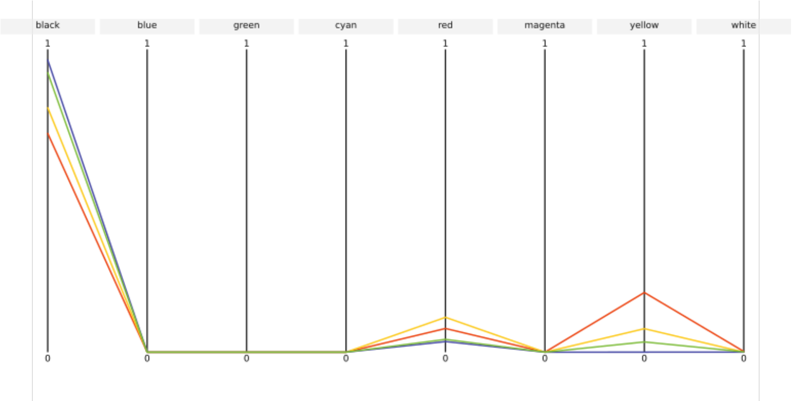
2^3

3^3

4^3

16^3

Impact of Number of Representants

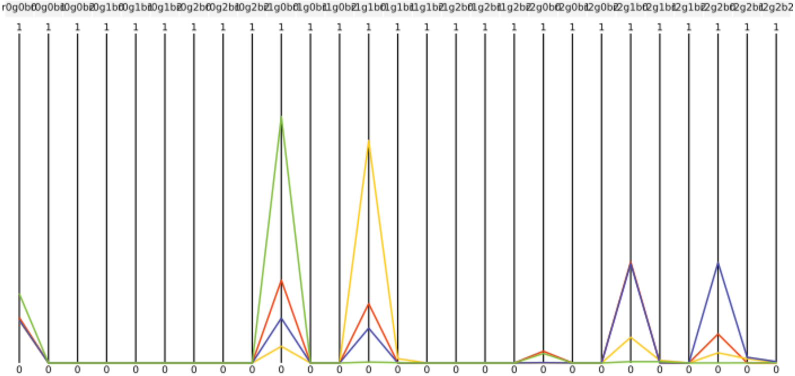


The histogram for each image is essentially a visualization of a vector:

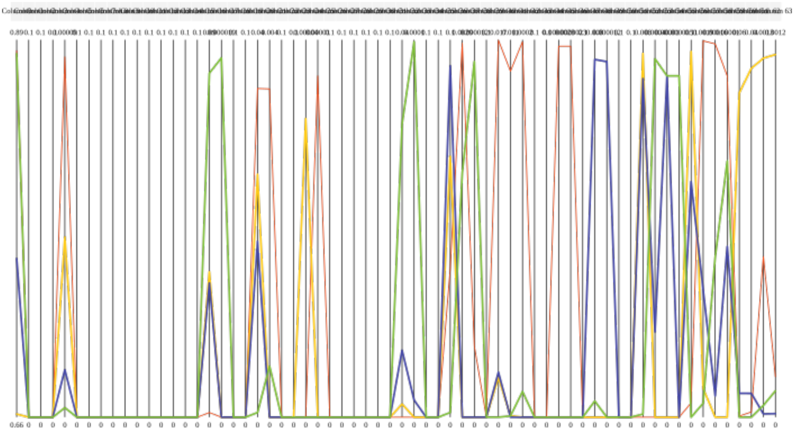
$(0.77, 0, 0, 0, 0.08, 0, 0.15, 0)$ $(0.8, 0, 0, 0, 0.11, 0, 0.09, 0)$

$(0.9, 0, 0, 0, 0.05, 0, 0.05, 0)$ $(0.955, 0, 0, 0, 0.045, 0, 0, 0)$

Impact of Number of Representants



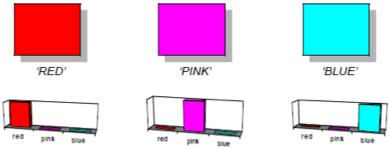
Impact of Number of Representants



Distances for Color Histograms

Euclidean distance for images P and Q using the color histograms h_P and h_Q:

$$\mathrm{dist}(P,Q) = \sqrt{(h_P - h_Q) \cdot (h_P - h_Q)^{\mathsf{T}}}$$



$$\mathrm{dist}(\text{RED}, \text{PINK}) = \sqrt{2}$$
$$\mathrm{dist}(\text{RED}, \text{BLUE}) = \sqrt{2}$$
$$\mathrm{dist}(\text{BLUE}, \text{PINK}) = \sqrt{2}$$

A ‘psychologic’ distance would consider that red is (in our perception) more similar to pink than to blue.

Distances for Color Histograms

Quadratic form with 'psychological' similarity matrix

$$A = \begin{bmatrix} 1 & a_{12} & \dots & \\ a_{21} & 1 & \dots & \\ \vdots & & \ddots & \vdots \\ & & \dots & 1 \end{bmatrix}$$ where a_{ij} ($\stackrel{?}{=} a_{ji}$) describe the subjective similarity of the features i and j in the color histogram:

$$\mathrm{dist}_A(P, Q) = \sqrt{(h_P - h_Q) \cdot A \cdot (h_P - h_Q)^\mathsf{T}}$$

$$A' = \begin{bmatrix} 1 & 0.9 & 0 \\ 0.9 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{dist}(\text{RED}, \text{PINK}) = \sqrt{0.2}$$
$$\mathrm{dist}(\text{RED}, \text{BLUE}) = \sqrt{2}$$
$$\mathrm{dist}(\text{BLUE}, \text{PINK}) = \sqrt{2}$$

Outline

Texts as “Bag of Words”

- texts can be seen as sets of words or vectors of terms
- a term can be
  - a single word
  - a phrase or part of a sentence
- typical feature extraction from texts: transform a text (document) into a vector of term frequencies:

$$d \mapsto (f_{t_1 d}, f_{t_2 d}, \ldots, f_{t_n d})$$

where $f_{t_i d}$ is the frequency of term t_i in document d.

The aim of machine learning is to build computer systems that can adapt to their environments and learn from experience. Learning techniques and methods from this field are successfully applied to a variety of learning tasks in a broad range of areas, including, for example, spam recognition, text classification, gene discovery, financial forecasting.

$$\mapsto \begin{pmatrix} \vdots \\ \text{machine: } 1 \\ \text{learn: } 4 \\ \text{to: } 3 \\ \vdots \end{pmatrix}$$

Problems with Term Frequencies

1. Many words are totally pointless for distinguishing texts (the, a, it, that, ...).
2. The same word can appear differently (learn, learning; go, went).
3. The feature space typically becomes very high dimensional (often $n > 10000$).
4. Not all terms are equally useful.
5. Most words from a dictionary don't occur at all in any of the compared documents ($f = 0$).

additional linguistic problems:

- different words with the same (or closely similar) meaning (“buy”, “purchase”)
- words with very different meaning (“mouse”)

Problem 1

- Many words are totally pointless for distinguishing texts (the, a, it, that, . . .).
- Solution: eliminate these words before computing the term frequency vector.
- Such words are called stopwords, list of stopwords for many languages exist.

Problem 2

- The same word can appear differently (learn, learning; go, went).
- Solution: stemming. Any word is mapped to its stem (base or root form).
- For English texts, algorithmic stemming is possible (Porter's stemming algorithm, see `http://tartarus.org/~martin/PorterStemmer/index.html`).
- For other languages, dictionaries are required.

Problem 3

- The feature space typically becomes very high dimensional (often $d > 10000$).
- Solution: selection of the most important terms (feature selection).
- Example: intermediate document frequency
  - very frequent terms appear in almost all documents
  - very rare terms appear in almost no document

  In both cases, these terms are not helpful to distinguish most of the documents.

Problem 3

Approach:

1. document frequency for all terms: $DF(t_i) = \frac{|\{d|t_i \in d\}|}{|\mathcal{D}|}$
2. sort terms according to $DF(t_i)$

Rank	Term	DF
1.	t_{23}	0.82
2.	t_{17}	0.65
3.	t_{14}	0.52
4.	...	...

3. sort terms according to score$(t_i) = DF(t_i) \cdot$ rank(t_i)
   examples:
$$\text{score}(t_{23}) = 0.82 \cdot 1$$
$$\text{score}(t_{17}) = 0.65 \cdot 2$$
4. chose the k terms with largest score

Problem 4

- Not all terms are equally useful.
- Solution: TF-IDF (term frequency – inverse document frequency)
  - higher weights for rare words
  - higher weights for terms that are more frequent than others in some document
- TF is the relative term frequency in some document d: $TF(t,d) = \frac{n(t,d)}{\sum_{t_i \in d} n(t_i,d)}$
- IDF is the inverse document frequency of t for all documents: $IDF(t) = \frac{|\mathcal{D}|}{|\{d|d \in \mathcal{D} \wedge t \in d\}|}$
- feature vector for document $d = \begin{pmatrix} TF(t_1,d) \cdot IDF(t_1) \\ TF(t_2,d) \cdot IDF(t_2) \\ \vdots \\ TF(t_k,d) \cdot IDF(t_k) \end{pmatrix}$

Problem 5

- Most words from a dictionary don't occur at all in any of the compared documents ($f = 0$).
- As a consequence, the Euclidean distance will be very similar for many documents.
- Solution: use different distance measures
- Jaccard coefficient: documents as sets of terms:

$$\text{dist}_{\text{Jaccard}}(D_1, D_2) = 1 - \frac{|D_1 \cap D_2|}{|D_1 \cup D_2|}$$

- cosine coefficient (possibly for TF-IDF vectors):

$$\text{dist}_{\text{cosine}}(D_1, D_2) = 1 - \frac{\langle D_1, D_2 \rangle}{\|D_1\| \cdot \|D_2\|}$$

Outline

Your Choice of a Distance Measure

There are hundreds of distance functions [Deza and Deza, 2009].

- For time series: DTW, EDR, ERP, LCSS, ...
- For texts: Cosine and normalizations
- For sets – based on intersection, union, ... (Jaccard)
- For clusters (single-link, average-link, etc.)
- For histograms: histogram intersection, "Earth movers distance", quadratic forms with color similarity
- With normalization: Canberra, ...
- Quadratic forms: $d(x, y) := \sqrt{(x-y)M(x-y)^\mathsf{T}}$ for some positive (usually symmetric) definite matrix M.

Note that:

Choosing the appropriate distance function can be seen as a part of "preprocessing".

Summary

You learned in this section:

- *features and feature spaces*
- *categories of features (categorical/nominal, ordinal, metric)*
- *basic univariate feature descriptors (frequency (relative/absolute), mode, median, mean)*
- *distances (L_p-norms, weighted, quadratic form)*
- *feature (vector) descriptors for texts and for images*

Outline

Recommended Reading

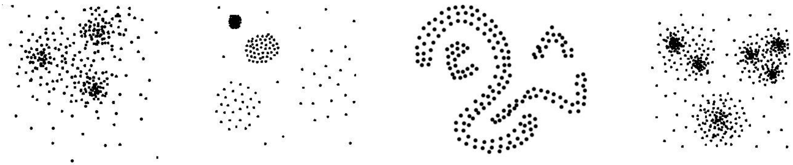
Recommended Reading:

- *Tan et al. [2006], Chapter 8.1+8.2+8.5*
- *Tan et al. [2020], Chapter 5.1+5.2+5.5*
- *Han et al. [2011], Chapter 10.1+10.2*
- *Zaki and Meira Jr. [2014], Chapter 13.1*

Outline

Purpose of Clustering

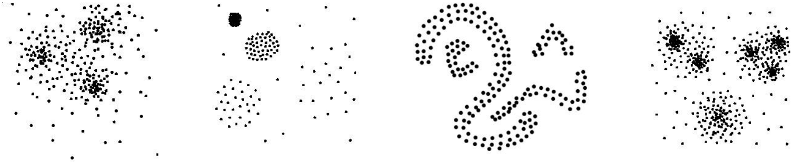
- identify a finite number of categories (classes, groups: “clusters”) in a given dataset
- *similar* objects shall be grouped in the same cluster, *dissimilar* objects in different clusters
- “similarity” is highly subjective, depending on the application scenario



Clustering is Unsupervised

Clustering is unsupervised, that is, we do not have any external knowledge to guide or to supervise the clustering process:

- we cannot learn rules to sort points into clusters
- we do not know how many clusters there are
- we do not know how the clusters are characterized
- there is no unique criterion to judge on the quality of a derived clustering solution (evaluation)



Challenges

Assume we are given a hypothetical quality function q to decide for a given partition of a set of n points whether or not this partition constitutes a (good) clustering.

- naïve method: test q on all possible clusterings with k partitions (clusters) $(2 \leq k \leq ?)$
- problems:
  - there are $\mathcal{O}(k^n)$ many partitions in k clusters
  - and we don't have this function q, actually

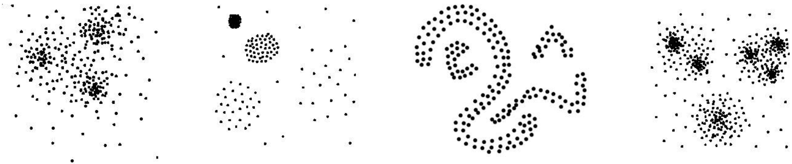
Therefore, we need heuristic solutions for both problems:

- *efficient* search for solutions
- efficient and effective modelling of q

There are many such heuristic solutions around, that is, we have a plethora of clustering algorithms in the literature.

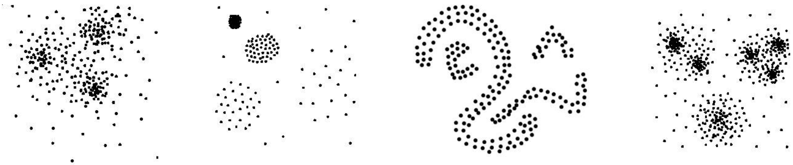
Categories of Clustering Approaches

- partitioning
  - model: clusters are compact sets of points
  - parameter: (usually) number k of clusters, distance measure
  - looks for a flat partitioning into k cluster with maximal compactness



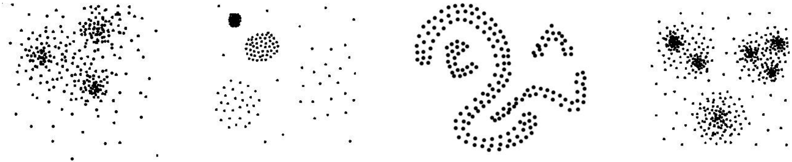
Categories of Clustering Approaches

- density-based
  - model: clusters are areas of high density, separated by areas of low density
  - parameter: minimal density in some cluster, distance measure
  - looks for a flat partitioning into clusters exceeding some minimal density



Categories of Clustering Approaches

- hierarchical
  - model: compactness, density,. . .
  - parameter: distance measure for points and for clusters
  - looks for a hierarchy of clusters (e.g., given as a tree), joins the most similar clusters at a given level of the hierarchy
  - flat clusters can be derived by cutting the tree on some level

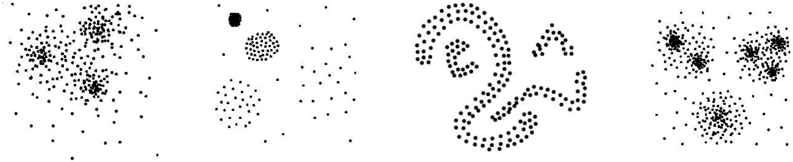


Categories of Clustering Approaches

Alternative categorizations could sort approaches according to the algorithmic approach or to the application scenario of specialized approaches:

- many variants of techniques, such as fuzzy clustering, graph-theoretic algorithms, neuronal nets
- specializations to special data characteristics (time series, graphs, high-dimensional data, stream data, uncertain data etc.)

examples: Aggarwal and Reddy [2013]



Outline

Objective

Partitional clustering partitions a dataset into k clusters, typically minimizing some cost function (compactness criterion).

Central assumptions for approaches in this family are typically:

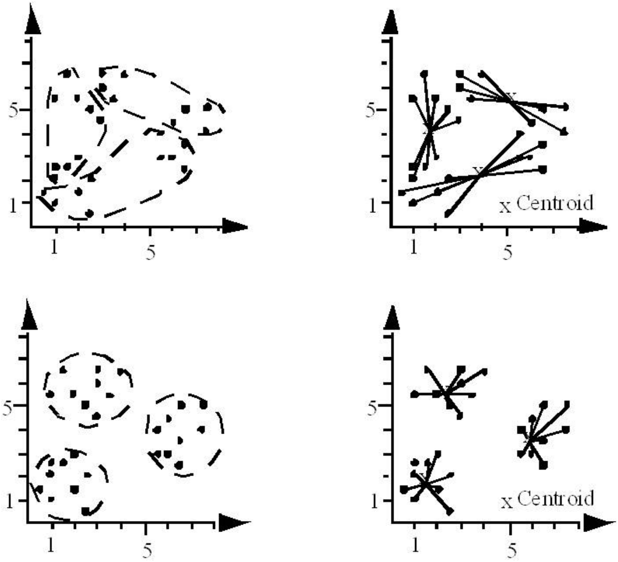
- number k of clusters known
- clusters are characterized by their compactness
- compactness measured by some distance function (e.g., distance of all objects in a cluster from some cluster representative is minimal)
- criterion of compactness typically leads to convex or even spherically shaped clusters

Basic Strategy

A complete search for the globally optimal solution is intractable, thus:

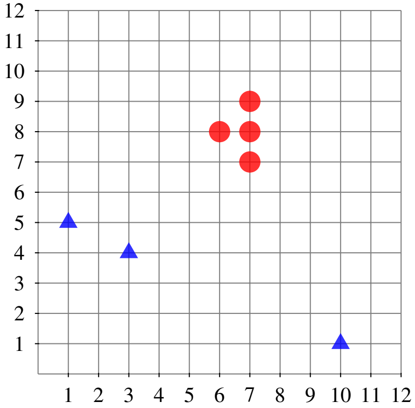
- locally optimizing algorithms
  - choose k initial representatives
  - assign each object to its closest representative
  - optimize the representatives iteratively
- types of cluster representatives
  - mean (centroid) – central object is constructed
  - some cluster object (e.g., medoid) – central object is chosen
  - Gaussian distribution model for a cluster – expectation maximization

Construction of Central Points: Example



Construction of Central Points: Basics

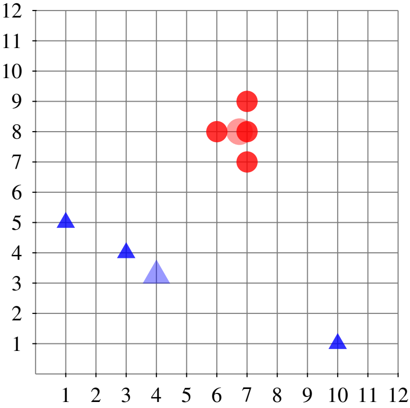
- objects are points $x = (x_1, \ldots, x_d)$ in Euclidean vector space $\mathbb{R}^d$, dist = Euclidean distance (L_2)
- centroid μ_C: mean vector of all points in cluster C



$$\mu_{C_i} = \frac{1}{|C_i|} \cdot \sum_{o \in C_i} o$$

Construction of Central Points: Basics

- objects are points $x = (x_1, \ldots, x_d)$ in Euclidean vector space $\mathbb{R}^d$, $\mathrm{dist} =$ Euclidean distance (L_2)
- centroid μ_C: mean vector of all points in cluster C



$$\mu_{C_i} = \frac{1}{|C_i|} \cdot \sum_{o \in C_i} o$$

Construction of Central Points: Basics

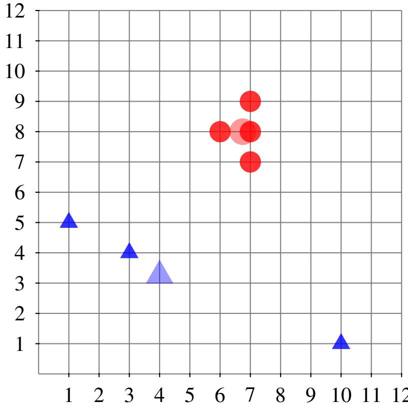
- measure of compactness for a cluster C:

$$TD^2(C) = \sum_{p \in C} \text{dist}(p, \mu_C)^2$$

(a.k.a. SSQ: sum of squares or SSE: sum of squared error)

- measure of compactness for a clustering

$$TD^2(C_1, C_2, \ldots, C_k) = \sum_{i=1}^{k} TD^2(C_i)$$



Construction of Central Points: Basics

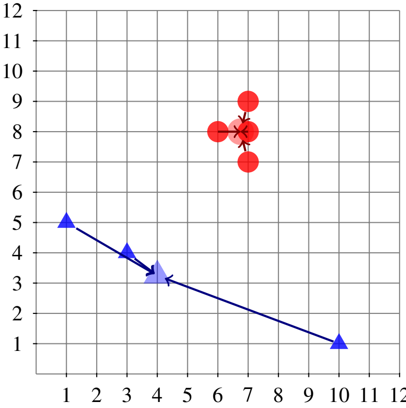
- measure of compactness for a cluster C:

$$TD^2(C) = \sum_{p \in C} \text{dist}(p, \mu_C)^2$$

(a.k.a. SSQ: sum of squares or SSE: sum of squared error)

- measure of compactness for a clustering

$$TD^2(C_1, C_2, \ldots, C_k) = \sum_{i=1}^{k} TD^2(C_i)$$

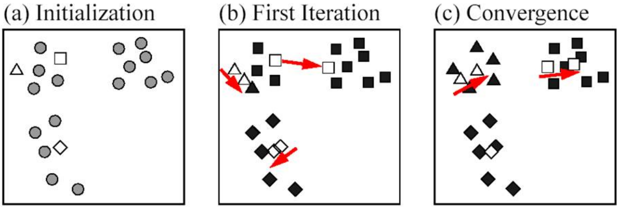


Algorithm: Clustering by Minimization of Variance [Forgy, 1965, Lloyd, 1982]

Algorithm 4.1 (Clustering by Minimization of Variance)

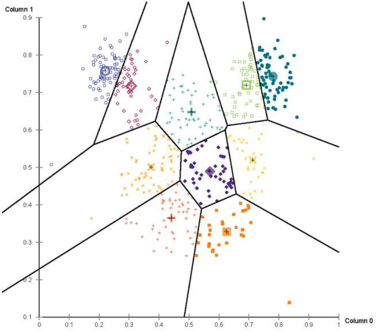
1. *start with k (e.g., randomly selected) points as cluster representatives (or a random partition in k "clusters")*
2. *repeat:*
   2.1 *assign each point to the closest representative*
   2.2 *compute new representatives based on the given partitions (centroid of the assigned points)*

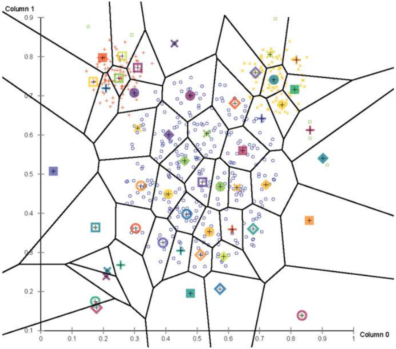
until there is no change in assignment



Model

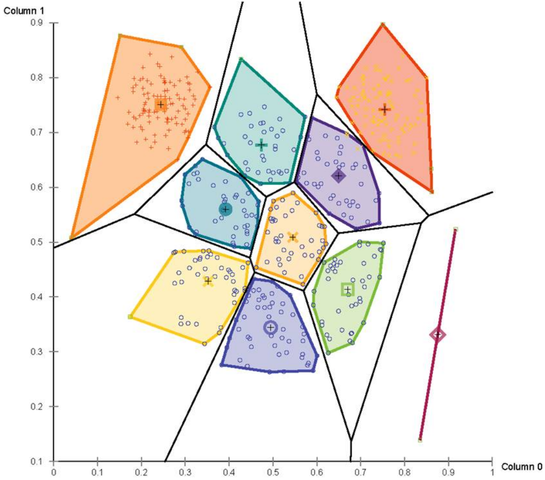
The assignment to the closest representative corresponds to a Voronoi parcelling



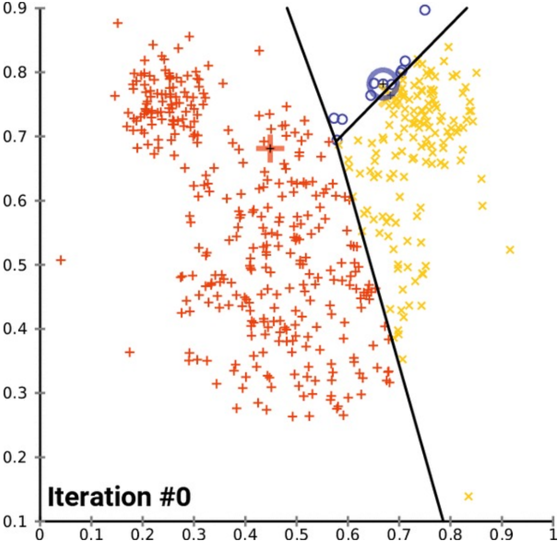


Model

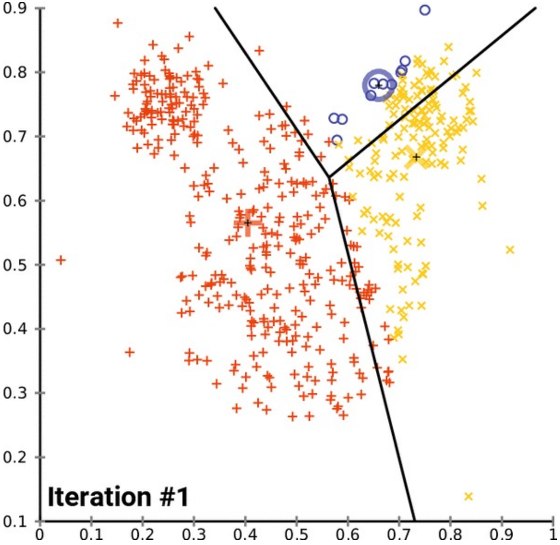
Voronoi parcelling $\neq$ convex hull!



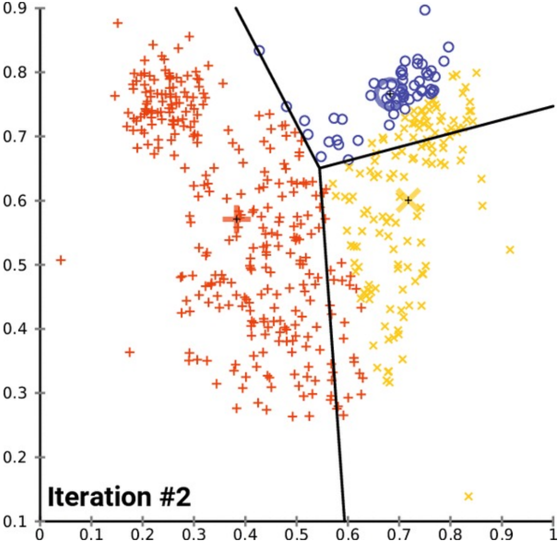
Convergence: Visualization of Evolving Voronoi Parcelling



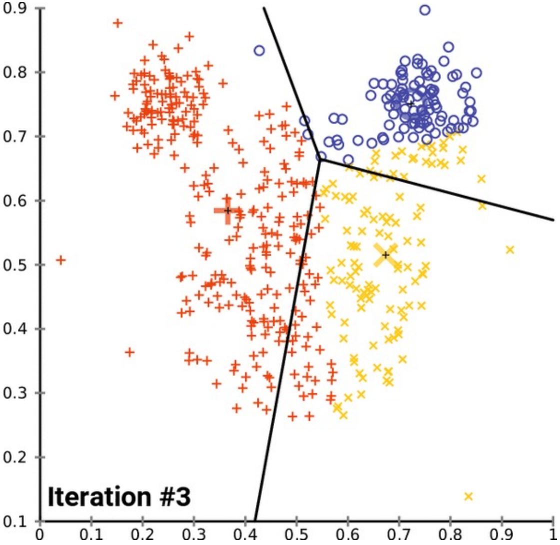
Convergence: Visualization of Evolving Voronoi Parcelling



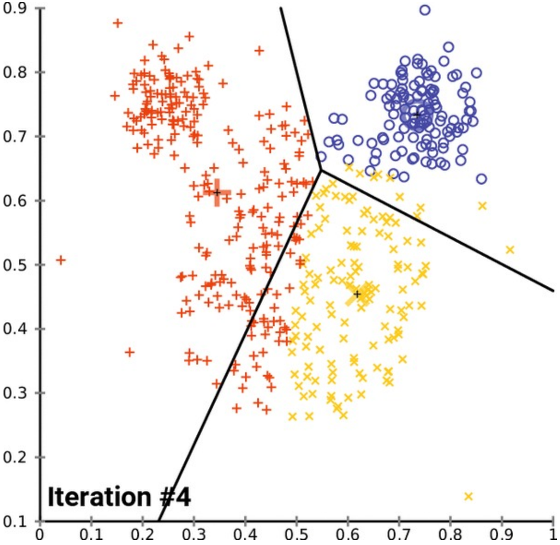
Convergence: Visualization of Evolving Voronoi Parcelling



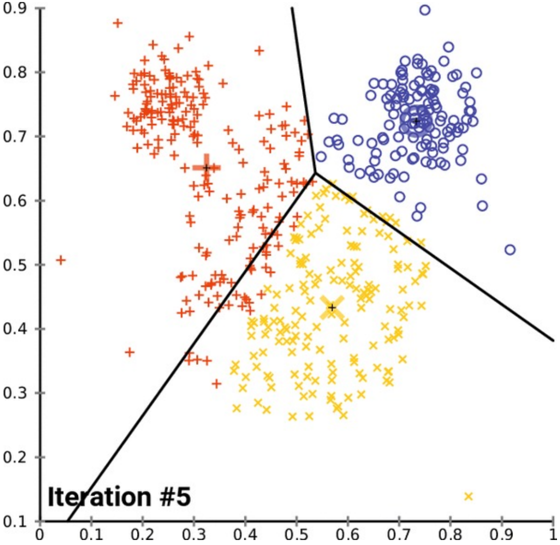
Convergence: Visualization of Evolving Voronoi Parcelling



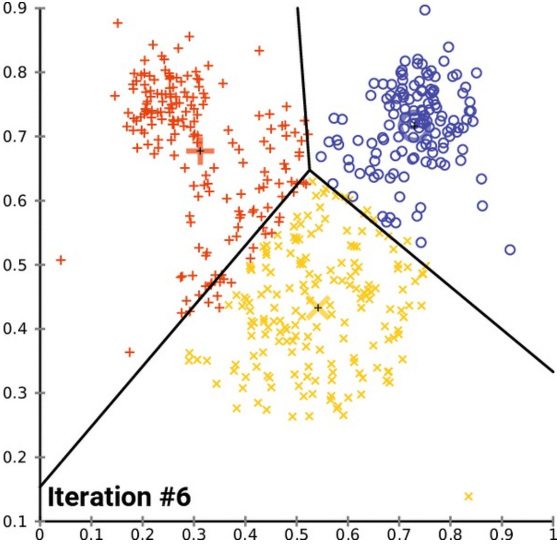
Convergence: Visualization of Evolving Voronoi Parcelling



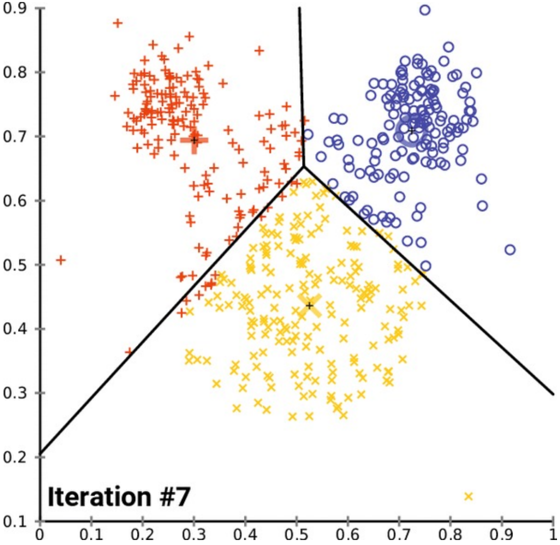
Convergence: Visualization of Evolving Voronoi Parcelling



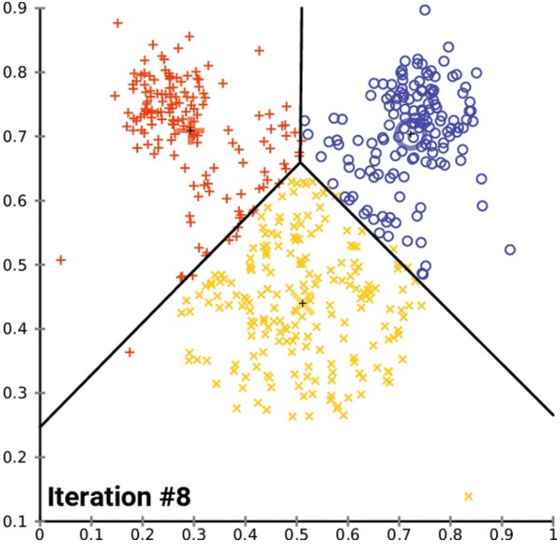
Convergence: Visualization of Evolving Voronoi Parcelling



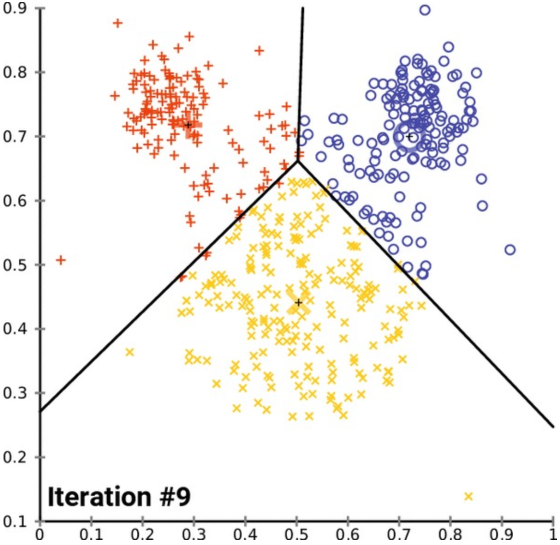
Convergence: Visualization of Evolving Voronoi Parcelling



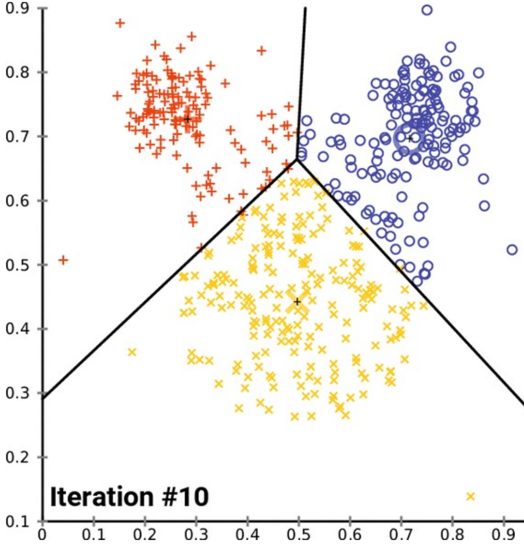
Convergence: Visualization of Evolving Voronoi Parcelling



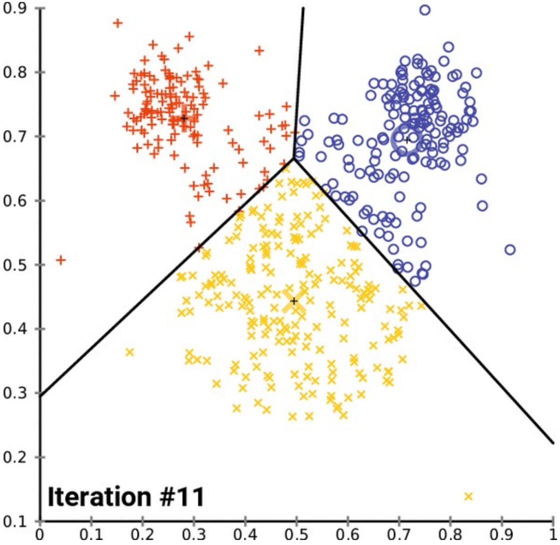
Convergence: Visualization of Evolving Voronoi Parcelling



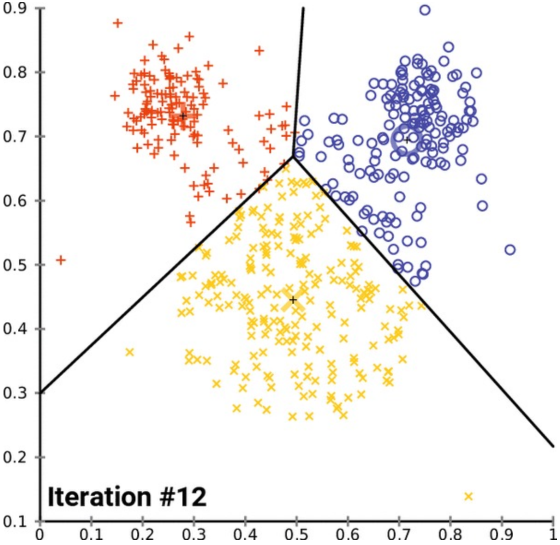
Convergence: Visualization of Evolving Voronoi Parcelling



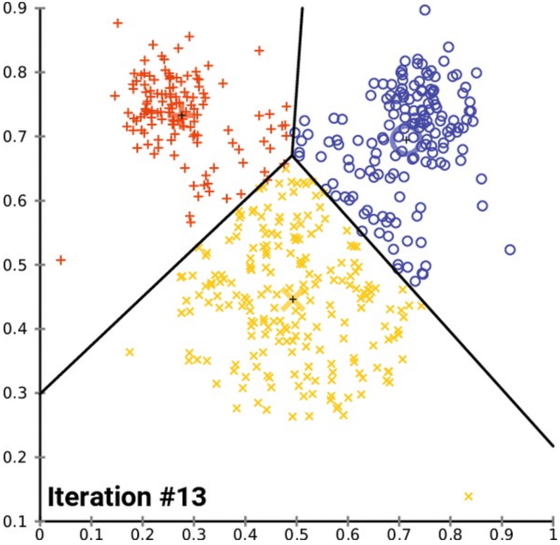
Convergence: Visualization of Evolving Voronoi Parcelling



Convergence: Visualization of Evolving Voronoi Parcelling



Convergence: Visualization of Evolving Voronoi Parcelling



k-means

k-means [MacQueen, 1967] is the best known variant of the basic algorithm:

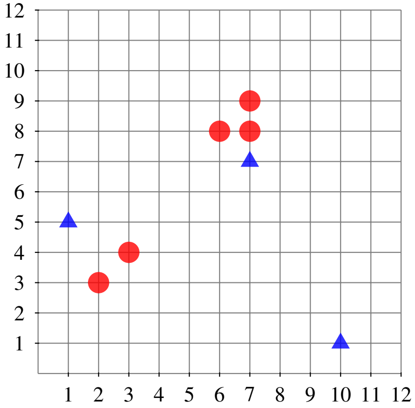
- a centroid is immediately updated when some point changes its assignment
- k-means has very similar properties, but the result now depends on the order of data points in the input file

Note that:

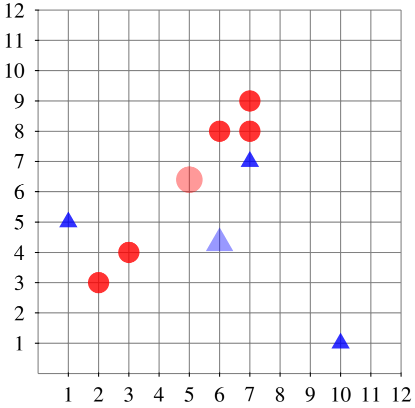
The name "k-means" is often used indifferently for any variant of the basic algorithm, in particular also for the algorithm of Forgy [1965], Lloyd [1982].

Outline

k-means Clustering – Lloyd/Forgy Algorithm



k-means Clustering – Lloyd/Forgy Algorithm

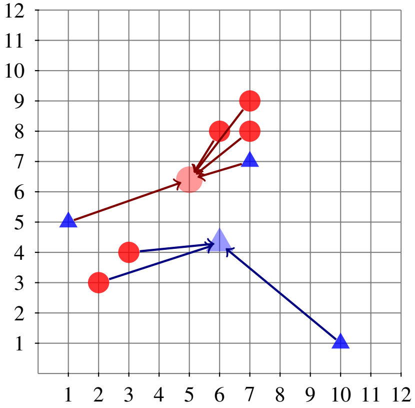


recompute centroids:

$$\mu \approx (6.0, 4.3)$$

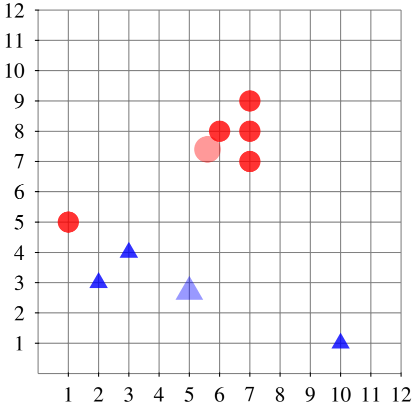
$$\mu \approx (5.0, 6.4)$$

k-means Clustering – Lloyd/Forgy Algorithm



reassign points

k-means Clustering – Lloyd/Forgy Algorithm

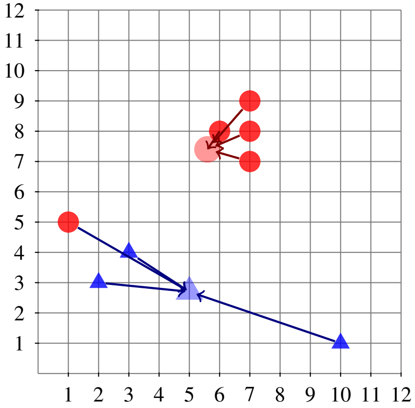


recompute centroids:

$$\mu \approx (5.0, 2.7)$$

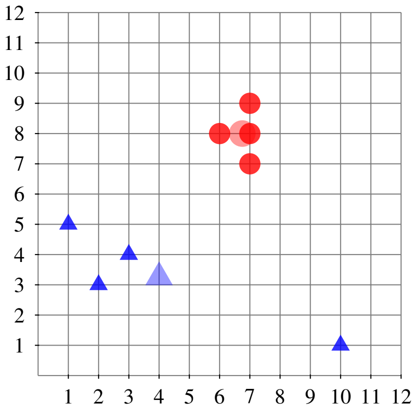
$$\mu \approx (5.6, 7.4)$$

k-means Clustering – Lloyd/Forgy Algorithm



reassign points

k-means Clustering – Lloyd/Forgy Algorithm

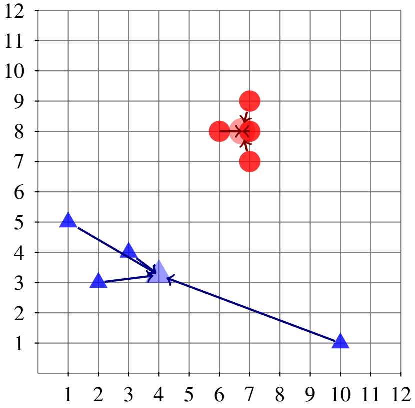


recompute centroids:

$\mu \approx (4.0, 3.25)$

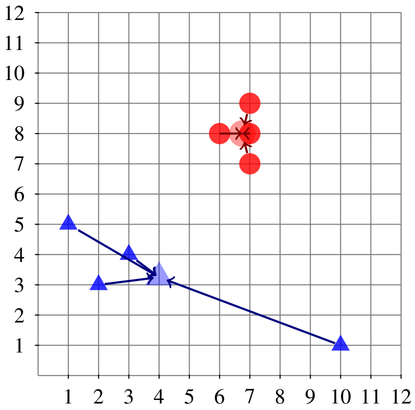
$\mu \approx (6.75, 8.0)$

k-means Clustering – Lloyd/Forgy Algorithm



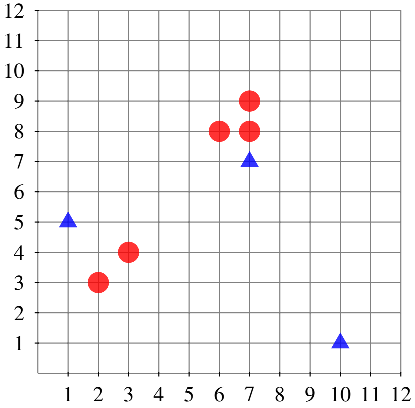
reassign points

k-means Clustering – Lloyd/Forgy Algorithm

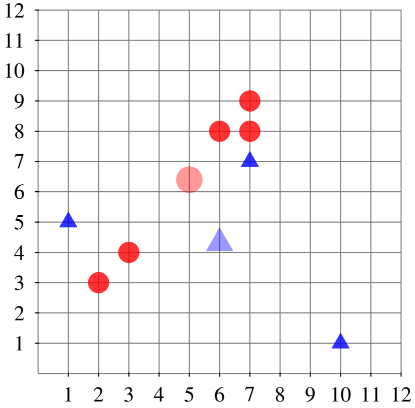


reassign points
no change
convergence!

k-means Clustering – MacQueen Algorithm



k-means Clustering – MacQueen Algorithm

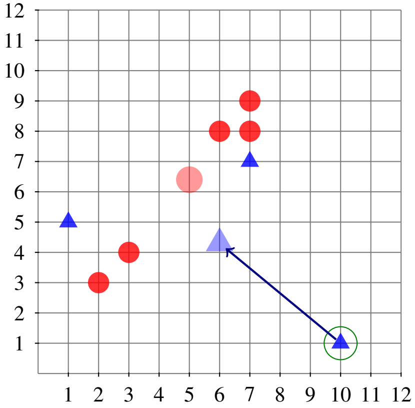


Centroids (e.g.: from previous iteration):

$\mu \approx (6.0, 4.3)$

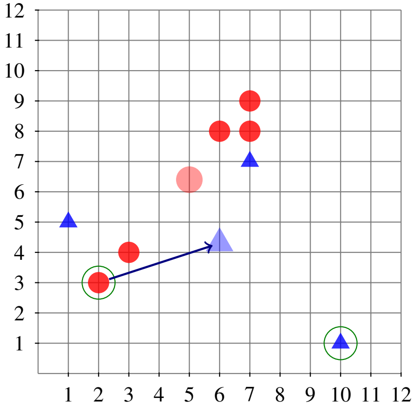
$\mu \approx (5.0, 6.4)$

k-means Clustering – MacQueen Algorithm



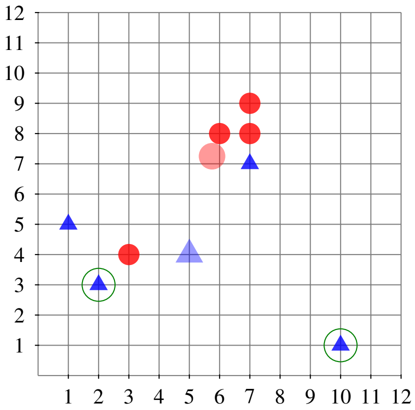
assign first point

k-means Clustering – MacQueen Algorithm



assign second point

k-means Clustering – MacQueen Algorithm

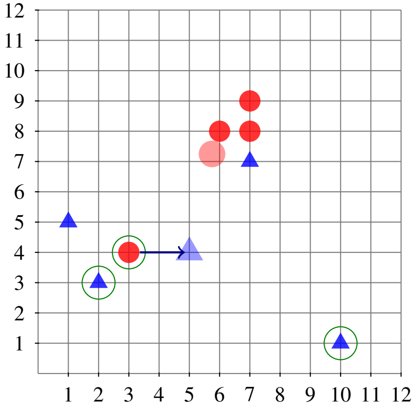


recompute centroids:

$\mu \approx (5.0, 4.0)$

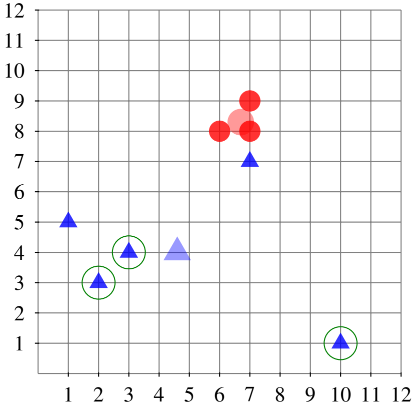
$\mu \approx (5.75, 7.25)$

k-means Clustering – MacQueen Algorithm



assign third point

k-means Clustering – MacQueen Algorithm

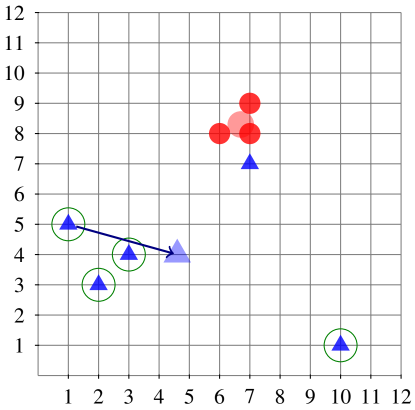


recompute centroids:

$$\mu \approx (4.6, 4.0)$$

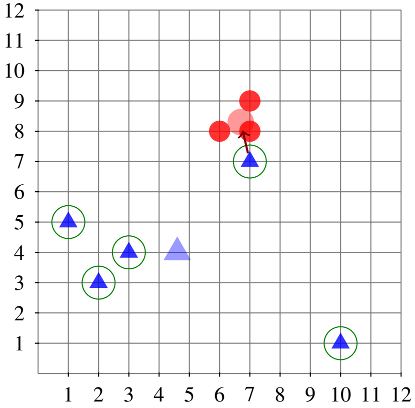
$$\mu \approx (6.7, 8.3)$$

k-means Clustering – MacQueen Algorithm



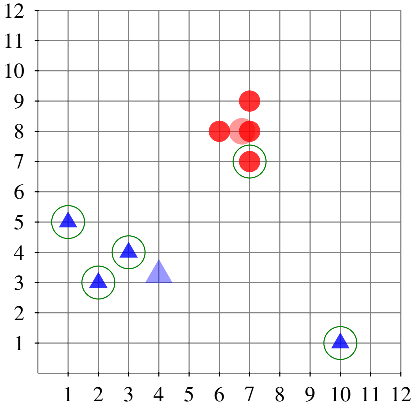
assign fourth point

k-means Clustering – MacQueen Algorithm



assing fifth point

k-means Clustering – MacQueen Algorithm

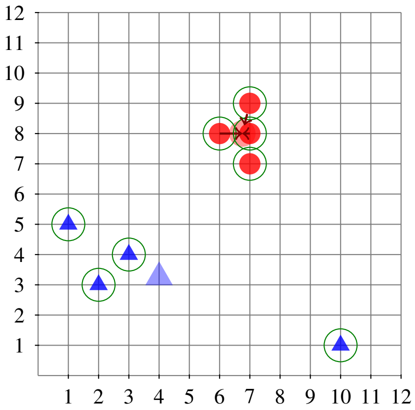


recompute centroids:

$\mu \approx (4.0, 3.25)$

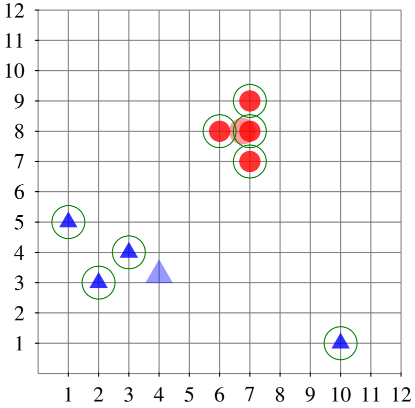
$\mu \approx (6.75, 8.0)$

k-means Clustering – MacQueen Algorithm



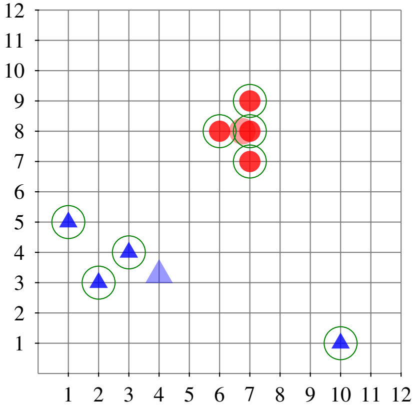
reassign more points

k-means Clustering – MacQueen Algorithm



reassign more points
possibly more iterations

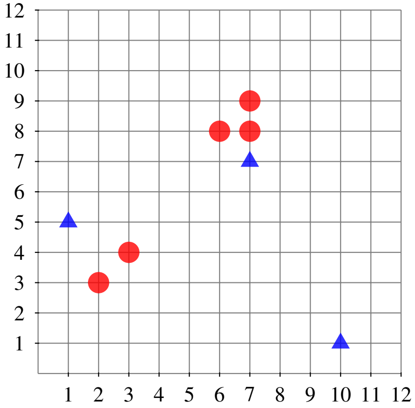
k-means Clustering – MacQueen Algorithm



reassign more points
possibly more iterations
convergence

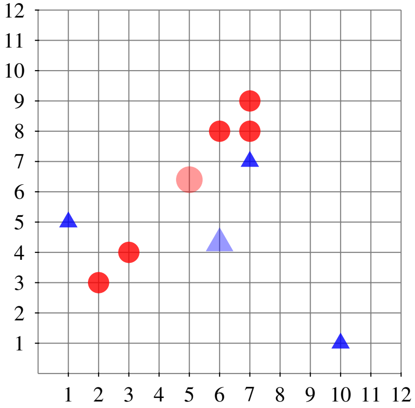
k-means Clustering – MacQueen Algorithm

Alternative Run – Different Order



k-means Clustering – MacQueen Algorithm

Alternative Run – Different Order



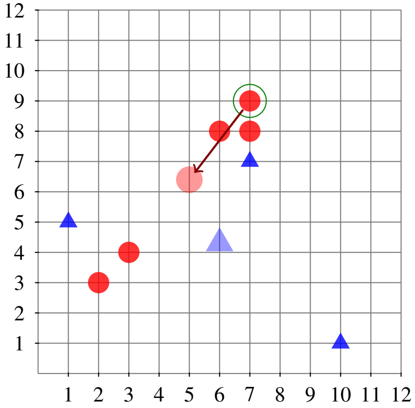
Centroids (e.g.: from previous iteration):

$$\mu \approx (6.0, 4.3)$$

$$\mu \approx (5.0, 6.4)$$

k-means Clustering – MacQueen Algorithm

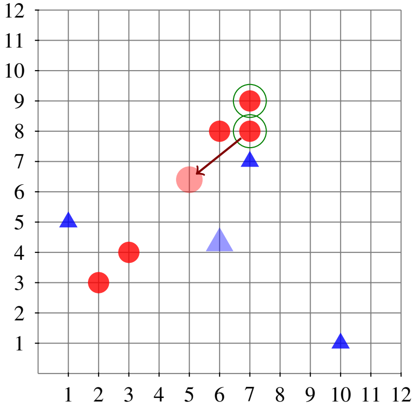
Alternative Run – Different Order



assign first point

k-means Clustering – MacQueen Algorithm

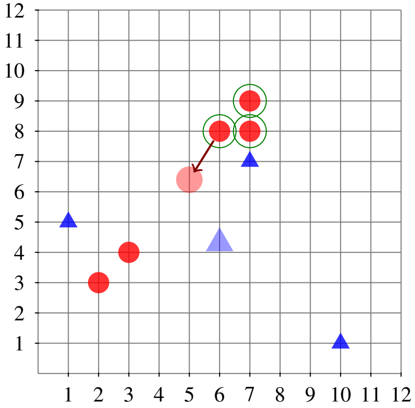
Alternative Run – Different Order



assign second point

k-means Clustering – MacQueen Algorithm

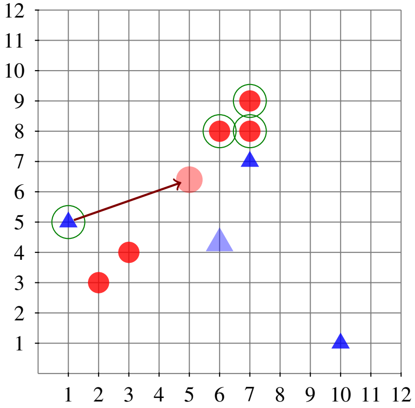
Alternative Run – Different Order



assign third point

k-means Clustering – MacQueen Algorithm

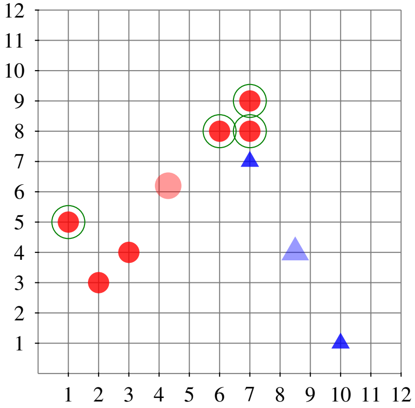
Alternative Run – Different Order



assign fourth point

k-means Clustering – MacQueen Algorithm

Alternative Run – Different Order



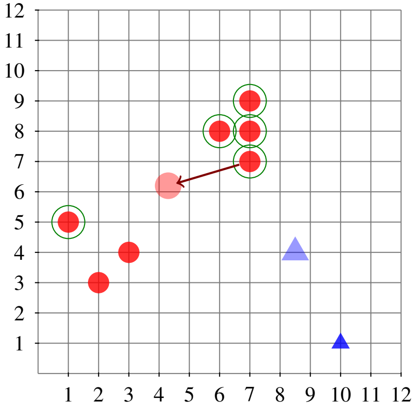
recompute centroids:

$\mu \approx (4.0, 8.5)$

$\mu \approx (4.3, 6.2)$

k-means Clustering – MacQueen Algorithm

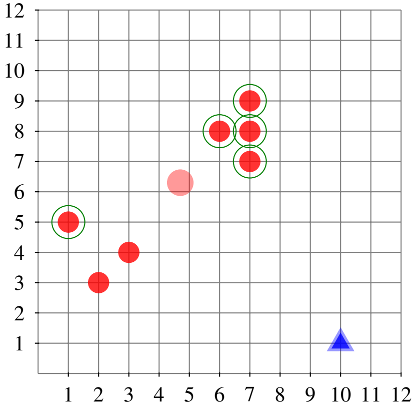
Alternative Run – Different Order



assign fifth point

k-means Clustering – MacQueen Algorithm

Alternative Run – Different Order



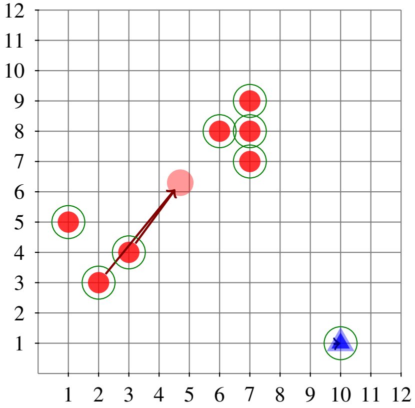
recompute centroids:

$$\mu \approx (10.0, 1.0)$$

$$\mu \approx (4.7, 6.3)$$

k-means Clustering – MacQueen Algorithm

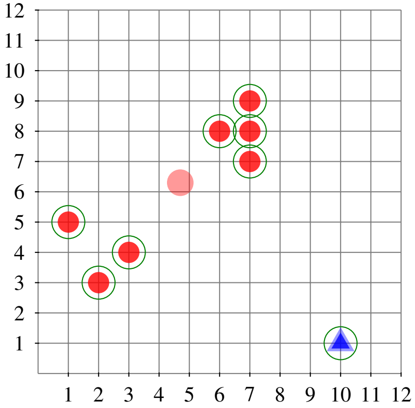
Alternative Run – Different Order



reasign more points

k-means Clustering – MacQueen Algorithm

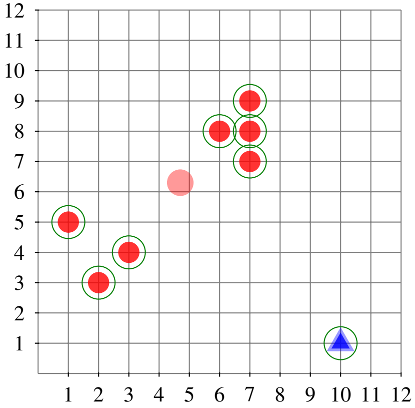
Alternative Run – Different Order



reasign more points
possibly more iterations

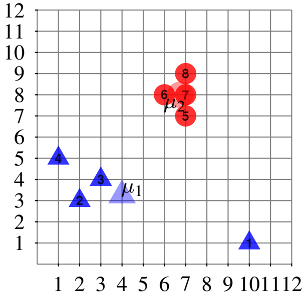
k-means Clustering – MacQueen Algorithm

Alternative Run – Different Order



reasign more points
possibly more iterations
convergence

k-means Clustering – Quality



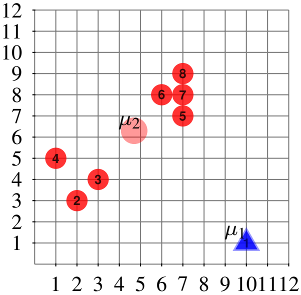
First solution: $TD^2 = 61\,\frac{1}{2}$

$$SSQ(\mu_1, p_1) = |4-10|^2 + |3.25-1|^2 = 36 + 5\,\tfrac{1}{16} = 41\,\tfrac{1}{16}$$
$$SSQ(\mu_1, p_2) = |4-2|^2 + |3.25-3|^2 = 4 + \tfrac{1}{16} = 4\,\tfrac{1}{16}$$
$$SSQ(\mu_1, p_3) = |4-3|^2 + |3.25-4|^2 = 1 + \tfrac{9}{16} = 1\,\tfrac{9}{16}$$
$$SSQ(\mu_1, p_4) = |4-1|^2 + |3.25-5|^2 = 9 + 3\,\tfrac{1}{16} = 12\,\tfrac{1}{16}$$
$$TD^2(C_1) = 58\,\tfrac{3}{4}$$

$$SSQ(\mu_2, p_5) = |6.75-7|^2 + |8-7|^2 = \tfrac{1}{16} + 1 = 1\,\tfrac{1}{16}$$
$$SSQ(\mu_2, p_6) = |6.75-6|^2 + |8-8|^2 = \tfrac{9}{16} + 0 = \tfrac{9}{16}$$
$$SSQ(\mu_2, p_7) = |6.75-7|^2 + |8-8|^2 = \tfrac{1}{16} + 0 = \tfrac{1}{16}$$
$$SSQ(\mu_2, p_8) = |6.75-7|^2 + |8-9|^2 = \tfrac{1}{16} + 1 = 1\,\tfrac{1}{16}$$
$$TD^2(C_2) = 2\,\tfrac{3}{4}$$

Note: $SSQ(\mu, p) = \textit{Euclidean}(\mu, p)^2 = L_2^2(\mu, p)$.

k-means Clustering – Quality



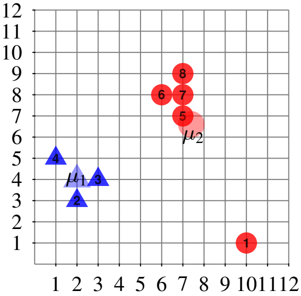
$SSQ(\mu_1, p_1) = |10 - 10|^2 + |1 - 1|^2 = 0$
$TD^2(C_1) = 0$

$SSQ(\mu_2, p_2) \approx |4.7 - 2|^2 + |6.3 - 3|^2 \approx 18.2$
$SSQ(\mu_2, p_3) \approx |4.7 - 3|^2 + |6.3 - 4|^2 \approx 8.2$
$SSQ(\mu_2, p_4) \approx |4.7 - 1|^2 + |6.3 - 5|^2 \approx 15.4$
$SSQ(\mu_2, p_5) \approx |4.7 - 7|^2 + |6.3 - 7|^2 \approx 5.7$
$SSQ(\mu_2, p_6) \approx |4.7 - 6|^2 + |6.3 - 8|^2 \approx 4.6$
$SSQ(\mu_2, p_7) \approx |4.7 - 7|^2 + |6.3 - 8|^2 \approx 8.2$
$SSQ(\mu_2, p_7) \approx |4.7 - 7|^2 + |6.3 - 9|^2 \approx 12.6$
$TD^2(C_2) \approx 72.86$

First solution: $TD^2 = 61\frac{1}{2}$
Second solution: $TD^2 \approx 72.68$

Note: $SSQ(\mu, p) = \mathit{Euclidean}(\mu, p)^2 = L_2^2(\mu, p)$.

k-means Clustering – Quality



$SSQ(\mu_1, p_2) = |2-2|^2 + |4-3|^2 = 0 + 1 = 1$
$SSQ(\mu_1, p_3) = |2-3|^2 + |4-4|^2 = 1 + 0 = 1$
$SSQ(\mu_1, p_4) = |2-1|^2 + |4-5|^2 = 1 + 1 = 2$
$TD^2(C_1) = 4$

$SSQ(\mu_2, p_1) = |7.4-10|^2 + |6.6-1|^2 = 6\frac{19}{25} + 31\frac{9}{25} = 38\frac{3}{25}$
$SSQ(\mu_2, p_5) = |7.4-7|^2 + |6.6-7|^2 = \frac{4}{25} + \frac{4}{25} = \frac{8}{25}$
$SSQ(\mu_2, p_6) = |7.4-6|^2 + |6.6-8|^2 = 1\frac{24}{25} + 1\frac{24}{25} = 3\frac{23}{25}$
$SSQ(\mu_2, p_7) = |7.4-7|^2 + |6.6-8|^2 = \frac{4}{25} + 1\frac{24}{25} = 2\frac{3}{25}$
$SSQ(\mu_2, p_8) = |7.4-7|^2 + |6.6-9|^2 = \frac{4}{25} + 5\frac{19}{25} = 5\frac{23}{25}$
$TD^2(C_2) = 50\frac{2}{5}$

First solution: $TD^2 = 61\frac{1}{2}$
Second solution: $TD^2 \approx 72.68$
Optimal solution: $TD^2 = 54\frac{2}{5}$
Try it yourself: `https://elki-project.github.io/`

Note: $SSQ(\mu, p) = \mathit{Euclidean}(\mu, p)^2 = L_2^2(\mu, p)$.

Discussion

pros

- efficient: $\mathcal{O}(k \cdot n)$ per iteration, number of iterations is usually in the order of 10.
- easy to implement, thus very popular

cons

- k-means converges towards a *local* minimum
- k-means (MacQueen-variant) is order-dependent
- deteriorates with noise and outliers (all points are used to compute centroids)
- clusters need to be convex and of (more or less) equal extension
- number k of clusters is hard to determine
- strong dependency on initial partition (in result quality as well as runtime)

Outline

Selection of Representative Points: k-medoids

works in a general metric space:

- requires only distance measure dist for pairs of objects
- medoid as representative: a central object of the cluster
- measure of compactness for a cluster C with medoid m_C:

$$TD(C) = \sum_{p \in C} \mathrm{dist}(p, m_C)$$

- measure of compactness for a clustering

$$TD(C_1, C_2, \ldots, C_k) = \sum_{i=1}^{k} TD(C_i)$$

- The medoid of a cluster is the point among the cluster members that minimizes TD.

k-modes [Huang, 1997]

For categorical data: use the mode as an analogue to the centroid:

- in numerical attributes: centroid $\bar{x}$ of C minimizes $TD^2(C,\bar{x}) = \sum_{p \in C} \operatorname{dist}(p,\bar{x})^2$
- categorical attributes: mode m of C minimizes $TD(C,m) = \sum_{p \in C} \operatorname{dist}(p,m)$, where m is not necessarily an element of C
- distance function for categorical attributes, e.g.:

$$\operatorname{dist}(x,y) = \sum_{i=1}^{d} \delta(x_i,y_i), \text{ where } \delta(x_i,y_i) = \begin{cases} 0 & \text{, if } x_i = y_i \\ 1 & \text{, else} \end{cases}$$

Find the Mode

- $TD(C,m) = \sum_{p \in C} \operatorname{dist}(p,m)$ is minimized, iff for $m = (m_1, \ldots, m_d)$ and all attributes $A_i, i = 1, \ldots, d$ holds: There is no attribute value more frequent than m_i in A_i.
- the mode is not necessarily unique
- example: $\mathcal{D} = \{(a,b),(a,c),(c,b),(b,c)\}$
- possible modes:

Find the Mode

- $TD(C, m) = \sum_{p \in C} \mathrm{dist}(p, m)$ is minimized, iff for $m = (m_1, \ldots, m_d)$ and all attributes $A_i, i = 1, \ldots, d$ holds: There is no attribute value more frequent than m_i in A_i.
- the mode is not necessarily unique
- example: $\mathcal{D} = \{(a, b), (a, c), (c, b), (b, c)\}$
- possible modes: $(a, b), (a, c)$

Algorithm k-modes

- initialization: randomly selected objects as initial modes
- use mode instead of centroid
- use categorical distance measure instead of squared Euclidean distance

Then follow Algorithm 4.1.

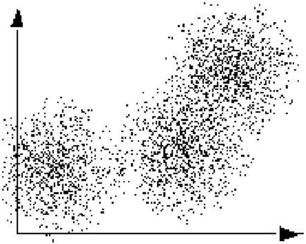
Outline

Choice of Initial Model [Fayyad et al., 1998]

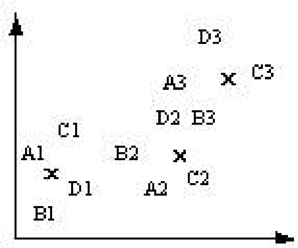
clustering on a small sample usually delivers good initial clusters, but some samples are deviating too much from the overall data distribution

- draw independently m samples
- clustering of each sample results in m estimates of the k cluster centers
- run clustering algorithm on $\mathcal{D}$ m-times (with the m estimated start configurations)
- choose the clustering result with the best value for some quality criterion

Example



$k = 3$ Gaussians



centers of $m = 4$ samples
X: true centers

k-means++ Initialization [Arthur and Vassilvitskii, 2007]

Algorithm 4.2 (k-means++ [Arthur and Vassilvitskii, 2007])

1. *Choose a first centroid randomly.*
2. *Compute for each point the distance to the closest of the already existing centroids.*
3. *Choose a new centroid from the points with a probability proportional to the squared distance.*
4. *Repeat 2 and 3 until k centroids have been chosen.*
5. *Continue with step 2 of Algorithm 4.1.*

Outline

Choice of k

Method:

- do a clustering for $k = 2, 3, 4, \ldots, n-1$ (or some other reasonable upper limit, e.g., $n/2$, $\sqrt{n}$)
- choose the best of the results

What is "the best"?

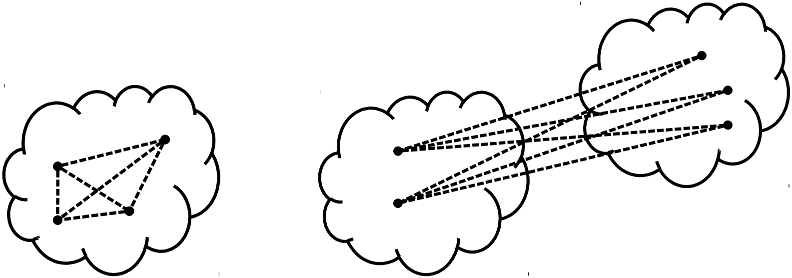
- quality measure needs to be independent of k
- k-means, k-medoid etc.: TD^2 and TD improve (i.e., become smaller) with increasing k

Evaluation of Clusterings

For the evaluation of clusterings we are mainly interested in two aspects:

cohesion: how tightly are members of one cluster connected, how compact are clusters?

separation: how well are different clusters separated?



Validity of Clusterings in General

- A validity measure for clusterings is typically some combination of cohesion and separation.
- Weights w_i could consider, e.g., the cluster size.

For a clustering C with k clusters $C_1, \ldots, C_k$, e.g.:

$$\text{overall\_validity}(C) = \sum_{i=1}^{k} w_i \cdot \text{validity}(C_i)$$

Validity based on some measure of proximity (similarity-, distance-function):

$$\begin{aligned} \text{cohesion}(C_i) &= \sum_{(x,y) \in C_i \times C_i} \text{proximity}(x, y) \\ \text{separation}(C_i, C_j) &= \sum_{(x,y) \in C_i \times C_j} \text{proximity}(x, y) \end{aligned}$$

Silhouette [Rousseeuw, 1987]

- let $a(o)$ be the average distance between o and all other members of the same cluster
- let $b(o)$ be the average distance between o and all members of another cluster that minimizes this average
- the silhouette of o is given by

$$s(o) = \frac{b(o) - a(o)}{\max(a(o), b(o))}$$

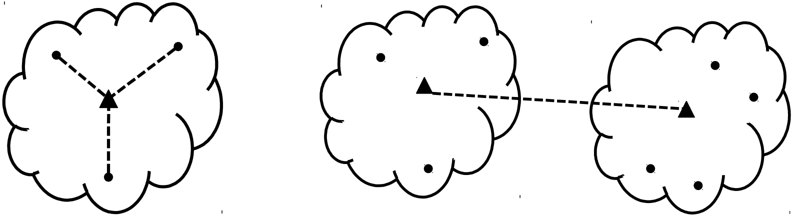
- it holds that $-1 \leq s(o) \leq 1$
- $s(o) \approx -1, 0, 1$: bad, indifferent, good assignment of o

Simplification

measure proximity w.r.t. the prototypes c_i of clusters C_i:

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

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



Even more simplified:

$$\text{separation}(C_i, C_j) = \text{proximity}(c_i, c_j)$$

Simplified Silhouette Coefficient [Rousseeuw, 1987]: Points

- let $a(o)$ be the distance between o and its "own" cluster representative
- let $b(o)$ be the distance between o and the closest "foreign" cluster representative
- the silhouette of o is given by

$$s(o) = \frac{b(o) - a(o)}{\max(a(o), b(o))}$$

- it holds that $-1 \leq s(o) \leq 1$
- $s(o) \approx -1, 0, 1$: bad, indifferent, good assignment of o

Example: Simplified Silhouette of Individual Points

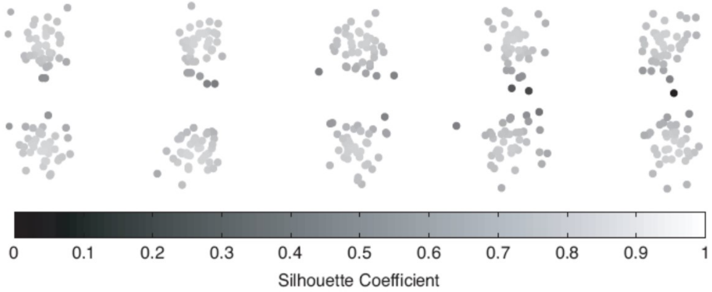


Figure 8.29. Silhouette coefficients for points in ten clusters.

(Figure from Tan et al. [2006].)

Simplified Silhouette Coefficient [Rousseeuw, 1987]: Clustering

- given the individual silhouette values $s(o)$
- the silhouette s_C of a clustering C is the average silhouette of all n points:

$$s_C = \frac{1}{n} \sum_{o \in \mathcal{D}} s(o)$$

- interpretation: $s_C > 0.7$: strong cluster structure, $s_C > 0.5$: reasonable cluster structure

Comparison: Silhouette – TD^2

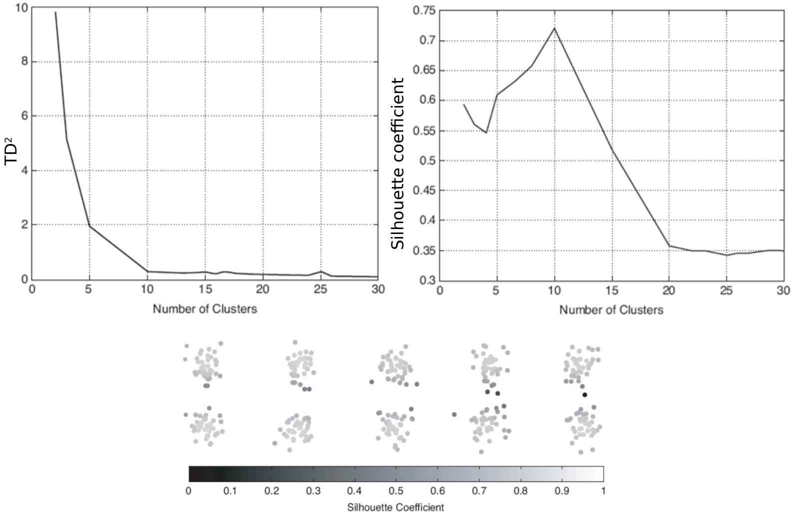
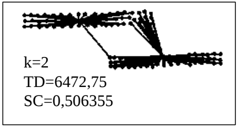
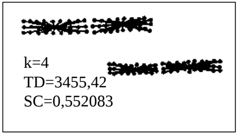
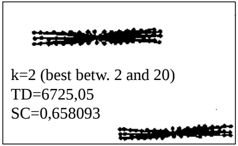


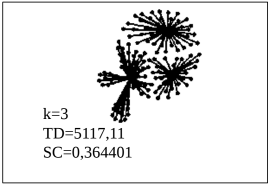
Figure 8.29. Silhouette coefficients for points in ten clusters.

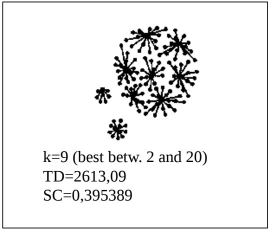
Examples: Silhouette – TD^2

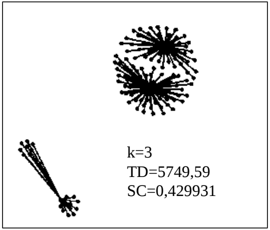


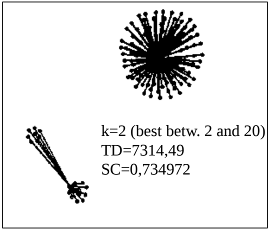






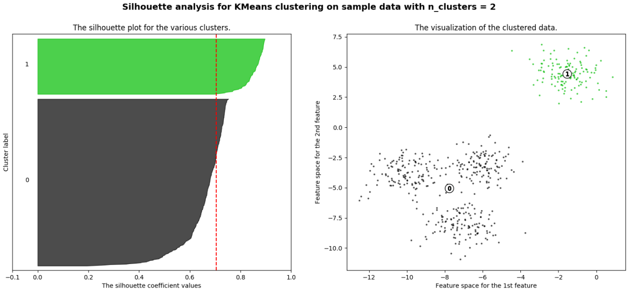






SDU SYDDANSK UNIVERSITET

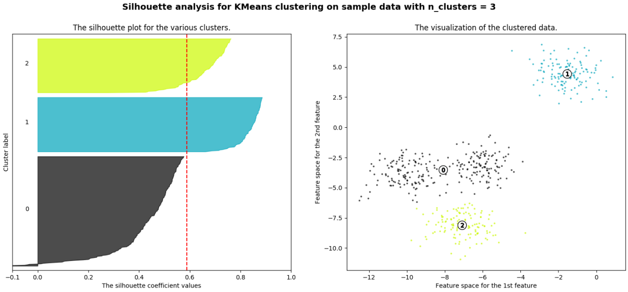
Visual Evaluation: Silhouette Plot



(Example by Pedregosa et al. [2011].)

Explore the code on `http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html`.

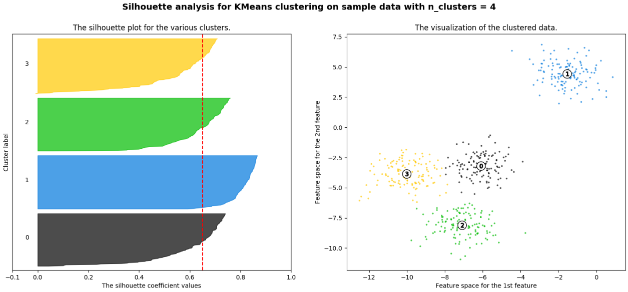
Visual Evaluation: Silhouette Plot



(Example by Pedregosa et al. [2011].)

Explore the code on `http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html`.

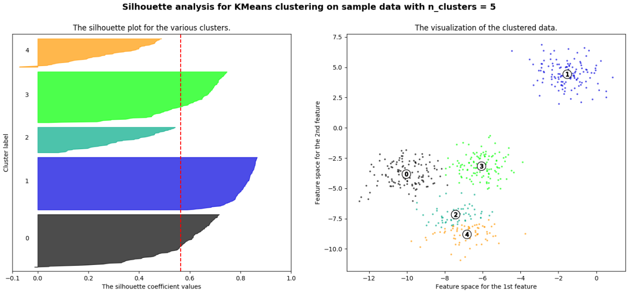
Visual Evaluation: Silhouette Plot



(Example by Pedregosa et al. [2011].)

Explore the code on `http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html`.

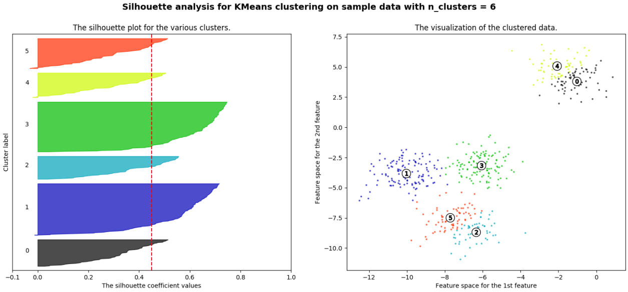
Visual Evaluation: Silhouette Plot



(Example by Pedregosa et al. [2011].)

Explore the code on `http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html`.

Visual Evaluation: Silhouette Plot

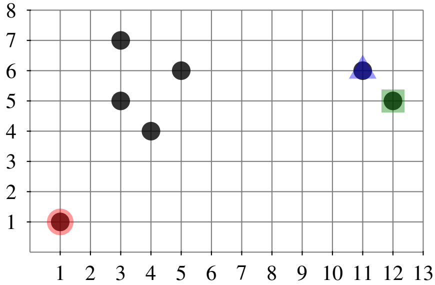


(Example by Pedregosa et al. [2011].)

Explore the code on `http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html`.

Outline

A Problem for the k-Means Clustering – Lloyd/Forgy Algorithm

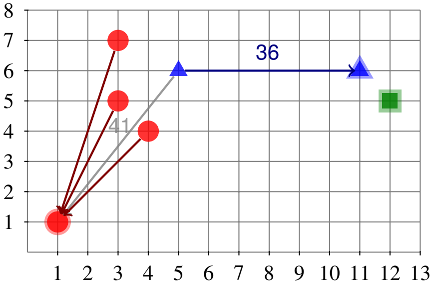


reassign objects

A Problem for the k-Means Clustering – Lloyd/Forgy Algorithm

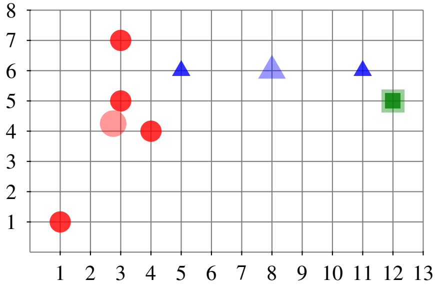
DM566

Melih Kandemir



reassign objects

A Problem for the k-Means Clustering – Lloyd/Forgy Algorithm



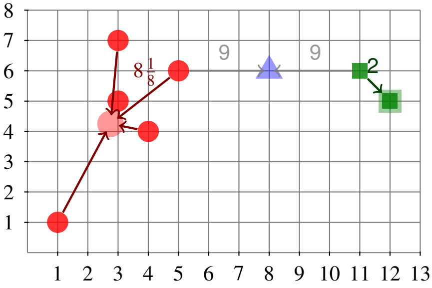
recompute centroids:

$\mu = (2.75, 4.25)$

$\mu = (8.0, 6.0)$

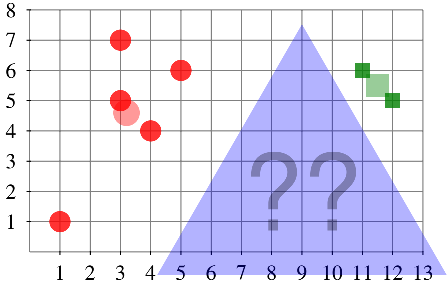
$\mu = (12.0, 5.0)$

A Problem for the k-Means Clustering – Lloyd/Forgy Algorithm



reassign objects

A Problem for the k-Means Clustering – Lloyd/Forgy Algorithm



recompute centroids:
Oops.
Average of an empty set?

Strategies for Handling Empty Clusters

There is no golden (standard) solution. Possibilities are:

- Report one cluster less.
- Choose the point that is farthest away from any current centroid as a new centroid.
- Choose (randomly) a point from that cluster that currently exhibits the largest TD^2 as a new centroid.

Note that:

Each of these solutions has a certain impact on the sum of squared error (TD^2, the clustering objective function).
For which of these solutions would you argue?

Outline

Opportunities for Further Research

- The silhouette is one of the most prominent, but just one among many internal clustering evaluation measures. Many measures are discussed by Vendramin et al. [2010].
- Although k-means is a time-honored, classic method, there is still a lot of research around it, for example to make it faster. Elkan [2003] proposes a speed-up based on properties of metric spaces.
- On the other hand, comparing runtimes of algorithms empirically is a bit tricky. Kriegel et al. [2017] discuss issues around that.

Summary

You learned in this section:

- *What is Clustering?*
- *Why do we need heuristic approaches to "solve" the clustering problem?*
- *Basic heuristic ideas for identifying "partitions" into k clusters*
  - *selection of representative points*
  - *optimization approaches for assignment of points to representatives:*
    - *minimization of variance [Forgy, 1965, Lloyd, 1982]*
    - *k-means [MacQueen, 1967]*
    - *k-medoids*
    - *k-modes*
  - *common ideas and differences between these approaches*
  - *evaluation, model selection, silhouette-coefficient*

Outline

Outline

Recommended Reading

Recommended Reading:

- *Mitchell [1997], Chapter 1*

Classification – Supervised Learning

- For some domain $\mathcal{D}$ and a set of classes $C = \{c_1, \ldots, c_k\}$, $k \geq 2$, each $o \in \mathcal{D}$ belongs uniquely to some $c \in C$, i.e., there is a function $f : \mathcal{D} \to C$ (see Slide 55).
- Given a set of objects $O = \{o_1, o_2, \ldots, o_n\} \subseteq \mathcal{D}$ and a mapping $(O \to C) \subset f$ (examples):
  We want to also map any object $o_m \in \mathcal{D} \setminus O$ to C.
- Supervised vs. unsupervised:
  - In clustering, we don't have any information on C.
  - In classification, we have examples (a training set) to guide (supervise) the learning process.

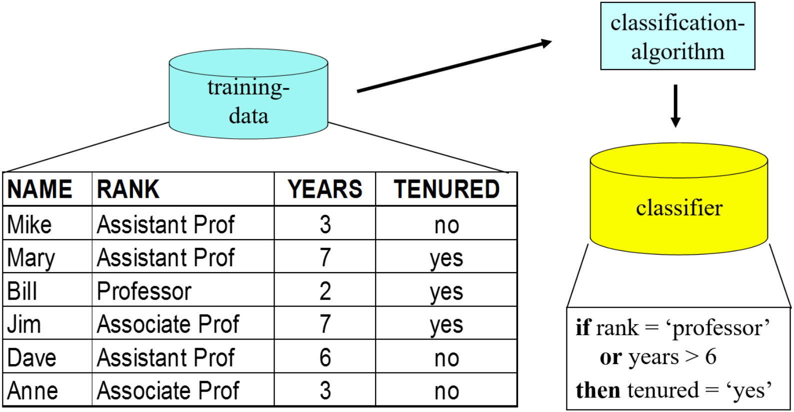
Classification – Supervised Learning

- A classifier is trained on some training set $TR \subseteq O$ to learn the mapping function (a model or hypothesis) $h : \mathcal{D} \to C$.
- Ideally we have $\forall o \in TR : h(o) = f(o)$ (if not for all, we should have this at least for most examples o).
- After training, the classifier should also work on $\mathcal{D} \setminus TR$ and predict the correct class, i.e., $h \approx f$.
- f is called "the target function".

Assumption 5.1 (The inductive learning assumption)

Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other examples.

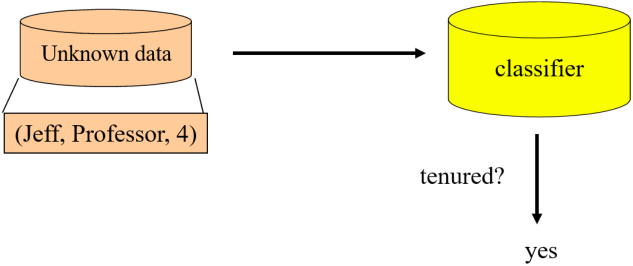
Training of a Classifier



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

if rank = 'professor'
or years > 6
then tenured = 'yes'

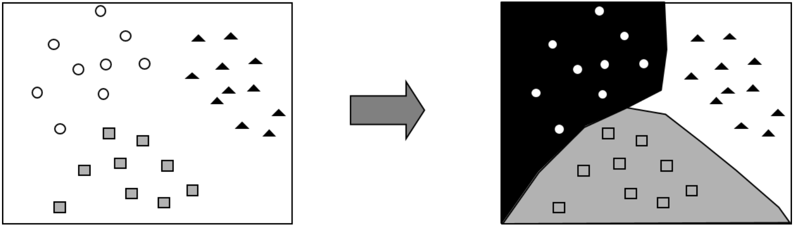
Application of a Classifier



Intuition

- Data $\subset \mathcal{D} \times C$ are generated by some (natural / technical / social / ...) statistical process.
- The observed data are examples for the effect of the process.
- The challenge for the learning algorithm is to generate a model to explain the process.
- h is an approximation of f, a hypothesis to explain the data.
- In the ideal case, the hypothesis h is interpretable and helps to understand the data-generating process.
- Pragmatically, h might be useful for predictions although it might not be interpretable.

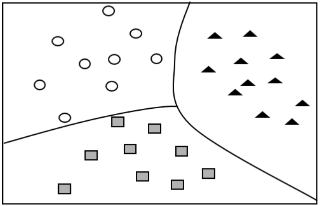
Classification Algorithms and their Hypothesis-Space

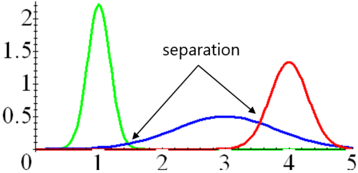


- Based on training data, a classifier typically provides a hypothesis that separates the examples belonging to different classes from each other.
- Each classification algorithm comes with (more or less strict) assumptions on how separation can be achieved or defined.

Classification Algorithms and their Hypothesis-Space

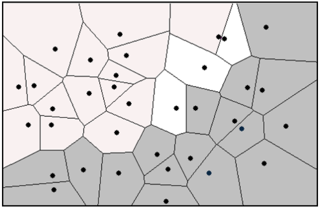
- These assumptions (implicitly) define a hypothesis space $\mathcal{H}$, the space of hypotheses that could possibly be learned by the given algorithm.





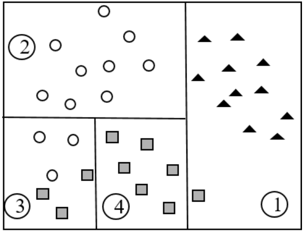
Classification Algorithms and their Hypothesis-Space

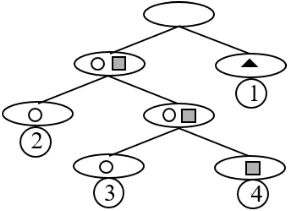
- These assumptions (implicitly) define a hypothesis space $\mathcal{H}$, the space of hypotheses that could possibly be learned by the given algorithm.



Classification Algorithms and their Hypothesis-Space

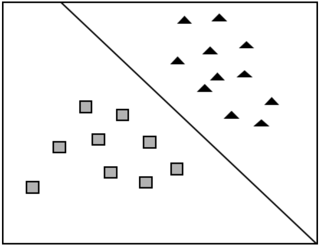
- These assumptions (implicitly) define a hypothesis space $\mathcal{H}$, the space of hypotheses that could possibly be learned by the given algorithm.





Classification Algorithms and their Hypothesis-Space

- These assumptions (implicitly) define a hypothesis space $\mathcal{H}$, the space of hypotheses that could possibly be learned by the given algorithm.



Outline

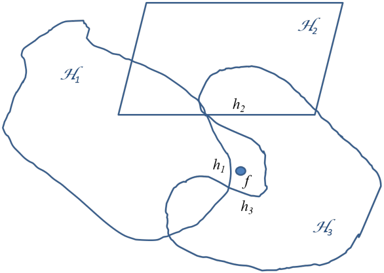
Recommended Reading

Recommended Reading:

- *Mitchell [1997], Chapter 2*

Hypothesis Space and Bias

- The hypothesis space is a manifestation of the so-called "bias".
- Some algorithm, due to the restrictions of its hypothesis space, will prefer certain hypotheses, i.e., the algorithm has a bias.



- In general, we do not know if $f \in \mathcal{H}$, i.e., whether or not we can actually approximate f (well enough) with some (or any) specific classification algorithm.
- Note that a hypothesis space can be infinite and yet restricted.

Example: Some Concept of Enjoying Sport

Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

$f: \text{Sky} \times \text{AirTemp} \times \text{Humidity} \times \text{Wind} \times \text{Water} \times \text{Forecast} \rightarrow \{\text{Yes,No}\}$

- Is there a general concept when to enjoy sport?
- To learn a concept from the example data, we need some assumption on this concept.
- Here we can assume, for example, that the concept is a conjunction of some attribute values.
- We thus have a finite hypothesis space, containing all possible conjunctions of concrete attribute values.

Coverage of Data Instances by Hypotheses

For this toy example, we define:

Example An example or instance x is a possible day, described as $x \in$ Sky $\times$ AirTemp $\times$ Humidity $\times$ Wind $\times$ Water $\times$ Forecast $= \mathcal{D}$

Positive/negative example An example x is positive, if $f(x) =$ Yes, negative otherwise. (Note that we know f on the training data only.)

Hypothesis A hypothesis defines a subset of $\mathcal{D}$. We write a hypothesis as vector containing

- specific values for attributes
- wildcards ('?') indicating that for some attribute any attribute value is acceptable
- $\emptyset$ indicating that no attribute value is acceptable.

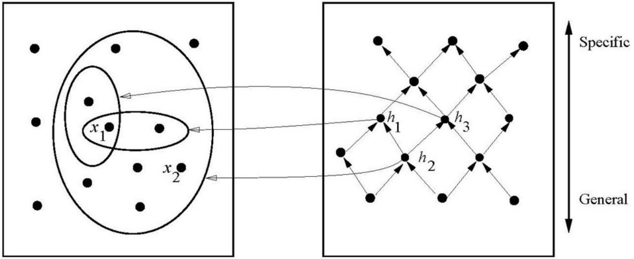
Coverage of Data Instances by Hypotheses

- For example the hypothesis $\langle$Sunny$, ?, ?, ?,$ Cool$, ?\rangle$ defines all examples where Sky=Sunny and Water=Cool.
- If we are interested in hypotheses about positive examples, we can interpret $\langle$Sunny$, ?, ?, ?,$ Cool$, ?\rangle$ as a rule:

```
if Sky=Sunny and Water=Cool
then EnjoySport=Yes
```

- An example x *satisfies* a hypothesis h, if and only if $f(x) =$ Yes.

The Assumptions of a Learning Algorithm Define the Hypothesis Space



$x_1 = \langle \text{Sunny}, \text{Warm}, \text{High}, \text{Strong}, \text{Cool}, \text{Same} \rangle$
$x_2 = \langle \text{Sunny}, \text{Warm}, \text{High}, \text{Light}, \text{Warm}, \text{Same} \rangle$

$h_1 = \langle \text{Sunny}, ?, ?, \text{Strong}, ?, ? \rangle$
$h_2 = \langle \text{Sunny}, ?, ?, ?, ?, ? \rangle$
$h_3 = \langle \text{Sunny}, ?, ?, ?, \text{Cool}, ? \rangle$

Definition 5.1

For any two hypotheses, h_j and h_k, over X, h_j is *more general than or equal to* h_k if and only if:

$$\forall x \in X : h_k(x) = \text{Yes} \Rightarrow h_j(x) = \text{Yes}$$

Basic Algorithm

Algorithm 5.1 (Find-S [Mitchell, 1997])

1. *Initialize h to the most specific hypothesis in $\mathcal{H}$*
2. *For each positive training instance x*
   For each attribute constraint a_i in h
   If the constraint a_i is satisfied by x
   Then do nothing
   Else replace a_i in h by the next more general constraint that is satisfied by x
3. *output hypothesis h*

Discussion:

- Finds the most specific hypothesis consistent with the positive examples in the training data – is it the only consistent hypothesis?
- Why should we prefer more specific hypotheses over more general ones?

Possible Hypotheses Under the Assumption of a Conjunctive Concept

Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

The following conjunctive hypotheses describe the concept “Yes” correctly for the training data:

$\langle \text{Sunny}, ?, ?, ?, ?, ? \rangle$ $\langle \text{Sunny}, ?, ?, \text{Strong}, ?, ? \rangle$

$\langle ?, \text{Warm}, ?, ?, ?, ? \rangle$ $\langle ?, \text{Warm}, ?, \text{Strong}, ?, ? \rangle$

$\langle \text{Sunny}, \text{Warm}, ?, ?, ?, ? \rangle$ $\langle \text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ? \rangle$

Reduce the Bias?

For some other training data, our model assumptions seem too strict:

Sky	**AirTemp**	**Humidity**	**Wind**	**Water**	**Forecast**	**EnjoySport**
Sunny	Warm	Normal	Strong	Cool	Change	Yes
Cloudy	Warm	Normal	Strong	Cool	Change	Yes
Rainy	Warm	Normal	Strong	Cool	Change	No

- No consistent hypothesis is possible under our assumptions:
- the most specific hypothesis for positive examples is $\langle ?, \text{Warm}, \text{Normal}, \text{Strong}, \text{Cool}, \text{Change} \rangle$
- this hypothesis covers also the negative example

Bias-free Learning

- Should we allow for more complex/generous assumptions to reduce the bias (i.e., to get rid of restrictions of the hypothesis space)?
- Allow disjunctions? Negations?
- A disjunctive hypothesis
  "if Sky=Sunny *or* Sky=Cloudy, then Yes"
  can list all positive examples.
- We could actually cover *any* concept with an arbitrarily complex hypothesis.
- Is this what we want?

Bias-free Learning is Futile

- Being able to cover *any* concept means, we don't rely on any assumption regarding possible concepts.
- The most complex hypothesis would describe the training data perfectly well: we can learn by heart all the examples.
- Learning by heart means we can perfectly predict all classes correctly for the *training examples* but we don't have any concept to *generalize to unseen data*.
- In general, a bias is necessary to avoid learning by heart.

Outline

Recommended Reading

Recommended Reading:

Best coverage related to our course:

- *Witten et al. [2011], Chapter 5*

The corresponding parts in other textbooks are either rather short or contain more advanced concepts as well:

- *Zaki and Meira Jr. [2020], Chapter 22*
- *Tan et al. [2006], Chapter 4.5 (4.6)*
- *Tan et al. [2020], Chapter 3.6*
- *Mitchell [1997], Chapter 5*

The Problem of Evaluation

- A database $\mathcal{D}$ is representing a domain by the sample of available data, there is a function $f : \mathcal{D} \to C$, mapping each object to a class $c_i \in C$.
- $O \subseteq \mathcal{D}$ is the set of objects where we know about the class (i.e., we know $f(o)$ for all $o \in O$, but not for any $o \in \mathcal{D} \setminus O$).
- Let h be a classifier (model, hypothesis), trained on a training set $TR \subseteq O$.

Problem:

- We want a good quality (performance, approximation of the target function f) of h over $\mathcal{D}$, yet we cannot know anything about the quality of h over $\mathcal{D} \setminus O$.

The Problem of Evaluation

- h has been optimized for TR.
- If we test h on TR, we will typically get an optimistic estimate of the performance over $\mathcal{D}$.
- The phenomenon that h performs better on TR than on $\mathcal{D}$ overall is called *overfitting*:
  - Often, the amount of training data is not sufficient to generalize reliably ("small sample size bias").
  - If the data sample is too small to truly represent the domain, there is no sufficient ground to reject overspecialized hypotheses.
  - In general, the weaker the bias of a learning algorithm, the more susceptible is it to overfitting.

The Separation of Training and Testing Data

To get a more realistic estimate of the performance of some classifier over $\mathcal{D}$, we separate training data $TR \subset O$ and test data $TE \subset O$, $TR \cap TE = \emptyset$.

- TR is used to train the classifier (fit the model, select the hypothesis).
- TE is used to evaluate the classifier.
- The purpose is to estimate the performance (success-/error-rate) of the classifier. Therefore:
  - TR and TE need to be independent ($TR \cap TE = \emptyset$).
  - Both, TR and TE should represent the classification problem as good as possible.
  - For some learning problems (benchmark datasets), a separation into TR and TE is available.
- Problem: If it is not clear whether O is already too small to allow for a good generalization, we do not really want TR to be even smaller.

m-fold Cross-Validation

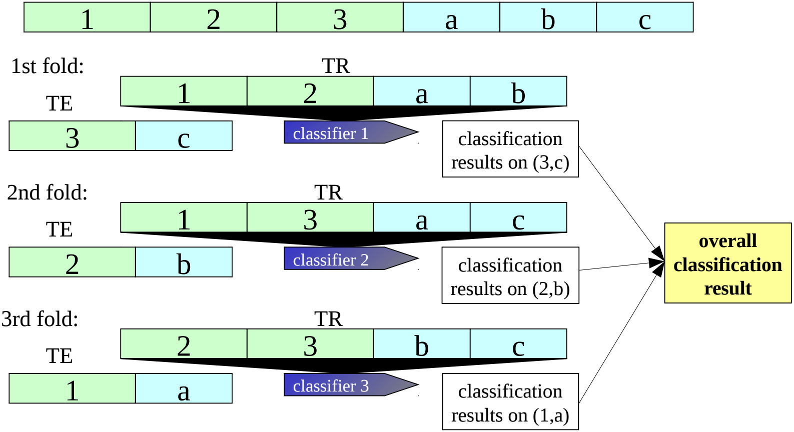
Motivation: We want to use as much data as possible for training, and as much as possible for testing.

Algorithm 5.2 (m-fold cross-validation)

1. *Separate the set O in m equal-size, mutually disjoint subsets.*
2. *Get m different pairs of TR and TE by using each of the m subsets as TE once and the remaining $m-1$ subsets for training.*
3. *On these m pairs of TR and TE, train and test m independent classifiers.*
4. *Average the m observed performances.*
5. *Repeat 1-4 several times.*

m-fold Cross-Validation

For one iteration with $m = 3$, we have T_1 with class information a, T_2 with class information b, and T_3 with class information c:



Stratification

Stratification aims at representing the class proportions in each fold.

- minimum requirement: each class should be present in the training set.
- stratified cross-validation: the distribution of classes within each training and test set should reflect the distribution of the classes in O

Standard approach (rule of thumb):
10-fold, stratified cross-validation, repeated 10 times

Note that:

The evaluation procedure has the purpose of estimating the performance on $\mathcal{D} \setminus O$. In order to get the best possible classifier, we would use all available labeled data (O) for training.

The Bootstrap

In the bootstrap procedure, a training set is created from O by drawing with replacement:

- We take $|O|$ objects from O, where the same object could be drawn several times.
- A sample TR contains on average 63% of the objects in O. Some are present several times in TR, some ($\approx 37\%$) are not present at all:
  - For $|O| = n$, an individual object in O has a chance of being drawn of $\frac{1}{n}$ each turn, that is, it is *not* drawn with probability $1 - \frac{1}{n}$.
  - After n draws, a specific object has not been drawn with probability $\left(1 - \frac{1}{n}\right)^n$
  - For large n: $\left(1 - \frac{1}{n}\right)^n \approx e^{-1} \approx 0.368$, hence this procedure is also called "the 0.632 bootstrap"

Leave-one-out or Jackknife Test

Algorithm 5.3 (Leave-one-out)

- *$\forall o_j \in O$: take o_j as test object for a classifier trained on $O \setminus \{o_j\}$.*
- *average the performance estimate over all test objects*

Discussion:

- For $|O| = n$, this is an n-fold cross-validation.
- Pro: no random effect
- Con: stratification is not possible
- In general, the Jackknife test leads to a relatively pessimistic performance estimate.

Confusion Matrix

The confusion matrix represents the number of correctly and incorrectly predicted classes per actual and per predicted class:

predicted class

actual class

	class 1	class 2	class 3	class 4	class 5
class 1	35	1	1	1	4
class 2	0	31	1	1	5
class 3	3	1	50	1	2
class 4	1	0	1	10	2
class 5	3	1	9	15	13

correctly predicted objects

Quality Measures for Classifiers

Given a classifier h, a training set $TR \subseteq O$, and a test set $TE \subseteq O$. $f(o)$ is the actual class of o, $h(o)$ is the class predicted by the classifier h. Then we have:

accuracy of h on TE:

$$\mathrm{acc}_{TE}(h) = \frac{|\{o \in TE | h(o) = f(o)\}|}{|TE|}$$

true classification error:

$$\mathrm{err}_{TE}(h) = \frac{|\{o \in TE | h(o) \neq f(o)\}|}{|TE|}$$

apparent classification error:

$$\mathrm{err}_{TR}(h) = \frac{|\{o \in TR | h(o) \neq f(o)\}|}{|TR|}$$

Focus on Individual Classes

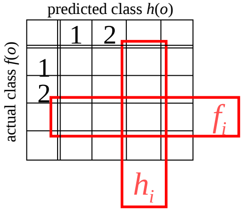
If we focus on a single class (the "positive" class) vs. all other classes (the "negative" class), the confusion matrix can be read as follows:

	predicted positive	predicted negative
given positive	TP (true positive)	FN (false negative)
given negative	FP (false positive)	TN (true negative)

This notation is also often used in two-class problems, where we have a particular interest to detect cases of the "positive" class, e.g., medical tests on specific diseases.

Quality Measures for Individual Classes

recall: proportion of test objects of some class c_i that have been predicted correctly



$$f_i = \{o \in TE | f(o) = c_i\} :$$

$$\text{recall}_{TE}(h, i) = \frac{|\{o \in f_i | h(o) = f(o)\}|}{|f_i|}$$

precision: proportion of test objects predicted as class c_i that actually belong to class c_i

$$h_i = \{o \in TE | h(o) = c_i\} :$$

$$\text{precision}_{TE}(h, i) = \frac{|\{o \in h_i | h(o) = f(o)\}|}{|h_i|}$$

Outline

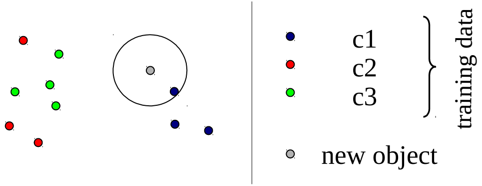
Recommended Reading

Recommended Reading:

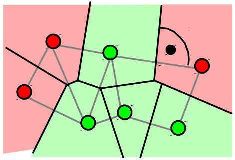
- *Tan et al. [2006], Chapter 5.2*
- *Tan et al. [2020], Chapter 6.3*
- *Witten et al. [2011], Chapter 4.7*
- *Hastie et al. [2001], Chapter 2.3*
- *Mitchell [1997], Chapter 8*

Instance-based Learning

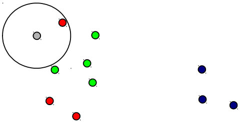
- a simple classifier: assign to a new object the class of the nearest neighbor in the training data



- we can visualize class regions by Voronoi cells



k-Nearest Neighbor Classification



- potential problem: nearest neighbor might be an outlier, somehow unusual, misleading
- take more than one neighbor into consideration: k nearest neighbor classifier

decision set: the set of (k) nearest neighbors considered for the classification decision

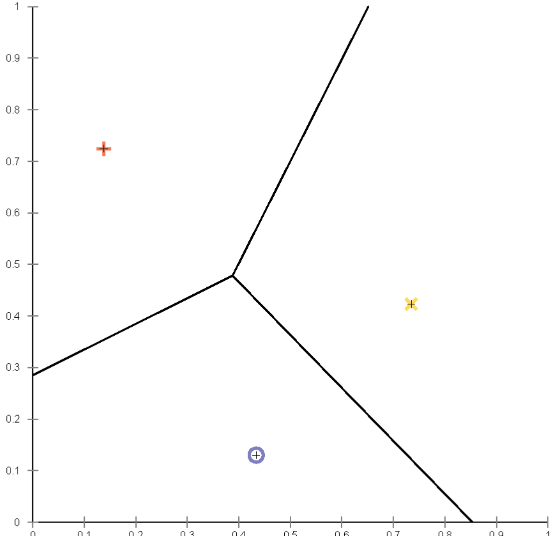
decision rule: how to decide the class, given the potentially different classes of the k nearest neighbors
  - take the majority vote
  - potentially weighted votes

Let $x_1, \ldots, x_k$ be the k nearest neighbors of instance x_q:

$$h(x_q) = \arg\max_{c \in C} \sum_{i=1}^{k} w_i \delta(c, f(x_i)) \text{ where } \delta(a,b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ otherwise} \end{cases}$$

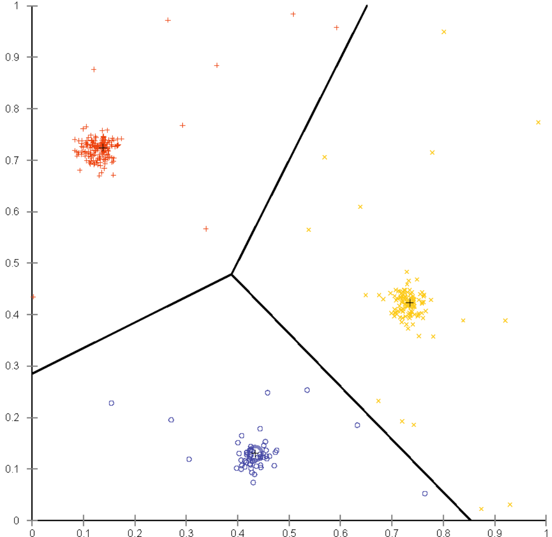
“Training” and Prediction for kNN Classifier

small training set: simple decision boundaries



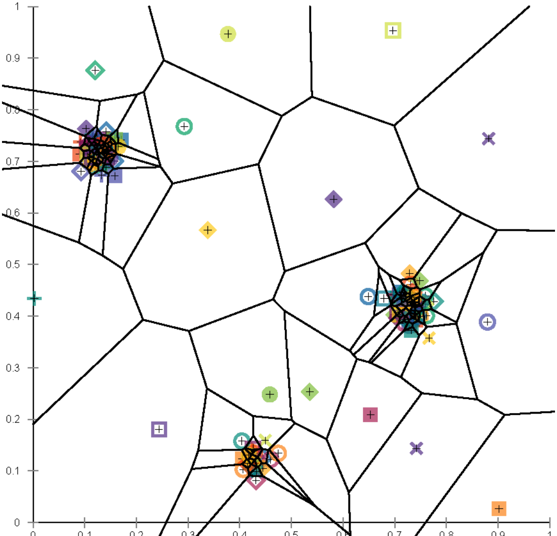
“Training” and Prediction for kNN Classifier

small training set: simple decision boundaries



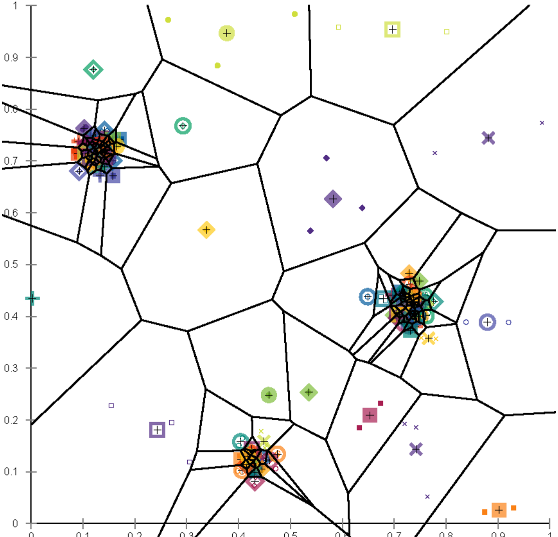
“Training” and Prediction for kNN Classifier

large training set: potentially complex decision boundaries



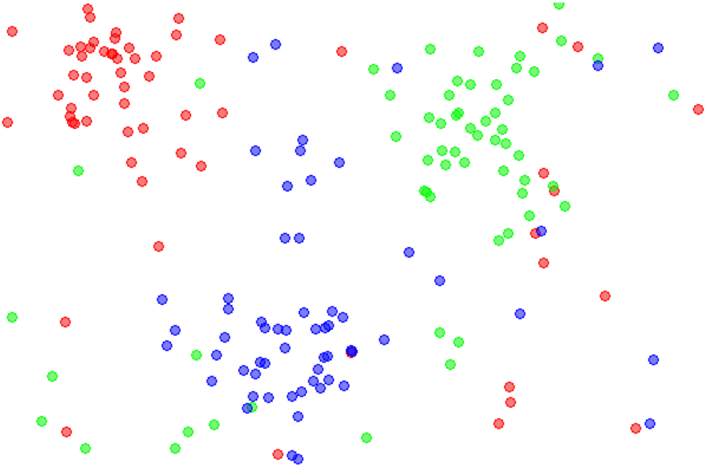
“Training” and Prediction for kNN Classifier

large training set: potentially complex decision boundaries



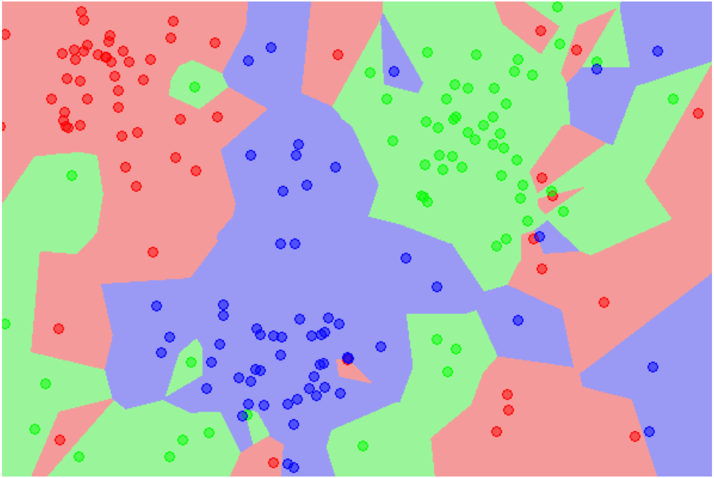
Complex Class Boundaries with Larger k

dataset



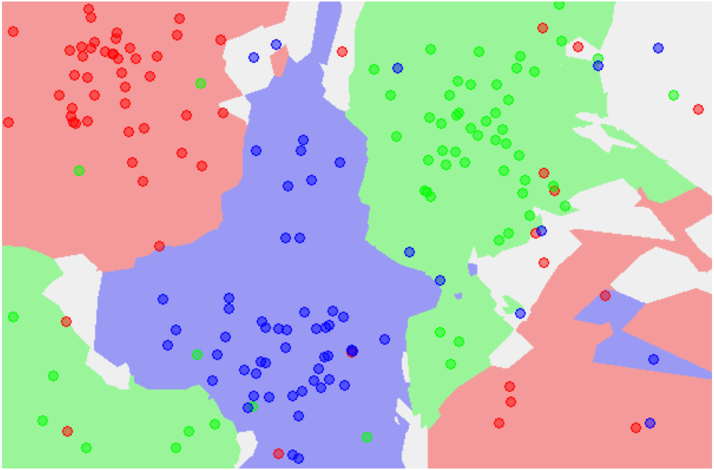
Complex Class Boundaries with Larger k

$k = 1$

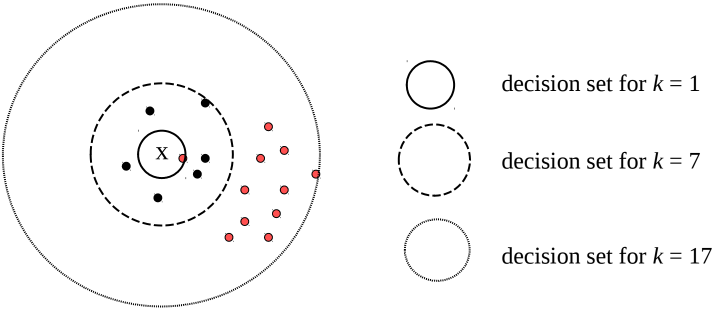


Complex Class Boundaries with Larger k

$k = 5$



Decision Set



- k too small: classifier is sensitive to outliers
- k too large: potentially takes objects belonging to other classes into the decision set
- medium k: best quality
- rule of thumb: $1 \ll k \leq 10$, but consider, e.g., size of classes in training set

Decision Rule

- standard: choose majority class in decision set
- advanced: put weights on the class votes
  - by distance, typically squared inverted distance:

$$\text{weight}(\text{dist}) = \frac{1}{\text{dist}^2}$$

  - by class proportions:
    If a class is small, it has a smaller chance of being the majority in some decision set.

Example: 2 classes, 95% A, 5% B, the labels of some decision set (e.g., labels of the 7 nearest neighbors of x) are $\{A, A, A, A, B, B, B\}$

- standard decision rule: $h(x) = A$
- votes weighted based on class size: $h(x) = B$

Instance-based Learning

- Instance-based learning does not provide an explicit description of the target function.
- Training examples are simply stored.
- Generalization beyond the training examples is postponed until a new instance must be classified (“lazy learner”).
- High flexibility (low bias) because the target function is actually estimated locally and differently for each new instance instead of once for the entire instance space.

Discussion

pros

- easy to apply: requires only training data and distance function
- often good classification accuracy
- incremental: easy adaptation to new training data
- no training required (“lazy learner”)

cons

- inefficient prediction: each decision requires k nearest neighbor query
- does not deliver explicit knowledge about classes
- difficult to choose k, esp. if classes are of very different size

Outline

Summary

You learned in this section:

- *What is Classification?*
- *hypothesis-space and bias*
- *Why is a bias unavoidable for learning and generalization?*
- *evaluation procedures (cross-validation, bootstrap, leave-one-out)*
- *quality measures for classifiers:*
  - *confusion matrix*
  - *accuracy & error (apparent vs. true)*
  - *precision & recall*
- *a simple classifier: k-nearest neighbors*
  - *properties, challenges, variants*
  - *lazy learning*

Outline

Can we Formalize “Confidence”?

We encountered a measure called “confidence” that should tell us how reliable a discovered association rule is.
We interpreted:

- The higher the confidence for some rule ‘$X \Rightarrow Y$’, the more likely Y is present in transactions that contain X.
- The confidence is an estimate of the conditional probability of Y given X.

Can we formalize this interpretation?

Recommended Reading:

Mitzenmacher and Upfal [2017], Chapter 1.

Outline

Sample Space

The sample space Ω is the set of all (disjoint) possible outcomes of some random process.

Examples:

- *If we role a dice, we have* $\Omega = \{1, 2, 3, 4, 5, 6\}$.
- *If we flip a coin, we have* $\Omega = \{H, T\}$.

Events

A subset $E \subseteq \Omega$ of individual outcomes of a random process can define an "event".

Examples:

- *We role a die. Every element of $\Omega = \{1, 2, 3, 4, 5, 6\}$ is a simple or elementary event.*
- *We could be interested in the event "The die shows an even number" $= \{2, 4, 6\} \subseteq \Omega$.*
- *We flip a coin. We could have the elementary event "head" $\subseteq \Omega$.*

A family of sets $\mathcal{F}$ represents the allowable events. Each set in $\mathcal{F}$ is a subset of Ω, i.e., $\mathcal{F} \subseteq \wp(\Omega)$.

Probability Function

Definition 6.1 (Probability Function)

A probability function is any function $\Pr : \mathcal{F} \to \mathbb{R}$ that satisfies the following conditions:

1. $\forall E : 0 \leq \Pr(E) \leq 1$;
2. $\Pr(\Omega) = 1$; and
3. for any finite or countably infinite sequence of pairwise mutually disjoint events $E_1, E_2, E_3, \ldots$:

$$\Pr\left(\bigcup_{i \geq 1} E_i\right) = \sum_{i \geq 1} \Pr(E_i).$$

Probability Space

Definition 6.2 (Probability Space)

A probability space is given by three components:

1. a sample space Ω;
2. the allowable events $\mathcal{F} \subseteq \wp(\Omega)$; and
3. a probability function $\Pr : \mathcal{F} \rightarrow \mathbb{R}$.

Event Combinations

Because events are sets, we can use standard set theory notation to express combinations of events.

- $E_1 \cap E_2$ denotes the occurrence of both, E_1 and E_2 (i.e., their co-occurrence).
- $E_1 \cup E_2$ denotes the occurrence of either E_1 or E_2 (or both).
- $E_1 \setminus E_2$ denotes the occurrence of event E_1 without E_2 occurring as well.
- $\overline{E} = \Omega \setminus E$ denotes the complementary event of E.

Event Combinations

Examples:

Suppose we roll two dice. Given events E_1 and E_2:

E_1 *the first die is a 1*

E_2 *the second die is a 1*

- $E_1 \cap E_2$: *both dice are 1*
- $E_1 \cup E_2$: *at least one of the dice lands on 1.*
- $E_1 \setminus E_2$: *the first die is a 1 and the second die is not.*

Event Combinations

Examples:

Let E be the event that by rolling a die we obtain an even number.

- *Then $\overline{E}$ is the event that we obtain an odd number.*
- *What are the events $\overline{E_1}$, $\overline{E_1 \cup E_2}$, $\overline{E_1 \cap E_2}$?*

Example: One Die

Consider the random process defined by the outcome of rolling a die:

$$\Omega_{\text{die}_1} = \{1, 2, 3, 4, 5, 6\}$$

Assuming a fair die, all sides have equal probability, thus:

$$\Pr(\{1\}) = \Pr(\{2\}) = \ldots = \Pr(\{6\}) = \frac{1}{6}$$

The probability of the event "odd number" is

$$\Pr(\{1, 3, 5\}) = \Pr(\{1\}) + \Pr(\{3\}) + \Pr(\{5\}) = \frac{1}{2}$$

Example: Two Dice

Consider the random process defined by the outcome of rolling two (fair) dice:

$$\Omega = \Omega_{\text{die}_1} \times \Omega_{\text{die}_2} = \{(i,j) | 1 \leq i,j \leq 6\}$$

Each (ordered) combination has a probability of $\frac{1}{36}$.

Example:

Probability of the event "sum $= 2$":

$$\Pr(\{(1,1)\}) = \frac{1}{36}$$

Example: Two Dice

Consider the random process defined by the outcome of rolling two (fair) dice:

$$\Omega = \Omega_{\text{die}_1} \times \Omega_{\text{die}_2} = \{(i,j) | 1 \leq i,j \leq 6\}$$

Each (ordered) combination has a probability of $\frac{1}{36}$.

Example:

Probability of the event "sum $= 3$":

$$\Pr(\{(1,2),(2,1)\}) = \Pr(\{(1,2)\}) + \Pr(\{(2,1)\}) = \frac{2}{36} = \frac{1}{18}$$

Example: Two Dice

Consider the random process defined by the outcome of rolling two (fair) dice:

$$\Omega = \Omega_{\text{die}_1} \times \Omega_{\text{die}_2} = \{(i,j) | 1 \leq i,j \leq 6\}$$

Each (ordered) combination has a probability of $\frac{1}{36}$.

Example:

Probability of the event E_1 =“sum bounded by 6”:

$$\begin{aligned} E_1 = & \{(1,1),(1,2),(1,3),(1,4),(1,5),(2,1),(2,2), \\ & (2,3),(2,4),(3,1),(3,2),(3,3),(4,1),(4,2),(5,1)\} \end{aligned}$$

$$\Pr(E_1) = \frac{15}{36}$$

Example: Two Dice

Consider the random process defined by the outcome of rolling two (fair) dice:

$$\Omega = \Omega_{\text{die}_1} \times \Omega_{\text{die}_2} = \{(i,j) | 1 \leq i,j \leq 6\}$$

Each (ordered) combination has a probability of $\frac{1}{36}$.

Example:

E_2 = *"both dice have odd numbers":*

$$E_2 = \{(1,1),(1,3),(1,5),(3,1),(3,3),(3,5),(5,1),(5,3),(5,5)\}$$

$$\Pr(E_2) = \frac{1}{4}$$

Example: Two Dice

Consider the random process defined by the outcome of rolling two (fair) dice:

$$\Omega = \Omega_{\text{die}_1} \times \Omega_{\text{die}_2} = \{(i,j) | 1 \leq i,j \leq 6\}$$

Each (ordered) combination has a probability of $\frac{1}{36}$.

Example:

E_3 = *"sum bounded by 6 and both dice have odd numbers":*

$$\begin{aligned} \Pr(E_3) &= \Pr(E_1 \cap E_2) \\ &= \Pr(\{(1,1),(1,3),(1,5),(3,1),(3,3),(5,1)\}) \\ &= \frac{1}{6} \end{aligned}$$

Combined Probability

Lemma 6.3 (Combined Probability)

For any two events E_1 and E_2:

$$\Pr(E_1 \cup E_2) = \Pr(E_1) + \Pr(E_2) - \Pr(E_1 \cap E_2)$$

Proof.

$$\begin{aligned}
\Pr(E_1) &= \Pr(E_1 \setminus (E_1 \cap E_2)) + \Pr(E_1 \cap E_2) \\
\Pr(E_2) &= \Pr(E_2 \setminus (E_1 \cap E_2)) + \Pr(E_1 \cap E_2) \\
\Pr(E_1 \cup E_2) &= \Pr(E_1 \setminus (E_1 \cap E_2)) \\
&\quad + \Pr(E_2 \setminus (E_1 \cap E_2)) \\
&\quad + \Pr(E_1 \cap E_2)
\end{aligned}$$

$\square$

Union Bound

Lemma 6.4 (Union Bound)

For any finite or countably infinite sequence of events $E_1, E_2, E_3, \ldots$*:*

$$\Pr\left(\bigcup_{i\geq 1} E_i\right) \leq \sum_{i\geq 1} \Pr(E_i).$$

Difference from condition 3 in Definition 6.1?

Outline

Independent Events

Definition 6.5 (Independent Events)

Two events E and F are *independent* if and only if

$$\Pr(E \cap F) = \Pr(E) \cdot \Pr(F).$$

More generally, events $E_1, E_2, \ldots, E_k$ are mutually independent if and only if

$$\forall I \subseteq [1, k] : \Pr\left(\bigcap_{i \in I} E_i\right) = \prod_{i \in I} \Pr(E_i).$$

Independent Events: Intuition

- If events A and B are *independent* then knowledge about event A does not change the probability of B.
- If A and B are *not independent*, then we can quantify the conditional probability of A subject to our kowledge of event B.

Example:

Probability of the event E_1 "outcome of a die roll is even": $\frac{3}{6}$.
Probability of the event E_2 "the outcome is ≤ 4": $\frac{4}{6}$.
Probability of $E_1 \cap E_2$: "the outcome is even and is ≤ 4":

$$\Pr(E_1 \cap E_2) = \frac{2}{6} = \frac{12}{36} = \frac{3}{6} \cdot \frac{4}{6} = \Pr(E_1) \cdot \Pr(E_2)$$

$\Rightarrow$ *The two events are independent.*

Independent Events: Intuition

- If events A and B are *independent* then knowledge about event A does not change the probability of B.
- If A and B are *not independent*, then we can quantify the conditional probability of A subject to our kowledge of event B.

Example:

Probability of the event E_1 "outcome of a die roll is even": $\frac{3}{6}$.
Probability of the event E_2 "the outcome is ≤ 3": $\frac{3}{6}$.
Probability of $E_1 \cap E_2$: "the outcome is even and is ≤ 3":

$$\Pr(E_1 \cap E_2) = \frac{1}{6} = \frac{6}{36} \neq \frac{9}{36} = \frac{3}{6} \cdot \frac{3}{6} = \Pr(E_1) \cdot \Pr(E_2)$$

$\Rightarrow$ *The two events are not independent.*

Conditional Probability

Definition 6.6 (Conditional Probability)

The *conditional probability* that event E occurs given that event F occurs is

$$\Pr(E|F) = \frac{\Pr(E \cap F)}{\Pr(F)}$$

The conditional probability is well-defined only if $\Pr(F) > 0$.

Note that:

If E and F are independent and $\Pr(F) \neq 0$, we have:

$$\Pr(E|F) = \frac{\Pr(E \cap F)}{\Pr(F)} = \frac{\Pr(E)\Pr(F)}{\Pr(F)} = \Pr(E)$$

Conditional Probability: Intuition

- We look for the probability of $E \cap F$ within the sets of events defined by F.
- Because F restricts the sample space, we normalize the probabilities by dividing by $\Pr(F)$.
- If E and F are independent, information about F should not affect the probability of E.

Example:

Probability of the event E_1 "outcome of a die roll is even": $\frac{3}{6}$.
Probability of the event E_2 "the outcome is ≤ 4": $\frac{4}{6}$.
Probability of $E_1 \cap E_2$: "the outcome is even and is ≤ 4":

$$\Pr(E_1 \cap E_2) = \frac{2}{6} = \frac{12}{36} = \frac{3}{6} \cdot \frac{4}{6} = \Pr(E_1) \cdot \Pr(E_2)$$

$\Rightarrow$ *The two events are independent.*

Conditional Probability: Intuition

- We look for the probability of $E \cap F$ within the sets of events defined by F.
- Because F restricts the sample space, we normalize the probabilities by dividing by $\Pr(F)$.
- If E and F are independent, information about F should not affect the probability of E.

Example:

Probability of the event E_1 "outcome of a die roll is even": $\frac{3}{6}$.
Probability of the event E_2 "the outcome is ≤ 3": $\frac{3}{6}$.
Probability of $E_1 \cap E_2$: "the outcome is even and is ≤ 3":

$$\Pr(E_1 \cap E_2) = \frac{1}{6} = \frac{6}{36} \neq \frac{9}{36} = \frac{3}{6} \cdot \frac{3}{6} = \Pr(E_1) \cdot \Pr(E_2)$$

$\Rightarrow$ *The two events are not independent.*

The Condition Defines a Probability Space

$\Pr(X|E)$ defines a proper probability function on the sample space E (cf. Definitions 6.1 and 6.2):

$$\Pr(\emptyset|E) = \frac{\Pr(\emptyset \cap E)}{\Pr(E)} = \frac{\Pr(\emptyset)}{\Pr(E)} = 0$$

$$\Pr(E|E) = \frac{\Pr(E \cap E)}{\Pr(E)} = \frac{\Pr(E)}{\Pr(E)} = 1$$

For any two disjoint events A and B:

$$\begin{aligned}\Pr(A \cup B|E) &= \frac{\Pr((A \cup B) \cap E)}{\Pr(E)} \\ &= \frac{\Pr(A \cap E) + \Pr(B \cap E)}{\Pr(E)} \\ &= \Pr(A|E) + \Pr(B|E)\end{aligned}$$

Outline

Quality Measures for Association Rules

Support: $s(X \Rightarrow Y) = s(X \cup Y)$

or in relative terms: frequency $f(X \cup Y) = \frac{s(X \cup Y)}{|\mathcal{D}|}$

Confidence: $\mathrm{conf}(X \Rightarrow Y) = \frac{s(X \cup Y)}{s(X)}$

Lift: $\mathit{Lift}(X \Rightarrow Y) = \frac{\mathrm{conf}(X \Rightarrow Y)}{f(Y)}$

Jaccard: $\mathit{Jaccard}(X \Rightarrow Y) = \frac{s(X \cup Y)}{s(X) + s(Y) - s(X \cup Y)}$

conviction: $\mathit{conviction}(X \Rightarrow Y) = \frac{1 - f(Y)}{1 - \mathrm{conf}(X \Rightarrow Y)}$

Probabilistic Interpretation: Support (Frequency)

The frequency of an itemset in the database can be seen as an empirical estimate of its probability, given the sample represented by the database:

$$\Pr(X) = \frac{s(X)}{|\mathcal{D}|}$$

Note that:

$$\Pr(X \cap Y) = \frac{s(X \cup Y)}{|\mathcal{D}|}$$

Although X and Y are sets in both cases,

- *probabilistically, $\cap$ denotes the co-occurrence of events,*
- *while for itemsets, $\cup$ denotes that both itemsets need to be present.*

Probabilistic Interpretation: Confidence

The confidence is the conditional probability that a transaction contains the consequent Y given that it contains the antecedent X:

$$\begin{aligned}\mathrm{conf}(X \Rightarrow Y) &= \frac{s(X \cup Y)}{s(X)}\\ &= \frac{\Pr(X \cap Y)}{\Pr(X)}\\ &= \Pr(Y|X)\end{aligned}$$

- The confidence of a rule $X \Rightarrow Y$ is not a useful measure unless we compare it with the frequency of Y, i.e., the prior (unconditional) probability.
- If we have $\Pr(Y|X) < \Pr(Y)$ this means that in the presence of X, Y becomes less likely as it is unconditionally!
  (Not the rule, but this fact could be interesting, though!)

Probabilistic Interpretation: Lift

We can see Lift as normalization of the confidence by the prior probability of the consequent:

$$\mathit{Lift}(X \Rightarrow Y) = \frac{\mathrm{conf}(X \Rightarrow Y)}{f(Y)} = \frac{\Pr(X \cap Y)}{\Pr(X)\Pr(Y)}$$

- ratio of the observed joined probability of X and Y to the joint probability expected for statistically independent events (Definition 6.5).
- Lift is a (symmetric!) measure for the surprise of a rule.
- Values around 1: boring.
- Much smaller/larger values: interesting!

Probabilistic Interpretation: Jaccard

The Jaccard coefficient in general is a measure for the similarity between two sets:

$$\begin{aligned}\mathit{Jaccard}(X \Rightarrow Y) &= \frac{s(X \cup Y)}{s(X) + s(Y) - s(X \cup Y)} \\ &= \frac{\Pr(X \cap Y)}{\Pr(X) + \Pr(Y) - \Pr(X \cap Y)} \\ &= \frac{\Pr(X \cap Y)}{\Pr(X \cup Y)} \qquad \text{(Lemma 6.3)}\end{aligned}$$

- A symmetric measure of how often both, X and Y, occur simultaneously, relative to the occurrence of both or either overall.
- Similarity of the itemsets X and Y based on their individual occurrences and their co-occurrences.

Probabilistic Interpretation: Conviction

Conviction of a rule measures the expected error: how often does X occur in a transaction where Y does not?
(How often does the rule fail?)

$$
\begin{aligned}
\textit{conviction}(X \Rightarrow Y) &= \frac{1 - f(Y)}{1 - \operatorname{conf}(X \Rightarrow Y)} \\
&= \frac{\Pr\left(\overline{Y}\right)}{1 - \frac{\Pr(X \cap Y)}{\Pr(X)}} = \frac{\Pr(X)\Pr\left(\overline{Y}\right)}{\Pr(X) - \Pr(X \cap Y)} \\
&= \frac{\Pr(X)\Pr\left(\overline{Y}\right)}{\Pr\left(X \cap \overline{Y}\right)} = \frac{1}{\textit{Lift}\left(X \Rightarrow \overline{Y}\right)}
\end{aligned}
$$

- compares the observed joint probability of X and $\overline{Y}$ with their joint probability expected for independence
- asymmetric measure

Recommended Reading

Recommended Reading:

On the probabilistic interpretation of (even more) quality measures for association rules: Zaki and Meira Jr. [2014], Chapter 12.1.

Outline

The Law of Total Probability

Theorem 6.7 (The Law of Total Probability)

Let $E_1, E_2, \ldots, E_n$ be mutually disjoint events in the sample space Ω, and let $\bigcup_{i=1}^{n} E_i = \Omega$. Then

$$\Pr(B) = \sum_{i=1}^{n} \Pr(B \cap E_i) = \sum_{i=1}^{n} \Pr(B|E_i) \Pr(E_i).$$

The Law of Total Probability

Proof.

Since the events $E_i (i = 1, \ldots, n)$ are disjoint and cover the entire sample space Ω, it follows that

$$\Pr(B) = \sum_{i=1}^{n} \Pr(B \cap E_i).$$

Further, by Definition 6.6,

$$\sum_{i=1}^{n} \Pr(B \cap E_i) = \sum_{i=1}^{n} \Pr(B|E_i) \Pr(E_i).$$

$\square$

Bayes' Rule

Theorem 6.8 (Bayes' Rule)

Let $E_1, \ldots, E_n$ be mutually disjoint events, and let $\bigcup_{i=1}^{n} E_i = \Omega$. Then for any other event B, $\Pr(B) > 0$, $j = 1, \ldots, n$:

$$\Pr(E_j|B) = \frac{\Pr(E_j \cap B)}{\Pr(B)} \tag{6.1}$$

$$= \frac{\Pr(B|E_j)\Pr(E_j)}{\sum_{i=1}^{n} \Pr(B|E_i)\Pr(E_i)} \tag{6.2}$$

Proof.

From Eq. 1 to Eq. 2, we use Definition 6.6 in the numerator, and Theorem 6.7 in the denominator. □

Bayes' Rule (Simple Form)

In its simple form, we have only two events, A and B, $\Pr(A) \neq 0$, $\Pr(B) \neq 0$:

$$
\begin{aligned}
\Pr(A|B) &= \frac{\Pr(A \cap B)}{\Pr(B)} \\
\Pr(B|A) &= \frac{\Pr(B \cap A)}{\Pr(A)} \\
\Rightarrow \Pr(A \cap B) &= \Pr(A|B)\Pr(B) = \Pr(B|A)\Pr(A) \\
\Rightarrow \Pr(A|B) &= \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}
\end{aligned}
$$

We do not require exhaustiveness of A or B here (i.e., $A \subseteq \Omega$, $B \subseteq \Omega$), since we do not apply Theorem 6.7, only Definition 6.6.

Bayes' Rule: Example 1

- We are given three coins, two of the coins are fair and the third coin is biased, showing head with probability $\frac{2}{3}$. We need to identify the biased coin.
- We flip each of the coins. The first and second coins come up with head, the third coin comes up with tail.
- What is the probability that the first coin is the biased one?
- Let E_i be the event that the i-th coin is the biased one, and let B be the event that the three coin flips came up head, head, tail.
- Prior probability: $\Pr(E_i) = \frac{1}{3}$ for $i = 1, 2, 3$.

Bayes' Rule: Example 1

$$\Pr(B|E_1) = \Pr(B|E_2) = \frac{2}{3} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{6}$$

and

$$\Pr(B|E_3) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{3} = \frac{1}{12}.$$

Thus, according to Bayes' rule:

$$\Pr(E_1|B) = \frac{\Pr(B|E_1)\Pr(E_1)}{\sum_{i=1}^{3}\Pr(B|E_i)\Pr(E_i)} = \frac{2}{5}.$$

The experiment increases the probability that the first coin is the biased one from $\frac{1}{3}$ to $\frac{2}{5}$.

Bayes' Rule: Example 2

- A doctor sees a patient with fever and rash.
- 80% of patient with flu, 45% of allergy patients, and 90% of infection patients have these symptoms.
- The doctor knows that 50% of the patients she sees have flu, 40% have allergy, and 10% have an infection.
- Should the doctor treat the patient for infection?

Bayesian Reasoning: The General Pattern

- There are alternative models to explain a fact.
- Each model defines a probability for the observed data.
- Which model is the best (i.e., the most likely) explanation?
- $E_1, E_2, \ldots, E_n$ are the alternative models.
- B is the observed data.
- For each model we know $\Pr(B|E_j)$ (i.e., how well the model explains the facts).

$$\Pr(E_j|B) = \frac{\Pr(E_j \cap B)}{\Pr(B)} = \frac{\Pr(B|E_j)\Pr(E_j)}{\sum_{i=1}^{n} \Pr(B|E_i)\Pr(E_i)}$$

- Difficulty: How do we know $\Pr(E_j)$?

Bayesian Approach

- Start with a *prior* model, giving some initial value to the model parameters.
- This model is then modified by incorporating new observations, to obtain a *posterior* model that captures the new information.

Example:

- A test shows that a patient has an infection.
- The test has 10% error rate.
- What is the probability that the patient has an infection?

Bayesian Approach: Example

- $A =$ "test is positive (i.e., the test says that the patient has an infection)"
- $B =$ "the patient actually has an infection"

$$\Pr(B|A) = \frac{\Pr(B \cap A)}{\Pr(A)} = \frac{\Pr(A|B)\Pr(B)}{\Pr(A|B)\Pr(B) + \Pr\left(A|\overline{B}\right)\Pr\left(\overline{B}\right)}$$

- What is $\Pr(B)$?

Without any prior knowledge we set $\Pr(B) = \Pr\left(\overline{B}\right) = \frac{1}{2}$:

$$\Pr(B|A) = \frac{\frac{9}{10} \cdot \frac{1}{2}}{\frac{9}{10} \cdot \frac{1}{2} + \frac{1}{10} \cdot \frac{1}{2}} = \frac{9}{10}$$

The estimate is dominated by the reliability of the test.

Bayesian Approach: Example

- $A =$ "test is positive (i.e., the test says that the patient has an infection)"
- $B =$ "the patient actually has an infection"

$$\Pr(B|A) = \frac{\Pr(B \cap A)}{\Pr(A)} = \frac{\Pr(A|B)\Pr(B)}{\Pr(A|B)\Pr(B) + \Pr\left(A|\overline{B}\right)\Pr\left(\overline{B}\right)}$$

- What is $\Pr(B)$?

Assume that we know a priori that the probability of the patient being infected is 80%. We set $\Pr(B) = \frac{4}{5}$:

$$\Pr(B|A) = \frac{\frac{9}{10} \cdot \frac{4}{5}}{\frac{9}{10} \cdot \frac{4}{5} + \frac{1}{10} \cdot \frac{1}{5}} = \frac{36}{37} \approx 0.97$$

The *posterior* probability is sensitive to the choice of *prior* probabilities.

Outline

Recommended Reading

Recommended Reading:

- *Mitchell [1997], Chapter 6.*
- *Zaki and Meira Jr. [2014], Chapter 18.*

Outline

Probabilistic Classification

- A classifier predicts for some object x_q which class c_i it belongs to.
- Often, the prediction can be expressed as probability estimate

$$\Pr(c_i|x_q)$$

- The classifier would then decide for the most likely class:

$$h(x_q) = \arg\max_{c_i \in C} \Pr(c_i|x_q)$$

- Often, this estimate is based on an estimate of how likely the object would be, if it would belong to this or to that class:

$$\Pr(x_q|c_i)$$

  How well can the object be explained if it belongs to a given class?

Proportions of Classes in the Decision Set

- Let $x_1, \ldots, x_k = kNN(x_q)$ be the k nearest neighbors of instance x_q, i.e., the decision set for the instance x_q.
- For a given instance x_q and classes $C = \{c_1, \ldots, c_m\}$:

$$E_j = \text{“}f(x_q) = c_j\text{”}$$

$$\Omega = \bigcup_{j=1}^{m} E_j$$

- The relative frequency of a class c_j in the decision set $kNN(x_q)$ is an empirical estimate of the probability of the event “$f(x_q) = c_j$”:

$$\begin{aligned}\Pr(E_j|x_q) &= \frac{|\{x_i | x_i \in kNN(x_q) \wedge f(x_i) = c_j\}|}{k} \\ &= \frac{\sum_{i=1}^{k} \delta(c_j, f(x_i))}{k} \text{ with } \delta(a,b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ otherwise} \end{cases}\end{aligned}$$

Proportions of Classes in the Decision Set

- We can therefore interpret the composition of the decision set as "class probability vector".
- For classes $C = \{c_1, \ldots, c_m\}$, the decision set for x_q yields a vector

$$\langle p_1, \ldots, p_m \rangle$$

where

$$\begin{aligned} p_j &= \Pr\left(\{f(x_q) = c_j\} | x_q\right) \\ &= \frac{|\{x_i | x_i \in kNN(x_q) \wedge f(x_i) = c_j\}|}{k} \\ &= \frac{\sum_{i=1}^{k} \delta(c_j, f(x_i))}{k} \end{aligned}$$

- How will the quality of the probability estimate depend on k?

Decision Rule — Weight by Distance

As discussed (Slides 237, 241), we can introduce a weight to the components in the decision rule:

$$h(x_q) = \arg\max_{c \in C} \sum_{i=1}^{k} w_i \delta(c, f(x_i))$$

- e.g.: $w_i = \frac{1}{\text{dist}(x_i, x_q)^2}$

$$\Pr_w(\{f(x_q) = c_j\} | x_q) = \frac{\sum_{i=1}^{k} w_i \delta(c_j, f(x_i))}{\sum_{i=1}^{k} w_i}$$

- How do these weights change the dependency on k?

kNN Classifier as Application of Bayes' Rule

- Let $k_i = |\{x|x \in kNN(x_q) \wedge f(x) = c_i\}|$.
- Let $n_i = |\{x|x \in O \wedge f(x) = c_i\}|$, i.e., $\Pr(c_i) = \frac{n_i}{|O|}$
- Let $V_k(x)$ be the volume of the $kNN(x)$.
- $\Pr(x|c_i) = \frac{\frac{k_i}{n_i}}{V_k(x)} = \frac{k_i}{n_i \cdot V_k(x)}$
- $\Pr(x|c_i) \cdot \Pr(c_i) = \frac{k_i}{n_i \cdot V_k(x)} \cdot \frac{n_i}{|O|} = \frac{k_i}{|O| \cdot V_k(x)}$

$$\Pr(c_i|x) = \frac{\Pr(x|c_i) \cdot \Pr(c_i)}{\sum_{j=1}^{m} \Pr(x|c_j) \cdot \Pr(c_j)}$$

$$= \frac{\frac{k_i}{|O| \cdot V_k(x)}}{\sum_{j=1}^{m} \frac{k_j}{|O| \cdot V_k(x)}} = \frac{k_i}{k}$$

$$h(x) = \arg\max_{c_i \in C} (\Pr(c_i|x)) = \arg\max_{c_i \in C} \left(\frac{k_i}{k}\right) = \arg\max_{c_i \in C}(k_i)$$

Decision Rule — Weight by Prior Class Probability

- To account for very imbalanced proportions of class sizes, we can adjust the prior probability.
- Because we *want* each class *a priori* to be equally likely, we set $\Pr(c_i) = \frac{1}{m}$.
- $\Pr(x|c_i) \cdot \Pr(c_i) = \frac{k_i}{n_i \cdot V_k(x)} \cdot \frac{1}{m} = \frac{k_i}{n_i \cdot m \cdot V_k(x)}$

$$\Pr(c_i|x) = \frac{\Pr(x|c_i) \cdot \Pr(c_i)}{\sum_{j=1}^{m} \Pr(x|c_j) \cdot \Pr(c_j)}$$

$$= \frac{\frac{k_i}{n_i \cdot m \cdot V_k(x)}}{\sum_{j=1}^{m} \frac{k_j}{n_j \cdot m \cdot V_k(x)}} = \frac{\frac{k_i}{n_i}}{\sum_{j=1}^{m} \frac{k_j}{n_j}}$$

$$h(x) = \underset{c_i \in C}{\arg\max} \left(\Pr(c_i|x) \right) = \underset{c_i \in C}{\arg\max} \left(\frac{k_i}{n_i} \right)$$

- How does the decision change with the choice of k?

Decision Rule — Weight by Prior Class Probability (Example)

- Given a training set O, $|O| = 100$, classes $C = \{A, B\}$, $n_A = 80$, $n_B = 20$
- We choose to set the prior probability $\Pr(A) = \Pr(B) = \frac{1}{2}$
- $k = 10$, classes of the $kNN(x)$ are: $\{A, A, A, A, A, A, B, B, B, B\}$, i.e., $k_A = 6$, $k_B = 4$

$$\Pr(A|x) = \frac{\Pr(x|A) \cdot \Pr(A)}{\Pr(x|A) \cdot \Pr(A) + \Pr(x|B) \cdot \Pr(B)}$$

$$= \frac{\frac{6}{80}}{\frac{6}{80} + \frac{4}{20}} = \frac{\frac{3}{40}}{\frac{3}{40} + \frac{8}{40}} = \frac{\frac{3}{40}}{\frac{11}{40}} = \frac{3}{11}$$

$$\Pr(B|x) = \frac{\frac{8}{40}}{\frac{11}{40}} = \frac{8}{11}$$

$$h(x) = \arg\max_{c_i \in C} \left(\Pr(c_i|x)\right) = \arg\max_{c_i \in C} \left(\frac{k_i}{n_i}\right)$$

Outline

Probabilities and Learning

- The aim of machine learning (or actually of science as such) could be put as "*find the best hypothesis to explain the observations*".
- If we approach learning probabilistically, "*best*" means "*most probable*", given the data $\mathcal{D}$ plus any initial knowledge about the prior probabilities of the various hypotheses in $\mathcal{H}$.
- The prior probability $\Pr(h)$ denotes the initial probability of hypothesis h before we observe the training data.
- The prior probability could reflect any background knowledge.
- The prior probability $\Pr(\mathcal{D})$ denotes the probability of the data (observations) without any knowledge on which hypothesis holds.

Prior and Posterior Probabilities, and Bayes' Theorem

- The conditional probability $\Pr(\mathcal{D}\,|h)$ denotes the probability of the observations (likelihood of the hypothesis), given some hypothesis h (i.e., assuming, h is correct).
- The probability $\Pr(h|\,\mathcal{D})$ is called the *posterior probability*, because it reflects our confidence that hypothesis h is correct *after* we have seen the training data $\mathcal{D}$.
- Given prior probabilities $\Pr(h)$, $\Pr(\mathcal{D})$, and conditional probability $\Pr(\mathcal{D}\,|h)$, the posterior probability $\Pr(h|\,\mathcal{D})$ can be computed by Bayes' theorem (Theorem 6.8):

$$\Pr(h|\,\mathcal{D}) = \frac{\Pr(\mathcal{D}\,|h) \cdot \Pr(h)}{\Pr(\mathcal{D})}$$

- Intuitive interpretation: $\Pr(h|\,\mathcal{D})$ increases with $\Pr(\mathcal{D}\,|h)$ and with $\Pr(h)$, it decreases as $\Pr(\mathcal{D})$ increases.

Maximum Likelihood and Maximum A Posteriori Hypothesis

- A classifier shall identify the most probable hypothesis $h \in \mathcal{H}$, given the observed data.
- We call such a maximally probable hypothesis a *maximum a posteriori* (MAP) hypothesis:

$$
\begin{aligned}
h_{\text{MAP}} &\equiv \arg\max_{h \in \mathcal{H}} \Pr(h|\mathcal{D}) \\
&= \arg\max_{h \in \mathcal{H}} \frac{\Pr(\mathcal{D}|h) \cdot \Pr(h)}{\Pr(\mathcal{D})} \\
&= \arg\max_{h \in \mathcal{H}} \Pr(\mathcal{D}|h) \cdot \Pr(h)
\end{aligned}
$$

- If we assume equal prior probabilities for all hypotheses (i.e., $\Pr(h_i) = \Pr(h_j) \forall h_i, h_j \in \mathcal{H}$), MAP is given by the maximum likelihood hypothesis:

$$h_{\text{ML}} \equiv \arg\max_{h \in \mathcal{H}} \Pr(\mathcal{D}|h)$$

Most Probable Hypothesis vs. Most Probable Classification

- Consider some hypothesis space $\mathcal{H} = \{h_1, h_2, h_3\}$ with $\Pr(h_1|\mathcal{D}) = 0.4$, $\Pr(h_2|\mathcal{D}) = 0.3$, $\Pr(h_3|\mathcal{D}) = 0.3$.
- Obviously, h_1 is the MAP hypothesis.
- Suppose a new instance x is encountered, where

$$h_1(x) = A$$
$$h_2(x) = B$$
$$h_3(x) = B$$

- Taking all hypotheses into account, we have:

$$\Pr(A|x) = 0.4$$
$$\Pr(B|x) = 0.6$$

- The most probable classification is different from the classification generated by the MAP hypothesis.

Bayes Optimal Classification

- We obtain the most probable classification by combining the predictions of all hypotheses weighted by the posterior probabilities.
- For the set of classes C, for any $c_j \in C$, we have

$$\Pr(c_j | \mathcal{D}) = \sum_{h_i \in \mathcal{H}} \Pr(c_j | h_i) \Pr(h_i | \mathcal{D})$$

- The optimal classification is therefore:

$$\arg\max_{c_j \in C} \sum_{h_i \in \mathcal{H}} \Pr(c_j | h_i) \Pr(h_i | \mathcal{D})$$

- Any system classifying new instances according to this rule is called a “Bayes optimal classifier”.

Example

With $C = \{A, B\}$ and the above example, the possible classifications of the new instance x are:

$$\begin{array}{lll} \Pr(h_1|\mathcal{D}) = 0.4 & \Pr(A|h_1) = 1 & \Pr(B|h_1) = 0 \\ \Pr(h_2|\mathcal{D}) = 0.3 & \Pr(A|h_2) = 0 & \Pr(B|h_2) = 1 \\ \Pr(h_3|\mathcal{D}) = 0.3 & \Pr(A|h_3) = 0 & \Pr(B|h_3) = 1 \end{array}$$

Therefore:

$$\sum_{h_i \in \mathcal{H}} \Pr(A|h_i) \Pr(h_i|\mathcal{D}) = 0.4$$

$$\sum_{h_i \in \mathcal{H}} \Pr(B|h_i) \Pr(h_i|\mathcal{D}) = 0.6$$

and

$$\arg\max_{c_j \in \{A,B\}} \sum_{h_i \in \mathcal{H}} \Pr(c_j|h_i) \Pr(h_i|\mathcal{D}) = B$$

Properties of the Bayes Optimal Classifier

- The predictions of the Bayes optimal classifier can correspond to the predictions of a hypothesis that is not contained in the original hypothesis space $\mathcal{H}$!
- The Bayes optimal classifier considers effectively a different hypothesis space $\mathcal{H}'$, including hypotheses that perform comparisons between linear combinations of predictions from multiple hypotheses in $\mathcal{H}$.

Note that:

The Bayes optimal learner maximizes the probability that the new instance is classified correctly, given the available data, hypothesis space, and prior probabilities over the hypotheses. Thus no other classification method using the same hypothesis space and same prior knowledge can outperform this method on average.

Outline

Estimates of Prior Probabilities

Consider a learning task to distinguish apples, oranges, and other fruits, where the objects are described by color and shape:

- 20% of the objects are apples
- 30% of the objects are oranges

} prior class probability

- 50% of the objects are round
- 40% of the objects have an orange color

} prior probability of some attribute value

Estimates of Posterior Probabilities

Posterior (conditional) probabilities model relations between attribute values and classes:

- 100% of the oranges are round: $\Pr(\text{shape=round}|\text{ORANGE})$
- 100% of the apples are round: $\Pr(\text{shape=round}|\text{APPLE})$
- 90% of the oranges have the color orange: $\Pr(\text{color=orange}|\text{ORANGE})$

Bayes Classification

For a given attribute value a_q, we can compute the posterior class probability according to Bayes' rule:

$$\Pr(c_j|a_q) = \frac{\Pr(a_q|c_j)\Pr(c_j)}{\Pr(a_q)} = \frac{\Pr(a_q|c_j)\Pr(c_j)}{\sum_{c_i \in C}\Pr(c_i)\Pr(a_q|c_i)}$$

We estimate probabilities from the training data.
For example, we have an object with color orange:

$$\begin{aligned}
&\Pr(\text{ORANGE}|\text{color=orange}) \\
&\quad = \frac{\Pr(\text{color=orange}|\text{ORANGE})\Pr(\text{ORANGE})}{\Pr(\text{color=orange})} \\
&\quad = \frac{0.9 \cdot 0.3}{0.4} \\
&\quad = 0.675
\end{aligned}$$

Maximum Likelihood Classification

Given all posterior class probabilities, we predict the most likely class:

$$
\begin{aligned}
h_{\text{MAP}} &= \arg\max_{c_i \in C} \Pr(c_i | a_q) \\
&= \arg\max_{c_i \in C} \frac{\Pr(a_q | c_i) \Pr(c_i)}{\Pr(a_q)} \\
&= \arg\max_{c_i \in C} \Pr(a_q | c_i) \Pr(c_i)
\end{aligned}
$$

Discrete Attributes

we can count relative frequencies to estimate probabilities:

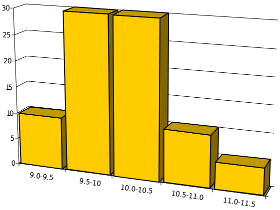
ID	shape	color	class
1	round	orange	A
2	round	green	A
3	round	yellow	A
4	square	green	A
5	oval	white	B

$$\Pr(\text{shape=round}|A) = \frac{3}{4}$$

$$\Pr(\text{color=green}|A) = \frac{2}{4}$$

$$\Pr(\text{shape=oval}|A) = \frac{0}{4}$$

Continuous Metric Attributes



$$\Pr(9.0 < \text{diameter} \leq 9.5|A) = 10\%$$
$$\Pr(9.5 < \text{diameter} \leq 10.0|A) = 30\%$$
$$\Pr(10.0 < \text{diameter} \leq 10.5|A) = 30\%$$
$$\Pr(10.5 < \text{diameter} \leq 11.0|A) = 10\%$$
$$\Pr(11.0 < \text{diameter} \leq 11.5|A) = 5\%$$

Zero Probabilities?

Note that:

- *Problem:* $\Pr($*shape=oval*$|A) = 0$ *would rule out any slight possibility of predicting an instance of class* A*.*
- *Heuristic solution: smoothing (use some artificial small minimum probability):*

$$\Pr(\text{shape} = \text{oval}|A) := \max\left\{\frac{0}{4}, \varepsilon\right\} \textit{ with } 0 < \varepsilon \ll 1$$

Multi-dimensional Data

- So far, we considered only one attribute.
- In multi-dimensional data, we need to estimate the combined probabilities of specific attribute values:

$$
\begin{aligned}
h_{\text{MAP}} &= \arg\max_{c_i \in C} \Pr(c_i | a_1 \cap a_2 \cap a_3 \cap \ldots \cap a_n) \\
&= \arg\max_{c_i \in C} \frac{\Pr(a_1 \cap a_2 \cap a_3 \cap \ldots \cap a_n | c_i) \Pr(c_i)}{\Pr(a_1 \cap a_2 \cap a_3 \cap \ldots \cap a_n)} \\
&= \arg\max_{c_i \in C} \Pr(a_1 \cap a_2 \cap a_3 \cap \ldots \cap a_n | c_i) \Pr(c_i)
\end{aligned}
$$

Example:

ID	*shape*	*color*	*class*
1	*round*	*orange*	*A*
2	*round*	*green*	*A*
3	*round*	*yellow*	*A*
4	*square*	*green*	*A*
5	*oval*	*white*	*B*

$$\Pr(\textit{shape=round} \cap \textit{color=orange} | A) = \frac{1}{4}$$

$$\Pr(\textit{shape=round} \cap \textit{color=green} | A) = \frac{1}{4}$$

Problems for the Bayes Classifier in Multi-dimensional Data

Problems:

- If we have n attributes, and each can take on r different values, we have r^n different attribute combinations.
- Typically, there are not enough training instances available to reliably estimate probabilities.

Example

ID	shape	color	class
1	round	orange	A
2	round	green	A
3	round	yellow	A
4	square	green	A
5	oval	white	B

$$\Pr(\text{shape=round} \cap \text{color=orange}|A) = \frac{1}{4}$$

$$\Pr(\text{shape=round} \cap \text{color=green}|A) = \frac{1}{4}$$

$$\Pr(\text{shape=round} \cap \text{color=yellow}|A) = \frac{1}{4}$$

$$\Pr(\text{shape=round} \cap \text{color=white}|A) = \frac{0}{4}$$

$$\Pr(\text{shape=oval} \cap \text{color=orange}|A) = \frac{0}{4}$$

$$\Pr(\text{shape=oval} \cap \text{color=green}|A) = \frac{0}{4}$$

$$\Pr(\text{shape=oval} \cap \text{color=yellow}|A) = \frac{0}{4}$$

$$\Pr(\text{shape=oval} \cap \text{color=white}|A) = \frac{0}{4}$$

$$\Pr(\text{shape=square} \cap \text{color=orange}|A) = \frac{0}{4}$$

$$\Pr(\text{shape=square} \cap \text{color=green}|A) = \frac{1}{4}$$

$$\Pr(\text{shape=square} \cap \text{color=yellow}|A) = \frac{0}{4}$$

$$\Pr(\text{shape=square} \cap \text{color=white}|A) = \frac{0}{4}$$

$$\Pr(\text{shape=round} \cap \text{color=orange}|B) = \frac{0}{1}$$

$$\vdots$$

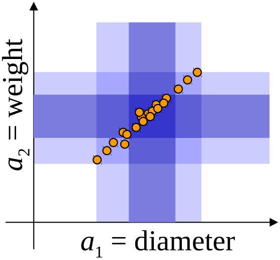
The probability estimates are unreliable, because the sample size is too small for each instance.

The Naïve Assumption: Independence

If we assume independence among the attributes, we can estimate the combined probabilities based on Definition 6.5 $(\Pr(E \cap F) = \Pr(E) \cdot \Pr(F))$:

$$h_{\text{MAP}} = \arg\max_{c_i \in C} \Pr(a_1 \cap a_2 \cap a_3 \cap \ldots \cap a_n | c_i) \Pr(c_i)$$

$$= \arg\max_{c_i \in C} \prod_{j=1}^{n} \Pr(a_j | c_i) \Pr(c_i) \qquad \text{(Ass. of Indep.)}$$



- The assumption might be wrong.
- Then we don't get the correct probabilities.
- But we *might* still get the correct maximum.
- In practice, the assumption often works despite *some* dependency among the attributes. advanced reading: Domingos

Naïve Bayes Classifier

- The "Naïve Bayes classifier" relies on the assumption of independence of attributes.
- The various $\Pr(c_i)$ and $\Pr(a_j|c_i)$ terms are estimated based on the relative frequencies of corresponding examples in the training data.
- The set of these estimates constitutes the learned hypothesis.
- The class prediction is based on these estimates according to:

$$h_{\text{naïve Bayes}} = \arg\max_{c_i \in C} \prod_{j=1}^{n} \Pr(a_j|c_i)\Pr(c_i)$$

- Because of the multiplication, the replacement of zero probabilities by some heuristic minimum probability ε (cf. slide 309) is particularly important.

Example: Should We Play Tennis Today?

ID	forecast	temperature	humidity	wind	play tennis?
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rainy	mild	high	strong	no

Example Prediction

Classify a new instance: $\langle \text{sunny}, \text{cool}, \text{high}, \text{strong} \rangle$

$$
\begin{aligned}
h_{\text{naïve Bayes}} &= \arg\max_{c_i \in \{\text{yes},\text{no}\}} \prod_{j=1}^{n} \Pr(a_j | c_i) \Pr(c_i) \\
&= \arg\max_{c_i \in \{\text{yes},\text{no}\}} \Pr(\text{sunny}|c_i) \Pr(\text{cool}|c_i) \Pr(\text{high}|c_i) \\
&\qquad \Pr(\text{strong}|c_i) \Pr(c_i)
\end{aligned}
$$

$$\Pr(\text{yes}) = \frac{9}{14} = 0.64 \qquad \Pr(\text{wind=strong}|\text{yes}) = \frac{3}{9} = 0.33$$

$$\Pr(\text{no}) = \frac{5}{14} = 0.36 \qquad \Pr(\text{wind=strong}|\text{no}) = \frac{3}{5} = 0.60$$

$$\vdots$$

Decision and Probability

The Naïve Bayes classifier decides by finding the class maximizing the product of probabilities:

$$\begin{aligned}
\Pr(\text{sunny}|\text{yes})\Pr(\text{cool}|\text{yes})\Pr(\text{high}|\text{yes})\Pr(\text{strong}|\text{yes})\Pr(\text{yes}) &= 0.0053\\
\Pr(\text{sunny}|\text{no})\Pr(\text{cool}|\text{no})\Pr(\text{high}|\text{no})\Pr(\text{strong}|\text{no})\Pr(\text{no}) &= 0.0206
\end{aligned}$$

$$h_{\text{naïve Bayes}}\left(\langle\text{sunny},\text{cool},\text{high},\text{strong}\rangle\right) = \text{no}$$

If we are interested in the conditional probability for "no", we could normalize these quantities to sum up to one:

$$\frac{0.0206}{0.0206+0.0053} = 0.795$$

Assumption of Independence is a Bias

- The assumption of independence can be seen as the bias inherent to the Naïve Bayes classifier.
- An unbiased probabilistic classifier is not practical due to a notorious lack of training examples.
  - In other words: in any practical scenario, it would hopelessly overfit.
- For data consisting of two classes and only 30 binary attributes, we would need more than 2 billion examples just to see each combination *once* (which is not good enough to derive reliable probability estimates).
- Relying on the bias, the classifier may have a tendency to be wrong (if the assumption does not hold).
- The bias is necessary to make generalization feasible.

Outline

Summary

You learned in this section:

- *Axioms of probability:*
  - *sample space*
  - *event*
  - *probability function*
  - *probability space*
- *independence and conditional probability*
- *probabilistic interpretation of quality measures for association rules*
- *Bayes' rule*

Summary

You learned in this section:

- *Bayesian Learning:*
  - *k nearest neighbor classifier as an application of Bayes' rule for learning*
  - *The principle of Bayesian learning:*
    - *prior and posterior probabilities*
    - *data as evidence to adapt probability estimates and to select hypotheses*
    - *MAP hypothesis*
    - *Bayesian reasoning*
  - *Bayes optimal classifier*
  - *Naïve Bayes classifier*

Outline

Outline

Recommended Reading

Recommended Reading:

- *Mitzenmacher and Upfal [2017], Chapter 2.*

Event vs. Values Associated with the Event

- When throwing two dice, we have the 36 basic events of equal probability given by the ordered pair of numbers $\Omega = \{(1,1),(1,2),\ldots,(6,5),(6,6)\}$.
- As opposed to being interested in some basic event as such, we are often interested in some more complex event, e.g., the sum of the two dice.
- We have in this case actually 11 different events (the possible outcomes of the sum).
- These 11 events have unequal probability.
- We can map any of the basic events to some of the 11 complex events (sum is 2, 3, . . . , 12).
- Any such function from the sample space to the real numbers is called a *random variable*.

Random Variable

Definition 7.1 (Random Variable)

A *random variable* X on a sample space Ω is a real-valued function from Ω to the real numbers:

$$X : \Omega \to \mathbb{R}$$

Definition 7.2 (Discrete Random Variable)

A *discrete random variable* is a random variable that takes on only a finite or countably infinite number of values.

For example, for some $I \subseteq \mathbb{R}$:

$$X : \Omega \to I$$

is a *discrete random variable*, if I is finite or countably infinite (e.g., $I = \mathbb{N}$, $I = \{0, 1\}$).

Random Variables and Basic Events

- For a discrete random variable X and $a \in \mathbb{R}$, the event "$X = a$" includes all basic events of the sample space in which the random variable X assumes the value a.
- "$X = a$" represents the set $\{s \in \Omega | X(s) = a\}$.
- The probability of that event is given by:

$$\Pr(X = a) = \sum_{s \in \Omega : X(s) = a} \Pr(s)$$

Example

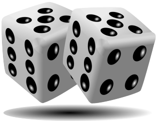
- Let X be the random variable representing the sum of two dice.
- $\Omega = \{(1,1),(1,2),\ldots,(1,6),(2,1),(2,2),\ldots,(6,6)\}$
- The event $X = 4$ corresponds to the set of basic events: $\{(1,3),(2,2),(3,1)\}$:

$$X((1,3)) = 4; \quad X((2,2)) = 4; \quad X((3,1)) = 4;$$

$$\forall (x,y) \in \Omega \setminus \{(1,3),(2,2),(3,1)\} : X((x,y)) \neq 4$$

- Therefore:

$$\Pr(X = 4) = \frac{3}{36} = \frac{1}{12}$$



Probability Distribution

Definition 7.3 (Probability Distribution of a Discrete Random Variable)

The *probability distribution* (or: *probability (mass) function (pmf)*) of a discrete random variable X is defined for every value x in the range of X by:

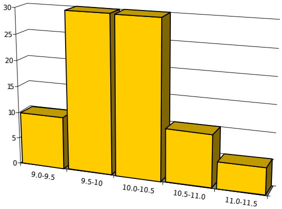
$$p_X(x) = \Pr(X = x) = \sum_{s \in \Omega, X(s) = x} \Pr(s),$$

where $p_X(x) \geq 0$ and $\sum_{-\infty < x < \infty} \Pr(x) = 1$.

The probability distribution of X says how the total probability of 1 is distributed among all possible values of X.

Probability Distribution: Example

(cf. slide 308:)



$$\Pr(9.0 < \text{diameter} \leq 9.5|A) = 10\%$$
$$\Pr(9.5 < \text{diameter} \leq 10.0|A) = 30\%$$
$$\Pr(10.0 < \text{diameter} \leq 10.5|A) = 30\%$$
$$\Pr(10.5 < \text{diameter} \leq 11.0|A) = 10\%$$
$$\Pr(11.0 < \text{diameter} \leq 11.5|A) = 5\%$$

Cumulative Distribution Function

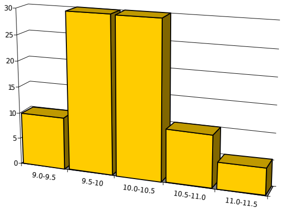
Definition 7.4 (Cumulative Distribution Function of a Discrete Random Variable)

The *cumulative distribution function (cmf)* of a discrete random variable X with pmf $p(x)$ is defined for every value x by:

$$F(x) = \Pr(X \leq x) = \sum_{y:y \leq x} p(y).$$

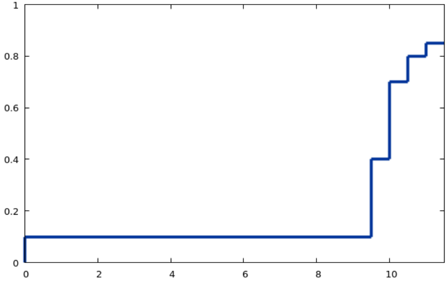
For any number x, the value of the cmf $F(x)$ is the probability that the observed value of X will be at most x.

Cumulative Distribution Function: Example



$$\Pr(\text{diameter} \leq 9.5|A) = 10\%$$
$$\Pr(\text{diameter} \leq 10.0|A) = 40\%$$
$$\Pr(\text{diameter} \leq 10.5|A) = 70\%$$
$$\Pr(\text{diameter} \leq 11.0|A) = 80\%$$
$$\Pr(\text{diameter} \leq 11.5|A) = 85\%$$

(cf. slide 308)



Independent Events

The definition of independence of events extends to random variables:

Definition 7.5 (Independence of Random Variables)

Two random variables X and Y are independent if and only if

$$\Pr((X = x) \cap (Y = y)) = \Pr(X = x) \cdot \Pr(Y = y)$$

for all values x and y.

Similarly, random variables $X_1, X_2, \ldots, X_k$ are mutually independent if and only if, for any subset $I \subseteq [1, k]$ and any values $x_i, i \in I$:

$$\Pr\left(\bigcap_{i \in I} X_i = x_i\right) = \prod_{i \in I} \Pr(X_i = x_i)$$

Expectation (or: Mean)

- The *expectation* (or: *mean*) is a basic characteristic of a random variable.
- It is the weighted average of the values assumed by the random variable.
- The weight of a value is the probability that the variable assumes that value.

Definition 7.6 (Expectation)

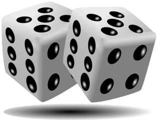
The *expectation* of a discrete random variable X is given by:

$$E[X] = \mu_X = \sum_i i \Pr(X = i),$$

where the summation is over all values i in the range of X. The expectation is finite if $\sum_i |i| \Pr(X = i)$ converges; otherwise, the expectation is unbounded.

Example

The expectation of the random variable X representing the sum of two dice is:



$$
\begin{aligned}
E[X] = \quad & 2 \cdot \frac{|\{(1,1)\}|}{|\Omega|} \\
& + 3 \cdot \frac{|\{(1,2),(2,1)\}|}{|\Omega|} \\
& + 4 \cdot \frac{|\{(1,3),(2,2),(3,1)\}|}{|\Omega|} \\
& + \ldots \\
& + 12 \cdot \frac{|\{(6,6)\}|}{|\Omega|} \\
= \quad & 2 \cdot \frac{1}{36} + 3 \cdot \frac{2}{36} + 4 \cdot \frac{3}{36} + \ldots + 12 \cdot \frac{1}{36} \\
= \quad & 7
\end{aligned}
$$

Example

The expectation of the random variable X representing the value of a single die is:

$$\begin{aligned}
E[X] &= 1 \cdot \frac{|\{1\}|}{|\Omega|} + 2 \cdot \frac{|\{2\}|}{|\Omega|} + 3 \cdot \frac{|\{3\}|}{|\Omega|} + \ldots + 6 \cdot \frac{|\{6\}|}{|\Omega|} \\
&= 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + \ldots + 6 \cdot \frac{1}{6} \\
&= \frac{21}{6} \\
&= 3.5
\end{aligned}$$

Note that:

The expected value of X, $E(X)$, is not necessarily a possible value of X.

Outline

Recommended Reading

Recommended Reading:

- *Mitzenmacher and Upfal [2017], Chapter 3.*
- *Not strictly technical reading but nevertheless quite entertaining are the books by Taleb [2001, 2007].*

Variance and Standard Deviation

Definition 7.7 (Variance)

The *variance* of a random variable X is given by:

$$\mathrm{Var}[X] = E\left[(X - E[X])^2\right] = E\left[X^2\right] - (E[X])^2$$

In retrospect, understand the name of Algorithm 4.1.

Definition 7.8 (Standard Deviation)

The *standard deviation* of a random variable X is

$$\sigma[X] = \sqrt{\mathrm{Var}[X]}$$

Note that:

If a random variable is constant, then its variance and standard deviation are zero.

Covariance

Definition 7.9 (Covariance)

The *covariance* of two random variables X and Y is

$$\mathrm{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

Theorem 7.10

For any two random variables X and Y,

$$\mathrm{Var}[X + Y] = \mathrm{Var}[X] + \mathrm{Var}[Y] + 2\,\mathrm{Cov}(X, Y)$$

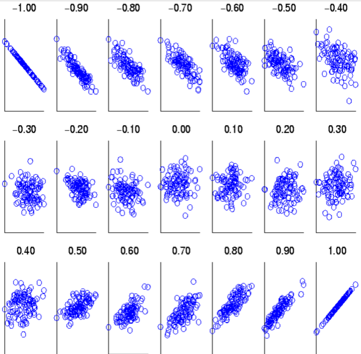
Proof: see Mitzenmacher and Upfal [2017], p. 49

Correlation

Definition 7.11 (Correlation)

The *correlation coefficient* of two random variables X and Y is given by

$$\mathrm{Corr}(X,Y) = \rho_{X,Y} = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}[X]}\sqrt{\mathrm{Var}[Y]}}$$



Independence and Expectation

Theorem 7.12

If X and Y are two independent random variables, then

$$E[X \cdot Y] = E[X] \cdot E[Y]$$

Corollary 7.13

If X and Y are independent random variables, then

$$\mathrm{Cov}(X, Y) = 0$$

and

$$\mathrm{Var}[X + Y] = \mathrm{Var}[X] + \mathrm{Var}[Y]$$

Proofs: see Mitzenmacher and Upfal [2017] pp. 49f.

Variance of Sum and Sum of Variances

Theorem 7.14

Let $X_1, X_2, \ldots, X_n$ be mutually independent random variables. Then

$$\mathrm{Var}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathrm{Var}[X_i]$$

Proof: by induction from Corrolary 7.13.

Example

We flip a coin and get a point for head, zero points for tail.

- $\Pr(X=0) = \Pr(X=1) = \frac{1}{2}$
- $E[X] = \frac{1}{2}$

$$\begin{aligned}
\mathrm{Var}[X] &= E\left[(X - E[X])^2\right] \\
&= \frac{1}{2}\left(1 - \frac{1}{2}\right)^2 + \frac{1}{2}\left(0 - \frac{1}{2}\right)^2 \\
&= \frac{1}{4} \\
\sigma[X] &= \frac{1}{2}
\end{aligned}$$

Mean and Median

For any random variable X with finite expectation $E[X]$ and finite median m,

1. the expectation $E[X]$ (= mean) is the value of c that minimizes $E[(X-c)^2]$
2. the median m is a value of c that minimizes $E[|X-c|]$.

Note that:

This is why we use different objective functions, TD^2 and TD, for k-means and k-medoids, respectively (see Slides 159 and 172).

Outline

Recommended Reading

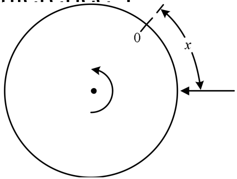
Recommended Reading:

- *The material of this section is covered by Mitzenmacher and Upfal [2017], Chapters 8&9.*
- *Broad coverage (although following a different systematic) is given by Devore and Berk [2012], scattered over Chapters 4&5.*

Outline

Probability Distributions in $\mathbb{R}$

- A continuous roulette wheel has circumference 1
- The outcome when the wheel stops after spinning is the clockwise distance x from the "0" to the arrow.



- $\Omega = [0, 1)$
- Any point on the circumference of the disk is equally likely to face the arrow when the disk stops.
- What is the probability p of a given outcome x?

Note that:

Notation for subsets of $\mathbb{R}$:

$$[a, b] = \{x | x \in \mathbb{R}, a \leq x \leq b\} \quad [a, b) = \{x | x \in \mathbb{R}, a \leq x < b\}$$
$$(a, b] = \{x | x \in \mathbb{R}, a < x \leq b\} \quad (a, b) = \{x | x \in \mathbb{R}, a < x < b\}$$

Probabilities in an Infinite Sample Space

Recall Definition 6.1:

1. $\forall E : 0 \leq \Pr(E) \leq 1$;
2. $\Pr(\Omega) = 1$; and
3. for any finite or countably infinite sequence of pairwise mutually disjoint events E_i: $\Pr\left(\bigcup_{i\geq 1} E_i\right) = \sum_{i\geq 1} \Pr(E_i)$.

- Let $S(k)$ be a set of k distinct points in the range $[0, 1)$, let p be the probability that any given point in $[0, 1)$ is the outcome of the experiment.
- Since the probability of any event is bounded by 1, we have: $\Pr(x \in S(k)) = kp \leq 1)$
- Since we can choose any integer k, to ensure $kp \leq 1$, we must have $p = 0$.

Probabilities in an Infinite Sample Space

Note that:

In an infinite sample space

- *there may be possible events that have probability 0;*
- *there may be events with probability 1*
  (e.g., the complement of an event with probability 0)
  that do not correspond to all possible outcomes
  (and therefore are not certain).

Cumulative Distribution Function

Definition 7.15 (Cumulative Distribution Function)

The probability distribution of a random variable X is given by its *(cumulative) distribution function* $F(x)$, where for any $x \in \mathbb{R}$:

$$F(x) = \Pr(X \leq x).$$

We say that a random variable X is *continuous*, if its distribution function $F(x)$ is a continuous function of x.

Note that:

- *For a continuous random variable X, we must have $\Pr(X = x) = 0$ for any specific value of x.*
- *As a consequence, we have $\Pr(X \leq x) = \Pr(X < x)$*

Probability Density Function

Definition 7.16 (Probability Density Function)

If there is a function $f(x)$ such that, for all $-\infty < a < \infty$,

$$F(a) = \int_{-\infty}^{a} f(t)dt,$$

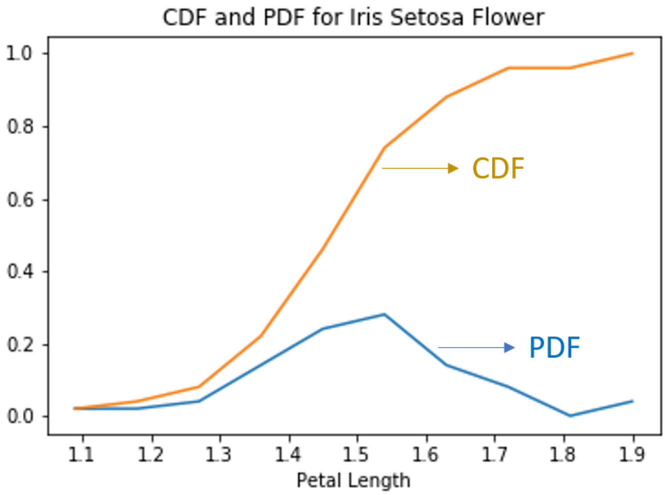
then $f(x)$ is called the *(probability) density function* of $F(x)$. Where the derivative of F is well-defined, we have

$$f(x) = F'(x).$$

For events in $\mathbb{R}$, the integral is the analogue to the sum of probabilities of simple events (cf. Def. 6.2):

$$\Pr(a \leq X < b) = \Pr(X \in [a, b)) = \int_{a}^{b} f(x)dx$$

CDF and PDF for Iris setosa



Joint Distribution

Definition 7.17 (Joint Distribution)

The *joint distribution function* of X and Y is

$$F(x,y) = \Pr(X \leq x, Y \leq y).$$

The variables X and Y have a joint density function f if, for all x, y:

$$F(x,y) = \int_{-\infty}^{y} \int_{-\infty}^{x} f(u,v) du dv$$

When the derivative exists, we denote

$$f(x,y) = \frac{\partial^2}{\partial x \partial y} F(x,y).$$

Outline

The Uniform Distribution

Definition 7.18 (Uniform Distribution)

When a random variable X assumes values in the interval $[a, b]$ such that all subintervals of equal length have equal probability, we say that

- X has a *uniform distribution* over the interval $[a, b]$
- X is uniform over the interval $[a, b]$

The uniform distribution is denoted by $\mathcal{U}[a, b]$.
The random variable X following a uniform distribution is denoted by $X \sim \mathcal{U}[a, b]$.

Note that:

Since the probability of taking on any specific value when $b > a$ is 0, the distributions $\mathcal{U}[a, b]$, $\mathcal{U}[a, b)$, $\mathcal{U}(a, b]$, and $\mathcal{U}(a, b)$ are essentially the same.

Probability Distribution and Density Function

The probability distribution function of a random variable X that is uniform over $[a, b]$ is:

$$F(x) = \begin{cases} 0 & \text{if } x \leq a, \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b, \\ 1 & \text{if } x \geq b \end{cases}$$

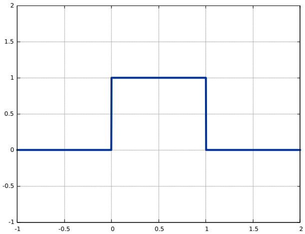
and its density function is

$$f(x) = \begin{cases} 0 & \text{if } x < a, \\ \frac{1}{b-a} & \text{if } a \leq x \leq b, \\ 0 & \text{if } x > b \end{cases}$$

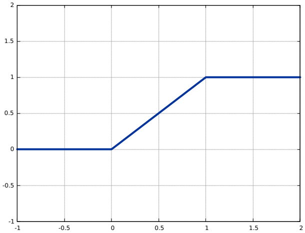
$$E[X] = \frac{b+a}{2}, \qquad \mathrm{Var}[X] = \frac{(b-a)^2}{12}$$

Example: $\mathcal{U}[0, 1]$

$f(x)$:



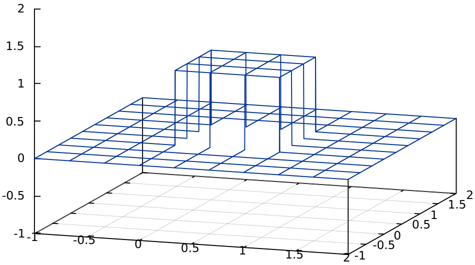
$F(x)$:



Two-dimensional Uniform Distribution

As an example of a joint distribution of two in the interval $[0,1]$ independently uniformly distributed random variables X and Y, we have a two-dimensional uniform distribution with density function

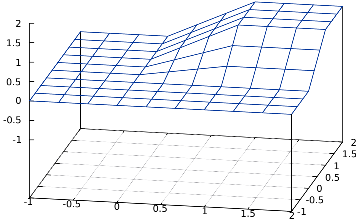
$$f(x,y) = \begin{cases} 1 & \text{if } (x,y) \in [0,1]^2 \\ 0 & \text{otherwise} \end{cases}$$



Two-dimensional Uniform Distribution

The joint probability distribution of $X \sim \mathcal{U}[0,1]$ and $Y \sim \mathcal{U}[0,1]$ is

$$F(x,y) = \begin{cases} 0 & \text{if } x < 0 \vee y < 0 \\ xy & \text{if } (x,y) \in [0,1]^2 \\ x & \text{if } 0 \leq x \leq 1 \wedge y > 1 \\ y & \text{if } 0 \leq y \leq 1 \wedge x > 1 \\ 1 & \text{if } x > 1 \wedge y > 1 \end{cases}$$



or equivalently

$$F(x,y) = \Pr(X \leq x, Y \leq y) = \int_0^{\min\{1,x\}} \int_0^{\min\{1,y\}} 1 \, dy \, dx$$

Two-dimensional Uniform Distribution

More general: let $S \subset \mathbb{R}^2$ such that the area of S is finite.

$$f(x,y) = \begin{cases} \frac{1}{\text{area}(S)} & \text{if } (x,y) \in S \\ 0 & \text{otherwise} \end{cases}$$

$$\Pr((X,Y) \in B) = \frac{\text{area}(B \cap S)}{\text{area}(S)}$$

Outline

The Normal Distribution

Definition 7.19 (Normal Distribution)

A random variable X follows a *normal distribution* (also: *Gaussian*) with parameters $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$, if the pdf of X for $-\infty < x < \infty$ is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

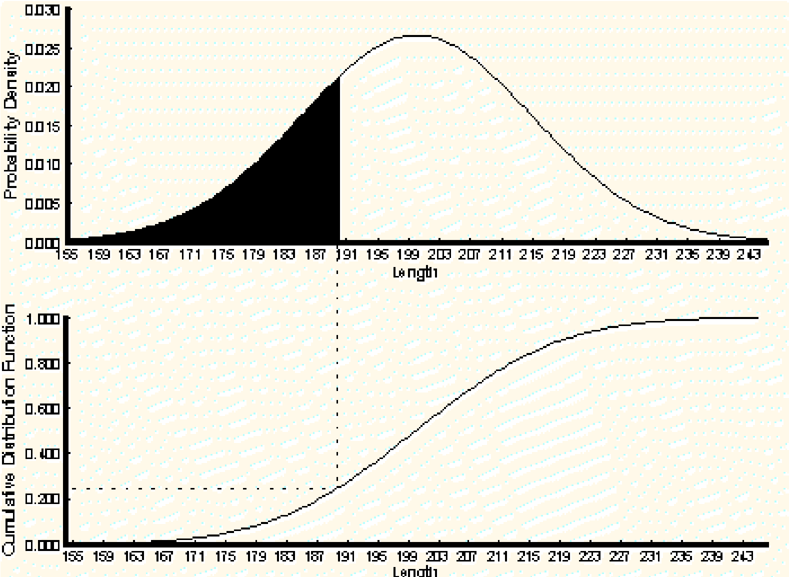
The normal distribution is denoted as $\mathcal{N}(\mu, \sigma^2)$.

Note that:

For the parameters μ and σ^2 we have:

$$\mu = E[X] \qquad \sigma^2 = \mathrm{Var}[X]$$

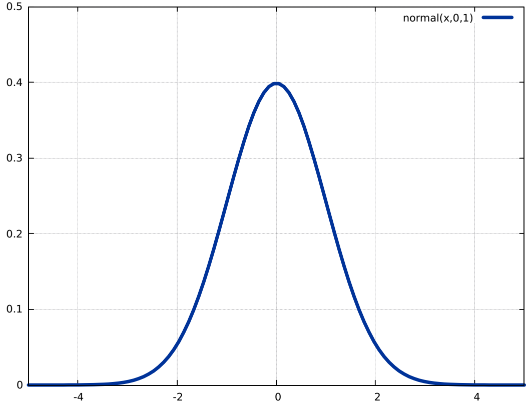
CDF and PDF for some Normal Distribution



Standard Normal Distribution

Definition 7.20 (Standard Normal Distribution)

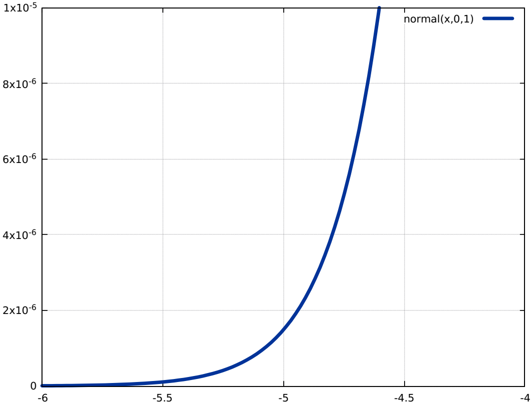
The *standard normal distribution* is given by $\mu = 0$ and $\sigma^2 = 1$, i.e., $\mathcal{N}(0, 1)$.



Standard Normal Distribution

Definition 7.20 (Standard Normal Distribution)

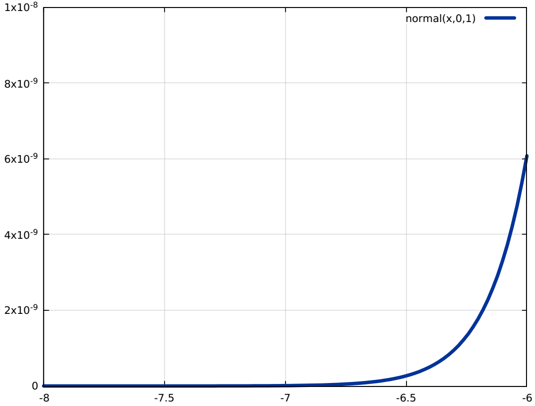
The *standard normal distribution* is given by $\mu = 0$ and $\sigma^2 = 1$, i.e., $\mathcal{N}(0, 1)$.



Standard Normal Distribution

Definition 7.20 (Standard Normal Distribution)

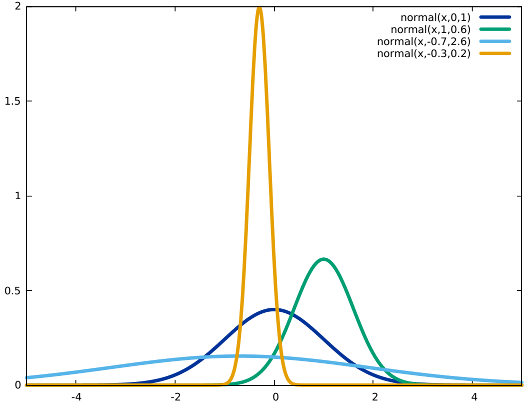
The *standard normal distribution* is given by $\mu = 0$ and $\sigma^2 = 1$, i.e., $\mathcal{N}(0, 1)$.



Geometrical Meaning of μ and σ

In general:

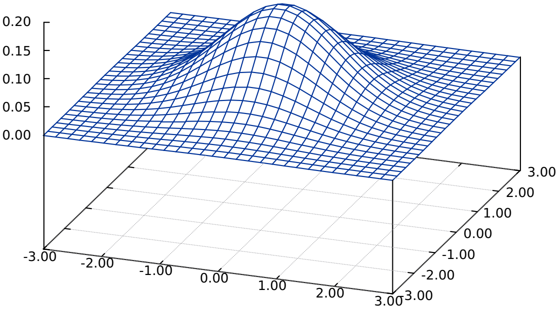
- μ is the center of the symmetric, bell-shaped curve $f(x)$.
- $\mu + \sigma$ and $\mu - \sigma$ are the inflection points of the curve.



Bivariate Normal Distribution

The joint probability density function of two random variables $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$ is

$$f(x,y) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-\frac{\left(\frac{x-\mu_1}{\sigma_1}\right)^2 - \frac{2\rho(x-\mu_1)(y-\mu_2)}{\sigma_1\sigma_2} + \left(\frac{y-\mu_2}{\sigma_2}\right)^2}{2(1-\rho^2)}}$$



Multivariate Normal Distribution

In the general, d-dimensional case for a vector of normal distributed random variables $\vec{X} = (X_1, \ldots, X_d)$, we have the multi-dimensional probability density function for $\vec{x} = (x_1, \ldots, x_d)$

$$f(\vec{x}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \cdot e^{-\frac{(\vec{x}-\vec{\mu})\Sigma^{-1}(\vec{x}-\vec{\mu})^{\mathsf{T}}}{2}}$$

where $|\Sigma|$ is the determinant of the covariance matrix:

$$\Sigma = \begin{pmatrix} \mathrm{Var}[X_1] & \mathrm{Cov}(X_1, X_2) & \mathrm{Cov}(X_1, X_3) & \cdots & \mathrm{Cov}(X_1, X_d) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}[X_2] & \mathrm{Cov}(X_2, X_3) & \cdots & \mathrm{Cov}(X_2, X_d) \\ \mathrm{Cov}(X_3, X_1) & \mathrm{Cov}(X_3, X_2) & \mathrm{Var}[X_3] & \cdots & \mathrm{Cov}(X_3, X_d) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_d, X_1) & \mathrm{Cov}(X_d, X_2) & \mathrm{Cov}(X_d, X_3) & \cdots & \mathrm{Var}[X_d] \end{pmatrix}$$

Standardizing

- If we have a random variable $X \sim \mathcal{N}(\mu, \sigma^2)$, then the random variable
$$Z = \frac{X - \mu}{\sigma}$$
is standard normal distributed.
- Standardizing a normal distribution is equivalent to expressing the distance from the mean value as some number of standard deviations (sometimes called the z-score).

Outline

SDU
SYDDANSK UNIVERSITET

Outline

Algebraic and Geometric View of Data

- If we have a dataset consisting of d numeric attributes and n data objects, we can describe the data as an $n \times d$ data matrix, where each row is a vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \in \mathbb{R}^d$.
- We can therefore work with the dataset by algebraic or geometric operations such as
  - compute distances and angles between vectors,
  - compute the length of a vector,
  - compute the average of a set of vectors, or
  - transform (multiply, rotate, project) the vectors.
- We follow an algebraic/geometric intuition when working with data in this manner.

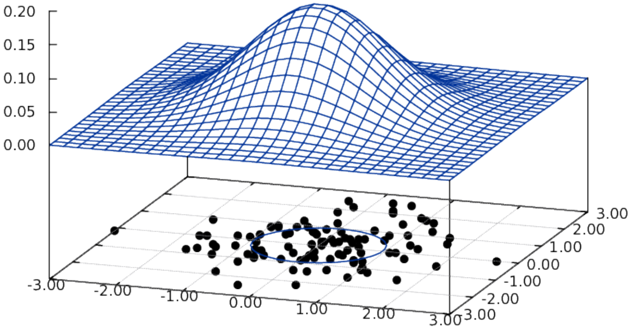
Probabilistic View of Data

- In the probabilistic view, we interpret each numeric attribute as a random variable.
- Column j in the data matrix is a sequence of n values drawn from the distribution of the corresponding random variable X_j.
- For some j, the observed values x_{ij}, $i = 1, \ldots, n$, can be seen in turn as identity random variables.
- For some j, all x_{ij}, $i = 1, \ldots, n$, can be considered mutually independent and identically distributed as X_j.
- The Central Limit Theorem therefore supports in many cases the choice of Gaussian distributions to model data adequately.
- However, several classes or clusters in the data means: there are different random variables and/or dependencies between the involved random variables.

Probabilistic View of Data

- Consequently, we can
  - reason about the distribution followed by each random variable X_j,
  - explore their independence or conditional probabilities,
  - study their joint distributions.
- Overall, we treat the data set at hand as *a sample* drawn from the joint distribution of d random variables.
- A simplifying assumption is the independence of the attributes (cf. Naïve Bayes).
- Note, however, that we can typically not assume that the attribute random variables $X_j, j = 1, \ldots, d$ are identically distributed.

Recommended Reading



Recommended Reading:

- *Compressed overview for an algebraic/geometric view on data: Zaki and Meira Jr. [2014], Chapter 1.3*
- *Compressed overview for a probabilistic view on data: Zaki and Meira Jr. [2014], Chapter 1.4*

Different Interpretations of a Single Algorithm

- We studied k-means clustering and k nearest neighbor classification.
- We introduced both algorithms from an algebraic point of view (computing distances in a vector space).
- Later on, we interpreted k nearest neighbor classification in a probabilistic framework and studied Bayesian learning.
- In this section, we will
  - re-interpret k-means clustering probabilistically and study a more complex clustering approach (EM-clustering), that follows the Bayesian framework;
  - re-interpret Bayesian learning geometrically, leading to the classification method Linear Discriminant Analysis (LDA).

Outline

Expectation Maximization

EM-clustering [Dempster et al., 1977]:

- objects are points $x = (x_1, \ldots, x_d)$ in Euclidean vector space $\mathbb{R}^d$, $\mathrm{dist} =$ Euclidean distance (L_2)
- cluster is represented by a probability density function
- typically the model is a multivariate normal distribution
- representative for a cluster C:
  - mean vector μ_k of all points in C_k
  - $d \times d$ covariance matrix Σ_k for the points in C_k
- probability density of C_k:

$$f_k(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} e^{-\frac{1}{2}(x-\mu_k)\cdot(\Sigma_k)^{-1}\cdot(x-\mu_k)^\mathsf{T}}$$

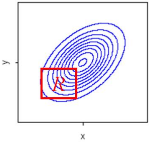
Idea of EM clustering

- each point belongs to several (that is, all) clusters, to each cluster k with a different probability $p(x|C_k)$
- algorithm consists of two alternating steps:
  - assignment of points to their clusters, not absolute but relative, according to probability
  - computation of new cluster representatives
- computation of μ_k needs to consider *relative* assignment
- what is the probability of belonging to some cluster?

Density Function to Model a Cluster

The model for a cluster C_k is a Gaussian pdf with cluster-specific mean and covariance matrix:

$$f_k(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} e^{-\frac{1}{2}(x-\mu_k)\cdot(\Sigma_k)^{-1}\cdot(x-\mu_k)^\mathsf{T}}$$



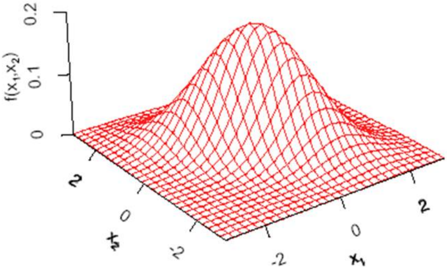
Note that:

- *To give probabilities, we would need to compute the integral over a small R around some point x to get an estimate of the probability for some hypothetical point to be located within R, or the relative share of points (e.g., 30%) of the cluster to be in this area.*
- *We therefore do not actually get probabilities here.*
- *However, the values of the p.d.f. are suitable for a relative comparison of probabilities — that is all we actually need here.*

Density Functions to Model Clusters

- For our purpose, we can use $f_k(x)$ as a *relative probability*.
- However: this would hold if some point is exclusively assigned to the cluster
- We therefore set the *conditional relative* probability as

$$p(x|C_k) \propto \frac{1}{\sqrt{(2\pi)^d|\Sigma_k|}} e^{-\frac{1}{2}(x-\mu_k)\cdot(\Sigma_k)^{-1}\cdot(x-\mu_k)^{\mathsf{T}}}$$



Mixture of Gaussians (MoG)

Assumed data generating process of a single observation x_n (a.k.a. the model):

$$p(z_n) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$$

$$p(x_n|z_n) = \prod_{k=1}^{K} \mathcal{N}(x_n|\mu_k, \Sigma_k)^{z_{nk}}$$

for one-hot-coded label $z_n \in \{0, 1\}^K$ and $\prod_{k=1}^{K} z_{nk} = 1$ denoting that $x_n \in C_k$.

Mixture of Gaussians (MoG)

We do not know z_n. It is a *latent* variable. Hence, we need to infer it. Denoting shorthand $\pi_k := p(z_{nk} = 1)$, we have

$$p(z_{nk} = 1|x_n) = \frac{\pi_k p(x_n|z_{nk} = 1)}{\sum_{j=1}^K \pi_j p(x_n|z_{nj} = 1)}$$
$$= \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

Let us compress the notation further: $p(z_{nk} = 1|x_n) \triangleq \gamma(z_{nk})$.

The Joint Distribution

Assume we have a data set consisting of N observations $\mathcal{D} = \{x_1, \dots, x_N\}$ independently sampled from the MoG data generating process, the joint distribution of all variables in the model is

$$p(\mathcal{D}, Z|\mu, \Sigma) = \prod_{n=1}^{N} p(x_n|z_n)p(z_n),$$

where

$$Z = \{z_1, \dots, z_N\},$$
$$\mu = \{\mu_1, \dots, \mu_K\},$$
$$\Sigma = \{\Sigma_1, \dots, \Sigma_K\}.$$

Marginalizing The Cluster Variables

We do not know Z and we are only able to infer a probabilistic estimation about it. We have two options:

- Either we are interested in this estimation itself. Then we are done when we calculate the posterior.
- Or we need to *account for* or *explain away* the uncertainty of our estimation and move forward to our downstream task.

The second is the case here and our downstream task is finding clusters. Then we need to *marginalize* Z by taking into account all possible outcomes of it weighted by their probability:

$$p(\mathcal{D}|\mu, \Sigma) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k).$$

Maximum Likelihood Estimation

Next, we need to find the μ, Σ values that explain the observations best, i.e. the values that *maximize* the probability of our observations (i.e. the likelihood of parameters). A function has an extremum when its derivative is zero. We wonder for which μ, Σ we have

$$\nabla_{\mu} p(\mathcal{D}|\mu, \Sigma) = 0,$$
$$\nabla_{\Sigma} p(\mathcal{D}|\mu, \Sigma) = 0.$$

Here ∇ is the *gradient* operator defined as

$$\nabla_x f(x_1, \ldots, x_M) = \left[\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_M}\right].$$

The Log-Likelihood

We need to calculate

$$\text{argmax}_{\mu,\Sigma} \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)$$

by finding μ, Σ values that satisfy

$$\nabla_{\mu,\Sigma} \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) = 0.$$

Derivative of a product is tedious: $(fgh)' = f'gh + fg'h + fgh'$.
Derivative of a sum is simpler: $(f + g + h)' = f' + g' + h'$.

The Log-Likelihood

The $\log(\cdot)$ function is monotonically increasing, hence $\forall x, y$ with $x > y$, it holds that $\log(x) > \log(y)$. Since the *order* of the values remain unchanged after log, we have

$$
\begin{aligned}
&\nabla_{\mu,\Sigma} \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \\
&= \log \nabla_{\mu,\Sigma} \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \\
&= \nabla_{\mu,\Sigma} \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \\
&= \sum_{n=1}^{N} \nabla_{\mu,\Sigma} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)
\end{aligned}
$$

Updating the means

Let us first take the gradient with respect to μ_k

$$\sum_{n=1}^{N} \nabla_{\mu_k} \log \sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)$$

$$= \sum_{n=1}^{N} \frac{\nabla_{\mu_k} \sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

$$= \sum_{n=1}^{N} \frac{\pi_k \nabla_{\mu_k} \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

$$= \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} \Sigma^{-1}(x_n - \mu_k)$$

$$= \sum_{n=1}^{N} \gamma(z_{nk}) \Sigma^{-1}(x_n - \mu_k).$$

Updating the means

Evaluate at zero

$$0 \triangleq \sum_{n=1}^{N} \gamma(z_{nk}) \Sigma^{-1} (x_n - \mu_k)$$

and solve for μ_k

$$
\begin{aligned}
0 &\triangleq \Sigma^{-1} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k) \\
&= \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)
\end{aligned}
$$

$$\Rightarrow \mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

Updating the covariance and the cluster priors

Repeating the same chain of operations for the gradient with respect to Σ_k gives

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}.$$

Denote $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$, intuitively the *effective* size of cluster k. Then the prior cluster probability can be updated as $\pi_k = N_k / N$.

The Expectation Maximization (EM) Algorithm

1. Initialize μ_k (e.g. k-means++), Σ_k (e.g. random), and π_k (e.g. uniform).
2. **E-Step.** Update class assignments

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}, \qquad N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

3. **M-Step.** Update means and variances

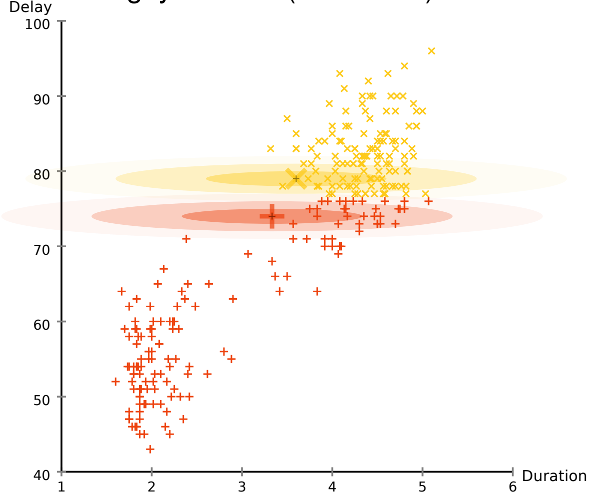
$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n,$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

4. Go to Step 2 if the last iteration changed $\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$ more than a threshold

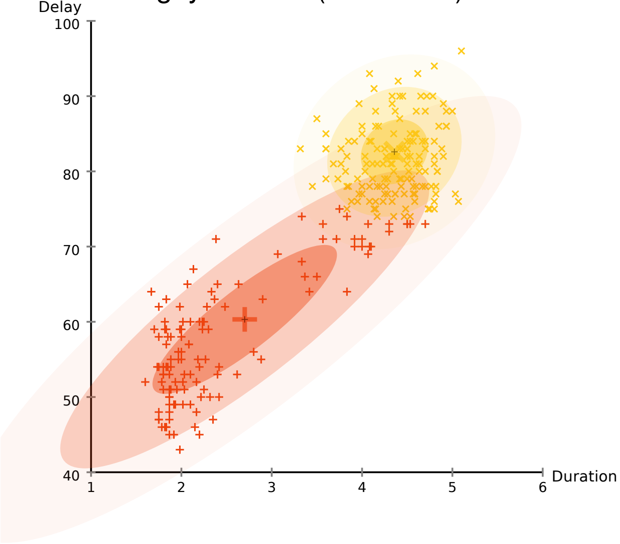
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 0):



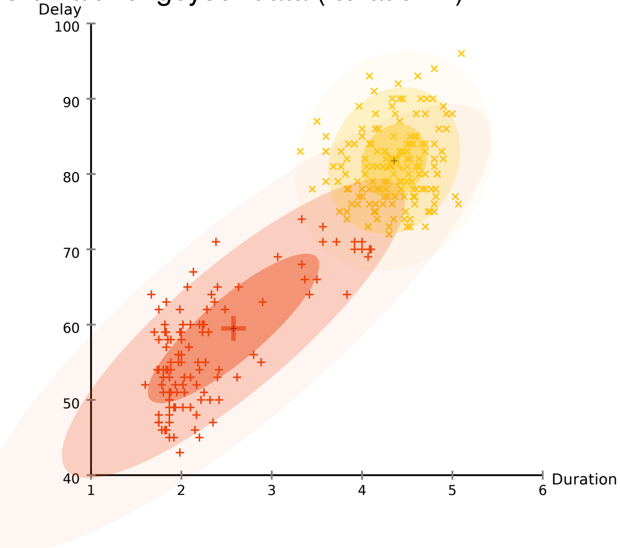
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 1):



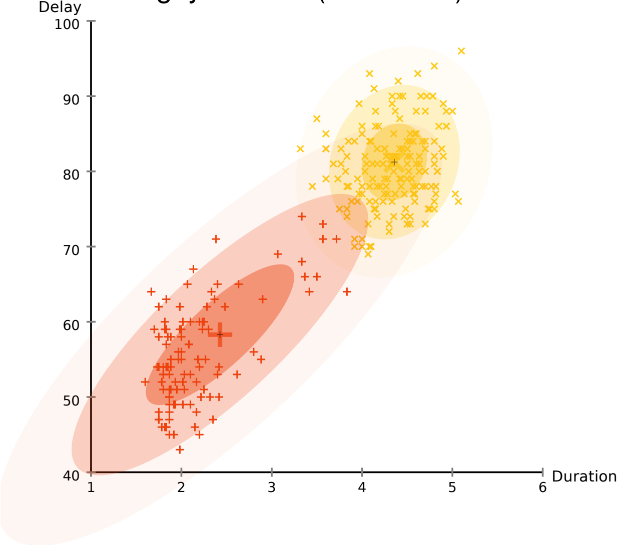
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 2):



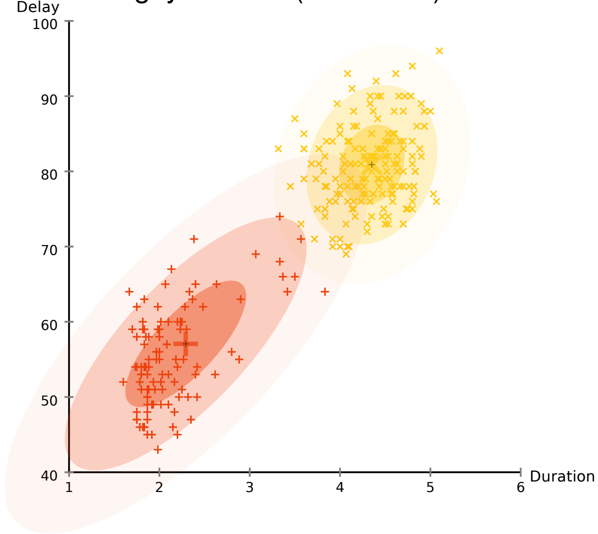
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 3):



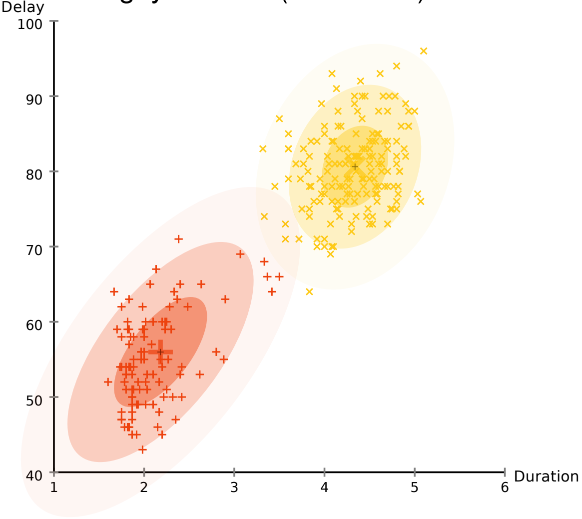
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 4):



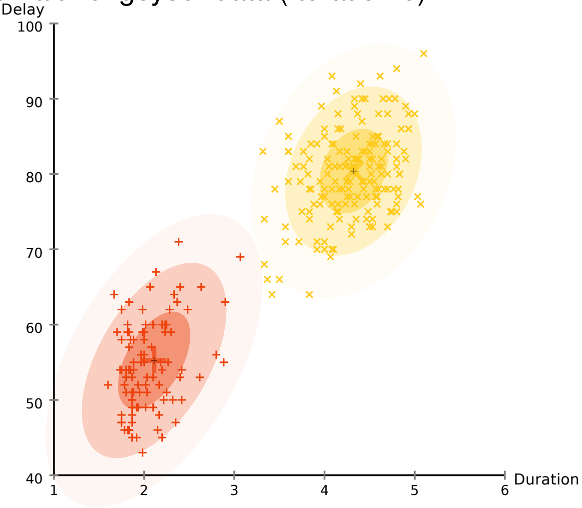
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 5):



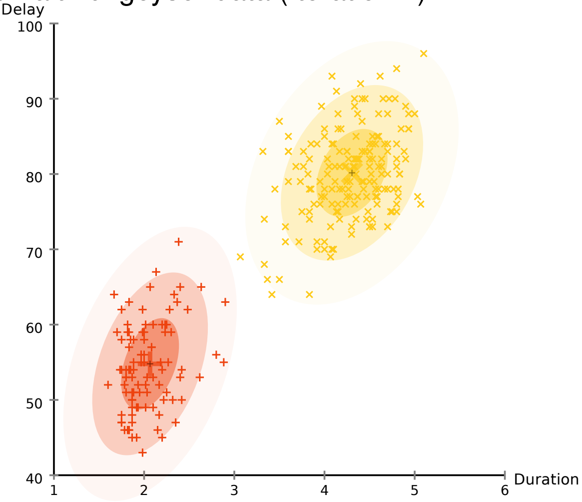
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 6):



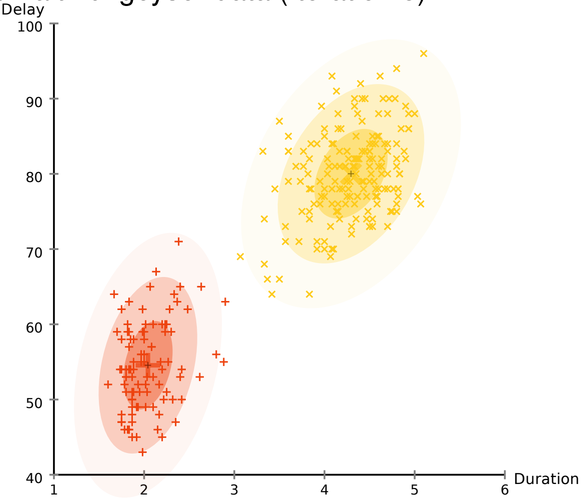
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 7):



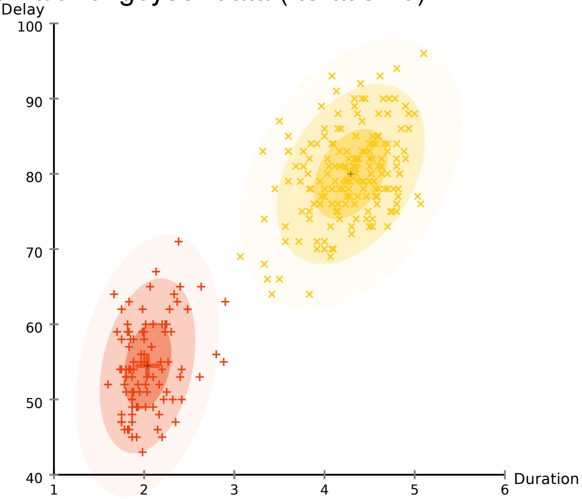
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 8):



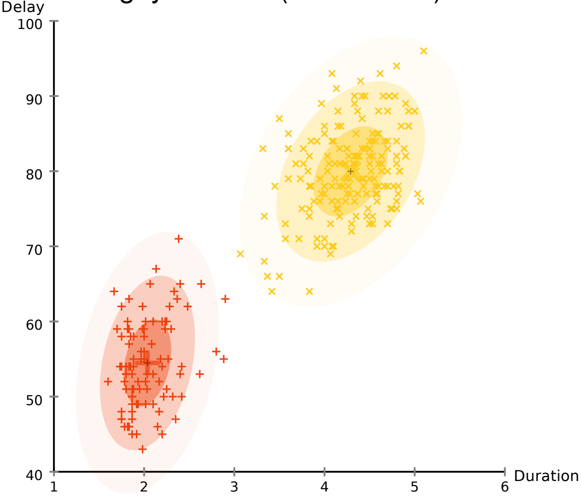
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 9):



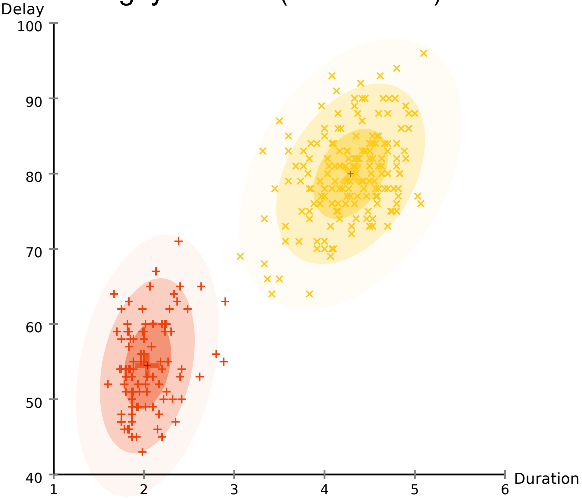
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 10):



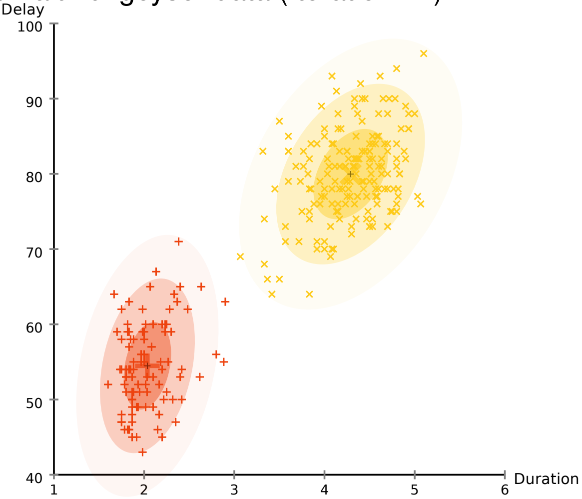
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 11):



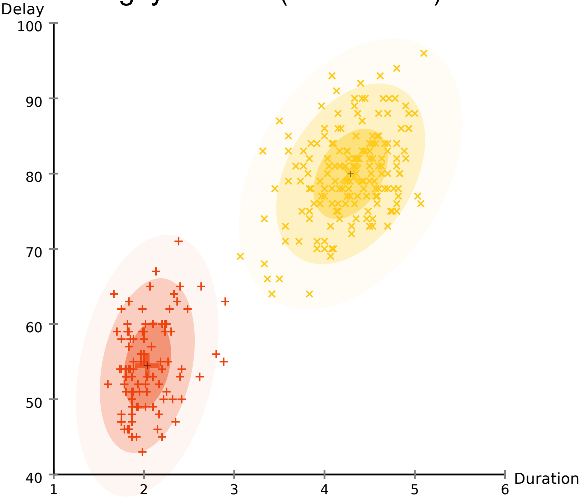
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 12):



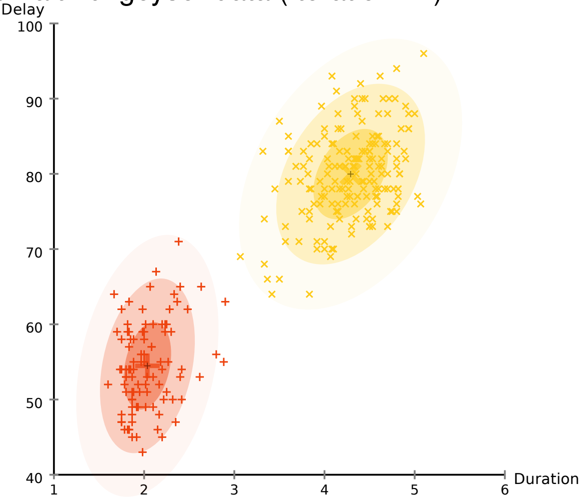
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 13):



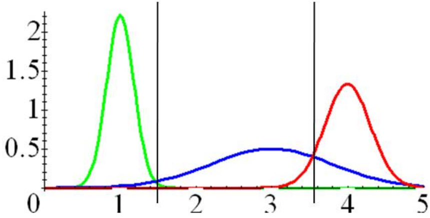
EM Example: Iterative Fitting of the Model Parameters μ_k and Σ_k

Old Faithful geyser data (Iteration 14):

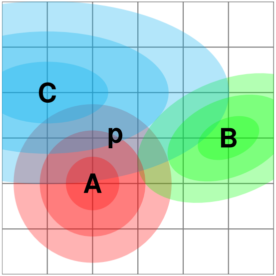


Discussion

- $\mathcal{O}(n \cdot K \cdot \#\text{iterations})$ – usually many iterations required
- Result quality and runtime depend strongly on initial model and good choice of k.
- Modification for returning K disjunct clusters: each object belongs only to the cluster with highest (relative) probability $p(x|z_{nk} = 1)$.



Multivariate Normal Distribution



Clusters can be:

- Spherical (A)
- Ellipsoid (C)
- Rotated ellipsoid (B)

But: same formula for each situation!

Probability density function of multivariate normal distribution:

$$f(x,\mu,\Sigma) := \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \cdot e^{-\frac{1}{2}\left((x-\mu)\Sigma^{-1}(x-\mu)^{\mathsf{T}}\right)}$$

Multivariate Normal Distribution

Dissecting the formula

$$f(x, \mu, \Sigma) := \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \cdot e^{-\frac{1}{2}\left((x-\mu)\Sigma^{-1}(x-\mu)^{\mathsf{T}}\right)}$$

Multivariate Normal Distribution

Dissecting the formula

$$f(x, \mu, \Sigma) := \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \cdot e^{-\frac{1}{2}\left((x-\mu)\Sigma^{-1}(x-\mu)^{\mathsf{T}}\right)}$$

Uni-variate normal distribution

$$f(x, \mu, \sigma) := \frac{1}{\sqrt{(2\pi)}\sigma} \cdot e^{-\frac{1}{2}\left(\frac{(x-\mu)}{\sigma}\right)^2}$$

Multivariate Normal Distribution

Dissecting the formula

$$f(x, \mu, \Sigma) := \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \cdot e^{-\frac{1}{2}\left((x-\mu)\Sigma^{-1}(x-\mu)^{\mathsf{T}}\right)}$$

Uni-variate normal distribution

$$f(x, \mu, \sigma) := \frac{1}{\sqrt{(2\pi)}\sigma} \cdot e^{-\frac{1}{2}\left(\frac{(x-\mu)}{\sigma}\right)^2}$$

Multivariate Normal Distribution

Dissecting the formula

$$f(x,\mu,\Sigma) := \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \cdot e^{-\frac{1}{2}\left((x-\mu)\Sigma^{-1}(x-\mu)^{\mathsf{T}}\right)}$$

Uni-variate normal distribution

$$f(x,\mu,\sigma) := \frac{1}{\sqrt{(2\pi)}\sigma} \cdot e^{-\frac{1}{2}\left(\frac{(x-\mu)}{\sigma}\right)^2}$$

normalization and squared distance from mean

Multivariate Normal Distribution

Dissecting the formula

$$f(x, \mu, \Sigma) := \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \cdot e^{-\frac{1}{2}\left((x-\mu)\Sigma^{-1}(x-\mu)^{\mathsf{T}}\right)}$$

Uni-variate normal distribution

$$f(x, \mu, \sigma) := \frac{1}{\sqrt{(2\pi)\sigma^2}} \cdot e^{-\frac{1}{2}\left((x-\mu)\sigma^{-2}(x-\mu)\right)}$$

normalization and squared distance from mean

Multivariate Normal Distribution

Dissecting the formula

$$f(x,\mu,\Sigma) := \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \cdot e^{-\frac{1}{2}\left((x-\mu)\Sigma^{-1}(x-\mu)^{\mathsf{T}}\right)}$$

Uni-variate normal distribution

$$f(x,\mu,\sigma) := \frac{1}{\sqrt{(2\pi)\sigma^2}} \cdot e^{-\frac{1}{2}\left((x-\mu)\sigma^{-2}(x-\mu)\right)}$$

Mahalanobis distance:

$$d_{\mathit{Mahalanobis}}(x,\mu,\Sigma) := \sqrt{(x-\mu)\Sigma^{-1}(x-\mu)^{\mathsf{T}}}$$

Multivariate Normal Distribution

Dissecting the formula

$$f(x, \mu, \Sigma) := \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \cdot e^{-\frac{1}{2}\left((x-\mu)\Sigma^{-1}(x-\mu)^\mathsf{T}\right)}$$

Uni-variate normal distribution

$$f(x, \mu, \sigma) := \frac{1}{\sqrt{(2\pi)\sigma^2}} \cdot e^{-\frac{1}{2}\left((x-\mu)\sigma^{-2}(x-\mu)\right)}$$

Mahalanobis distance:

$$d_{Mahalanobis}(x, \mu, \Sigma)^2 := (x-\mu)\Sigma^{-1}(x-\mu)^\mathsf{T}$$

Multivariate Normal Distribution

Dissecting the formula

$$f(x,\mu,\Sigma) := \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \cdot e^{-\frac{1}{2}\left((x-\mu)\Sigma^{-1}(x-\mu)^\mathsf{T}\right)}$$

Uni-variate normal distribution

$$f(x,\mu,\sigma) := \frac{1}{\sqrt{(2\pi)\sigma^2}} \cdot e^{-\frac{1}{2}\left((x-\mu)\sigma^{-2}(x-\mu)\right)}$$

Mahalanobis distance:

$$d_{Mahalanobis}(x,\mu,\Sigma)^2 := (x-\mu)\Sigma^{-1}(x-\mu)^\mathsf{T}$$

(compare the general quadratic form distance, Slide 119)

What is the effect of Σ^{-1}?

Multivariate normal distribution

Inverse of covariance matrix – Σ^{-1}

The covariance matrix is symmetric, and non-negative on the diagonal. Therefore it can usually be inverted (ignore constant dimensions).
Singular Value Decomposition:

$$\Sigma = V\Lambda V^{-1} \quad \Rightarrow \quad \Sigma^{-1} = V\Lambda^{-1}V^{-1} = V\Lambda^{-1}V^{T}$$

because

- V is an orthonormal matrix of eigenvectors, i.e. $VV^T = I$.
- Λ contains the eigenvalues on the diagonal and zero elsewhere.

$V \cong$ rotation, $\Lambda \cong$ scaling as in Principal Component Analysis.

Multivariate normal distribution

Inverse of covariance matrix – Σ^{-1}

Build Ω using $\omega_i = 1/\sqrt{\lambda_i} = \lambda_i^{-\frac{1}{2}}$. Then $\Omega\Omega = \Lambda^{-1}$.

$$\Sigma^{-1} = V\Omega\Omega^{\mathsf{T}}V^{\mathsf{T}} = V\Omega(V\Omega)^{\mathsf{T}}$$

$$d^2_{\text{Mahalanobis}} = (x-\mu)(x-\mu)^{\mathsf{T}}$$

norm $L_2(x-y) = \mathrm{dist}_{L_2}(x,y)$ Euclidean distance

Note that:

The Mahalanobis distance can be interpreted as Euclidean distance after having applied PCA on the data.

Multivariate normal distribution

Inverse of covariance matrix – Σ^{-1}

Build Ω using $\omega_i = 1/\sqrt{\lambda_i} = \lambda_i^{-\frac{1}{2}}$. Then $\Omega\Omega = \Lambda^{-1}$.

$$\Sigma^{-1} = V\Omega\Omega^{\mathsf{T}}V^{\mathsf{T}} = V\Omega(V\Omega)^{\mathsf{T}}$$

$$d^2_{\text{Mahalanobis}} = (x-\mu)(x-\mu)^{\mathsf{T}}$$

norm $L_2(x-y) = \operatorname{dist}_{L_2}(x,y)$ Euclidean distance

Note that:

The Mahalanobis distance can be interpreted as Euclidean distance after having applied PCA on the data.

Multivariate normal distribution

Inverse of covariance matrix – Σ^{-1}

Build Ω using $\omega_i = 1/\sqrt{\lambda_i} = \lambda_i^{-\frac{1}{2}}$. Then $\Omega\Omega = \Lambda^{-1}$.

$$\Sigma^{-1} = V\Omega\Omega^\mathsf{T} V^\mathsf{T} = V\Omega(V\Omega)^\mathsf{T}$$

$$d^2_{\text{Mahalanobis}} = (x-\mu)\Sigma^{-1}(x-\mu)^\mathsf{T}$$

norm $L_2(x-y) = \mathrm{dist}_{L_2}(x,y)$ Euclidean distance

Note that:

The Mahalanobis distance can be interpreted as Euclidean distance after having applied PCA on the data.

Multivariate normal distribution

Inverse of covariance matrix – Σ^{-1}

Build Ω using $\omega_i = 1/\sqrt{\lambda_i} = \lambda_i^{-\frac{1}{2}}$. Then $\Omega\Omega = \Lambda^{-1}$.

$$\Sigma^{-1} = V\Omega\Omega^{\mathsf{T}}V^{\mathsf{T}} = V\Omega(V\Omega)^{\mathsf{T}}$$

$$d^2_{\text{Mahalanobis}} = (x-\mu)V\Omega(V\Omega)^{\mathsf{T}}(x-\mu)^{\mathsf{T}}$$

norm $L_2(x-y) = \mathrm{dist}_{L_2}(x,y)$ Euclidean distance

Note that:

The Mahalanobis distance can be interpreted as Euclidean distance after having applied PCA on the data.

Multivariate normal distribution

Inverse of covariance matrix – Σ^{-1}

Build Ω using $\omega_i = 1/\sqrt{\lambda_i} = \lambda_i^{-\frac{1}{2}}$. Then $\Omega\Omega = \Lambda^{-1}$.

$$\Sigma^{-1} = V\Omega\Omega^{\mathsf{T}}V^{\mathsf{T}} = V\Omega(V\Omega)^{\mathsf{T}}$$

$$\begin{aligned} d^2_{\text{Mahalanobis}} &= (x-\mu)V\Omega(V\Omega)^{\mathsf{T}}(x-\mu)^{\mathsf{T}} \\ &= \langle (V\Omega)^{\mathsf{T}}(x-\mu), (V\Omega)^{\mathsf{T}}(x-\mu) \rangle \end{aligned}$$

norm $L_2(x-y) = \mathrm{dist}_{L_2}(x,y)$ Euclidean distance

Note that:

The Mahalanobis distance can be interpreted as Euclidean distance after having applied PCA on the data.

Multivariate normal distribution

Inverse of covariance matrix – Σ^{-1}

Build Ω using $\omega_i = 1/\sqrt{\lambda_i} = \lambda_i^{-\frac{1}{2}}$. Then $\Omega\Omega = \Lambda^{-1}$.

$$\Sigma^{-1} = V\Omega\Omega^{\mathsf{T}}V^{\mathsf{T}} = V\Omega(V\Omega)^{\mathsf{T}}$$

$$\begin{aligned} d^2_{\text{Mahalanobis}} &= (x-\mu)V\Omega(V\Omega)^{\mathsf{T}}(x-\mu)^{\mathsf{T}} \\ &= \langle (V\Omega)^{\mathsf{T}}(x-\mu), (V\Omega)^{\mathsf{T}}(x-\mu) \rangle \\ &= L_2((V\Omega)^{\mathsf{T}}(x-\mu))^2 \end{aligned}$$

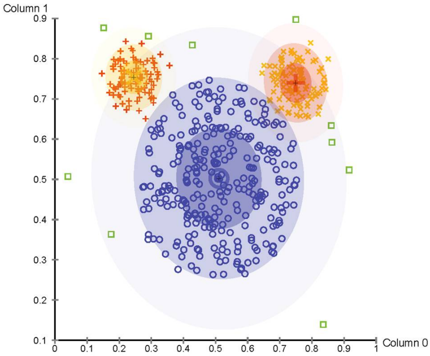
norm $L_2(x-y) = \mathrm{dist}_{L_2}(x,y)$ Euclidean distance

Note that:

The Mahalanobis distance can be interpreted as Euclidean distance after having applied PCA on the data.

Model

- (relative) assignment of points to closest cluster representative
- “closest” is given by Mahalanobis distance:
  - quadratic form distance
  - distance matrix depends on the corresponding cluster (covariance matrix of the cluster points)



Outline

Multivariate Gaussian Model for Each Class

Classification problem, given training data $TR \subseteq \mathbb{R}^d$:

- model each class k as a multivariate Gaussian:

$$f_k(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} e^{-\frac{1}{2}(x-\mu_k)\cdot(\Sigma_k)^{-1}\cdot(x-\mu_k)^{\mathsf{T}}}$$

- estimate the model parameters mean μ_k and covariance matrix (Σ_k) for each class k from the training data
- let $C_k \subseteq TR$ be the set of observations belonging to class k:

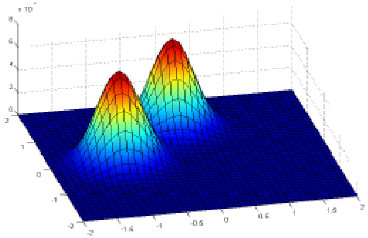
$$\mu_k = \frac{\sum_{x \in C_k} x}{|C_k|}$$

$$\left(\Sigma_k\right)_{mn} = \frac{\sum_{x \in C_k} (x_m - \mu_{km}) \cdot (x_n - \mu_{kn})}{|C_k|}$$

Maximum Likelihood Prediction

Given the Gaussian model for each class, we can predict the maximum likelihood class:

$$\begin{aligned} h(x) &= \arg \max_{k \in \{1,\ldots,K\}} \Pr(C_k|x) \\ &= \arg \max_{k \in \{1,\ldots,K\}} \Pr(x|C_k) \cdot \Pr(C_k) \\ &= \arg \max_{k \in \{1,\ldots,K\}} \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} e^{-\frac{1}{2}(x-\mu_k)\cdot(\Sigma_k)^{-1}\cdot(x-\mu_k)^{\mathsf{T}}} \cdot \Pr(C_k) \end{aligned}$$



Global Covariance Matrix

- Problem: sometimes the amount of available training examples per class is not sufficient to significantly determine the covariance matrix for each class.
- Combine the covariance matrices to a global covariance matrix:

$$(\Sigma) = \frac{\sum_{k=1}^{K} (\Sigma_k)}{|C|}$$

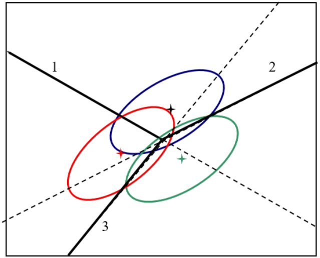
- classification rule:

$$h(x) = \arg \max_{k \in \{1,\dots,K\}} \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(x-\mu_k)\cdot(\Sigma)^{-1}\cdot(x-\mu_k)^{\mathsf{T}}} \cdot \Pr(C_k)$$

Linear Discriminant Analysis

$$
\begin{aligned}
h(x) &= \arg\max_{k\in\{1,\dots,K\}} \left\{ \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} e^{-\frac{1}{2}(x-\mu_k)\cdot(\Sigma)^{-1}\cdot(x-\mu_{c_i})^{\mathsf{T}}} \cdot \Pr(C_k) \right\} \\
&= \arg\max_{k\in\{1,\dots,K\}} \left\{ -\frac{1}{2}(x-\mu_k)\cdot(\Sigma)^{-1}\cdot(x-\mu_k)^{\mathsf{T}} + \log\Pr(C_k) \right\} \\
&= \arg\max_{k\in\{1,\dots,K\}} \left\{ x(\Sigma)^{-1}\mu_k^{\mathsf{T}} - \frac{1}{2}\mu_k\cdot(\Sigma)^{-1}\mu_k^{\mathsf{T}} + \log\Pr(C_k) \right\} \\
&= \arg\max_{k\in\{1,\dots,K\}} \sigma_k(x)
\end{aligned}
$$

- Compute only the linear discriminant function $\sigma_k(x)$ for each k, no actual probabilities.
- As only the means are different, we have a linear separation.



Outline

Parametric vs. Non-Parametric Learning

parametric learning Given a data sample, we assume it follows a specific family of distributions (e.g., Gaussian).

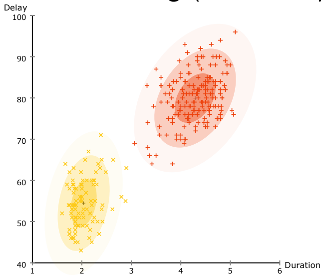
- We fit a Gaussian distribution to the data sample.
- "'Fitting" a distribution means, we learn the parameters of the distribution (in the example of a Gaussian, we learn μ and Σ).

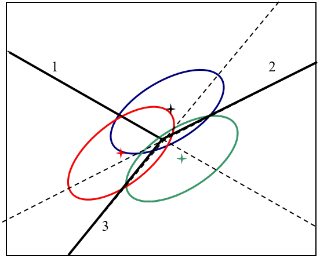
non-parametric learning Given a data sample, we try to infer probability estimates directly from the sample.

- We might approximate the shape of a density distribution without knowing its family.
- We do not necessarily need to derive an *explicit* density function in order to estimate probabilities.

Examples: Parametric Learning

- k-means (estimate μ_i of k Gaussians with unit variance and zero covariance)
- EM-clustering (estimate μ_i and Σ_i of k Gaussians)

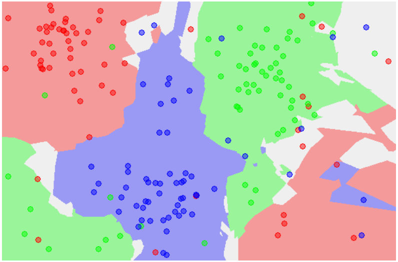




- LDA (estimate μ_i and Σ of k Gaussians and of the complete data set)

Examples: Non-Parametric Learning

- k nearest neighbor classifier (approximate class distribution boundaries, i.e., the shapes of implicit density distribution functions)



- naïve Bayes classifier (approximate probability estimates without deriving an explicit density function)

In both cases, we assume there is an underlying probability density distribution function f. We approximate this function f only locally but do not conclude on its shape overall.

Outline

The Problem of Density Estimation

The point-wise, local approximation of a probability density distribution function is more generally treated as non-parametric “density estimation”.

Recommended Reading:

- *The classic text on non-parametric density estimation is by Silverman [1986].*
- *Deep technical coverage by Duda et al. [2001], Chapter 4.*
- *Concise and illustrative introduction by Zaki and Meira Jr. [2014], Chapter 15.2. (Their material is used here.)*

Problem Statement

- We are given a data set consisting of the n points $x_1, \ldots, x_n \in \mathbb{R}^d$.
- We assume that these points are a sample of values assumed by some random variable $X \sim f(x)$ (probabilistic point of view).
- We do not know the underlying probability density distribution function f in general.
- We would like to know the value $f(x)$ for some given x.

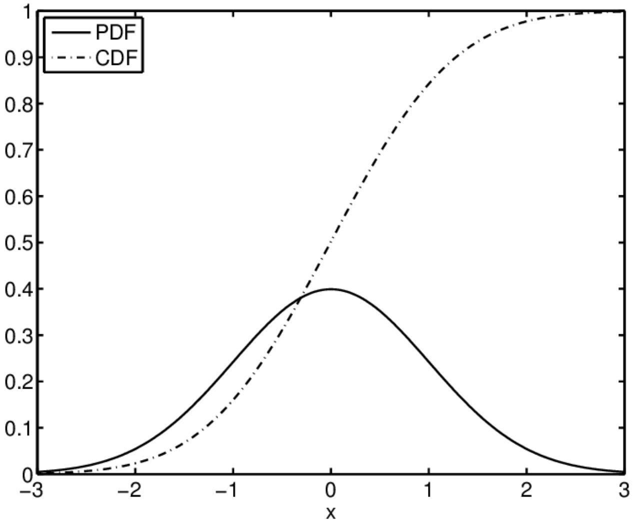
Estimation of the Cumulative Distribution Function

- In a one-dimensional scenario (i.e., $x_i \in \mathbb{R}$), the cumulative density distribution function can be directly estimated from the data.
- We have to count the points that are less than or equal to x:

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} I(x_i \leq x)$$

with $I(b) = \begin{cases} 1 & \text{if } b \\ 0 & \text{if } \neg b \end{cases}$

CDF and PDF



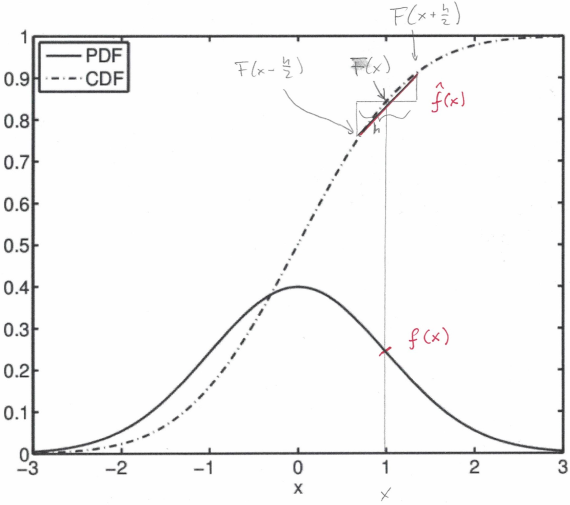
Estimation of the Probability Density Function Around x from the CDF

- From the estimate $\hat{F}$ of F, we can estimate the density function f by the derivative of $\hat{F}$.
- We take a small window of width h, centered at x:

$$\hat{f}(x) = \frac{\hat{F}\left(x+\frac{h}{2}\right) - \hat{F}\left(x-\frac{h}{2}\right)}{h} = \frac{k/n}{h} = \frac{k}{nh} \tag{7.1}$$

  where k is the number of points located within the window of size h around x.
- The density estimate is therefore given as the ratio of the fraction of points within the window $\frac{k}{n}$ to the volume h of the window.
- Choice of h:
  - Too large will consider many neighbor points and smoothen the estimated function.
  - Too small will not take enough points and does not give an accurate estimate.

CDF and PDF


PDF
CDF
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
−3
−2
−1
0
1
2
3
x
F(x + h/2)
F(x − h/2)
F(x)
h
f̂(x)
f(x)
x

Kernel Estimator

- Kernel density estimation uses a *kernel function* K.
- A kernel function is
  - non-negative, i.e., $\forall x : K(x) \geq 0$,
  - symmetric, i.e., $\forall x : K(-x) = K(x)$, and
  - integrates to 1, i.e., $\int K(x)\, dx = 1$.
- Therefore K qualifies as a probability density function itself.

Discrete Kernel

- The discrete kernel simply counts the number of points in a window of width h:

$$K(z) = \begin{cases} 1 & \text{if } |z| \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

- If $|z| = \left|\frac{x-x_i}{h}\right| \leq \frac{1}{2}$, the point x_i is within the window of width h centered at x.
- We can rewrite the density estimate from Eq. 7.1 by using the discrete kernel:

$$\hat{f}(x) = \frac{k}{nh} = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)$$

Kernel Density Estimation with Discrete Kernel

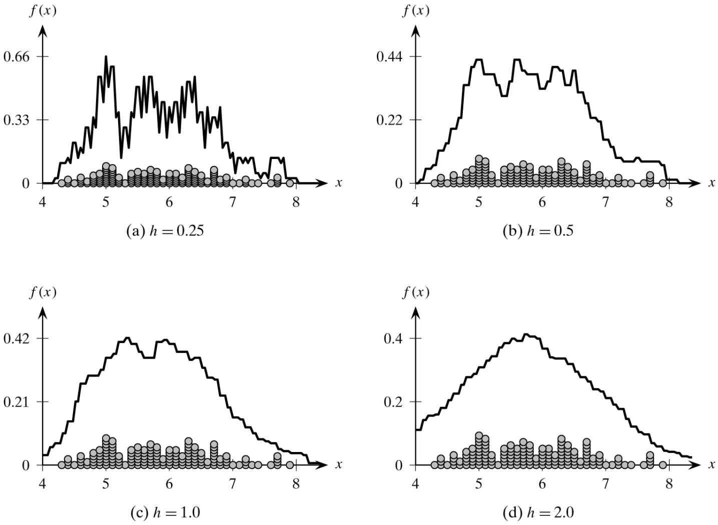
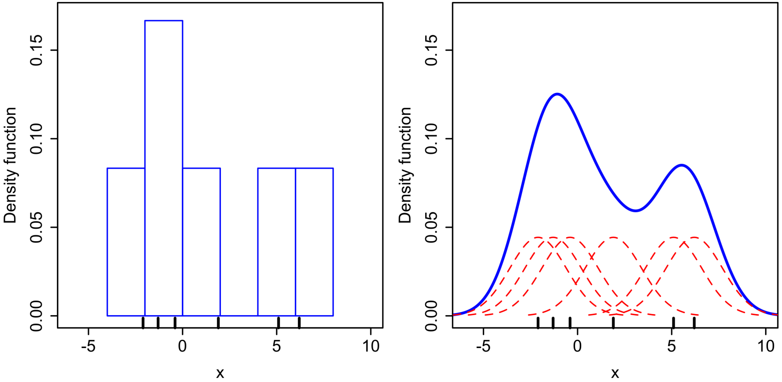


Figure from Zaki and Meira Jr. [2014].

SDU
SYDDANSK UNIVERSITET

CDF and PDF



Gaussian Kernel

- In the discrete kernel, we have an all-or-nothing contribution of the points around x to the estimate:
  - If some point falls in the window around x, it contributes $\frac{1}{hn}$ to the probability estimate $\hat{f}(x)$.
  - If some point is outside the window, it contributes 0.
- A Gaussian kernel results in a smoother change of the influence of some point:

$$K(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$$

$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}}$$

- The center of the window, x, relates to the mean, window size h relates to the standard deviation.

KDE with Gaussian Kernel

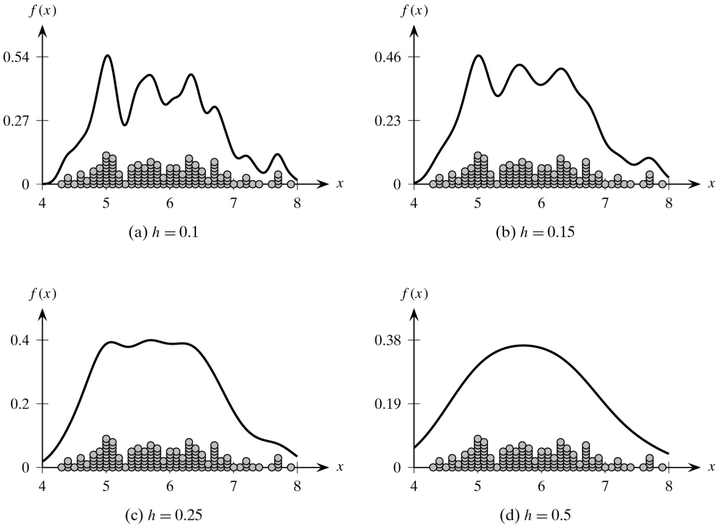


Figure from Zaki and Meira Jr. [2014].

Multivariate Density Estimation

- To estimate the density around a d-dimensional point $x \in \mathbb{R}^d$, we replace the one-dimensional window by a d-dimensional hypercube.
- Instead of the window length h, we need to account for the volume h^d.
- With some multivariate kernel function K, the density estimate becomes:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

Multivariate Kernel Function

- The discrete kernel function generalizes directly to a d-dimensional vector space:

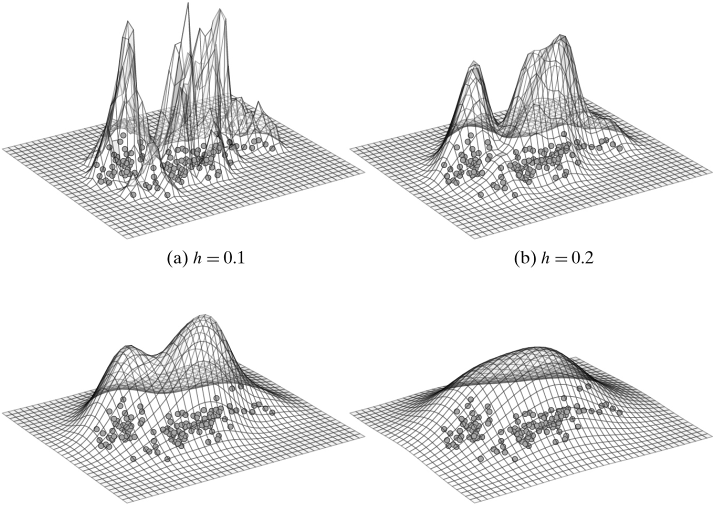
$$K(z) = \begin{cases} 1 & \text{if } |z_j| \leq \frac{1}{2} \text{ for all dimensions } j = 1, \ldots, d \\ 0 & \text{otherwise} \end{cases}$$

- For the d-dimensional Gaussian kernel, we assume that the covariance matrix is the identity matrix:

$$K(z) = \frac{1}{(2\pi)^{\frac{d}{2}}} e^{-\frac{zz^\mathsf{T}}{2}}$$

$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{(2\pi)^{\frac{d}{2}}} e^{-\frac{(x-x_i)(x-x_i)^\mathsf{T}}{2h^2}}$$

Two-dimensional Gaussian KDE



(a) $h = 0.1$ (b) $h = 0.2$ (c) $h = 0.35$ (d) $h = 0.6$

Figure from Zaki and Meira Jr. [2014].

Example Data – Gaussian KDE, $h = 20$

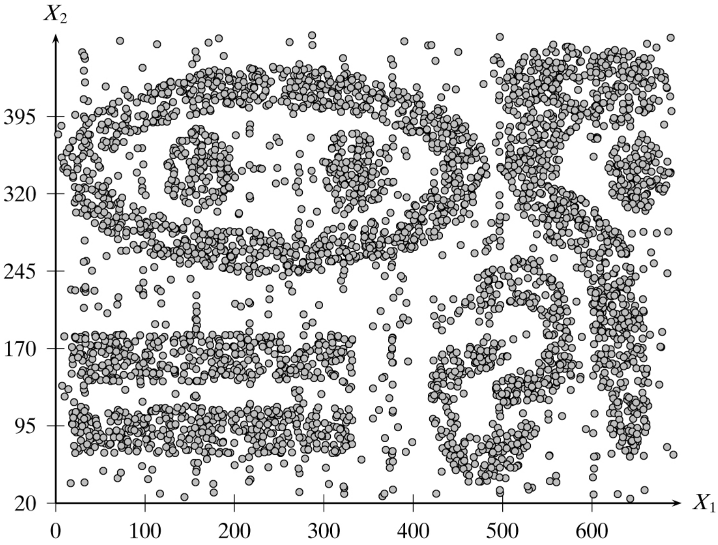


Figure from Zaki and Meira Jr. [2014].

Example Data – Gaussian KDE, $h = 20$

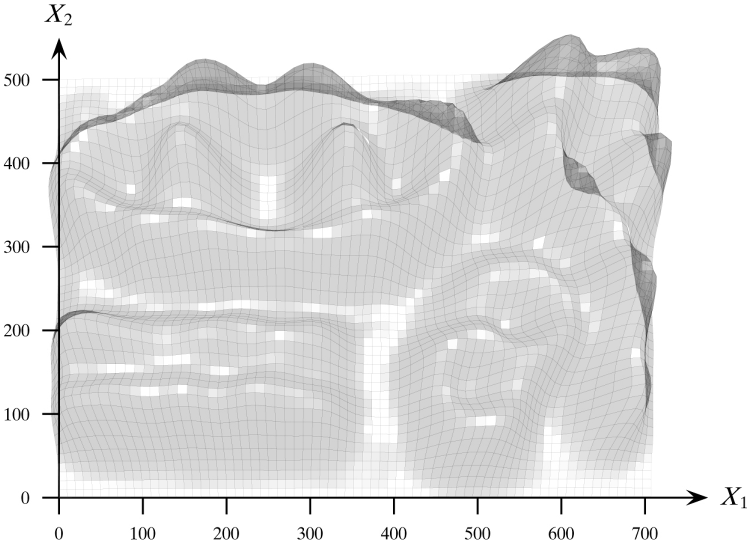


Figure from Zaki and Meira Jr. [2014].

Nearest Neighbor Density Estimation

- So far we kept the volume fixed by a fixed window width (or hypercube edge length) h.
- Problem: how to choose h in order to capture enough, but not too many points?
- Alternative: fix the number k of points, allow the volume to vary in order to capture k points.
- With Euclidean distance, we get the volume $V_k(x)$ of a d-dimensional hypersphere with the distance from x to its kth nearest neighbor as radius.
- The resulting k nearest neighbor density estimation is:

$$\hat{f}(x) = \frac{k}{nV_k(x)}$$

- Cf. the probabilistic reasoning on the $k$nn classifier (slide 291)!

Outline

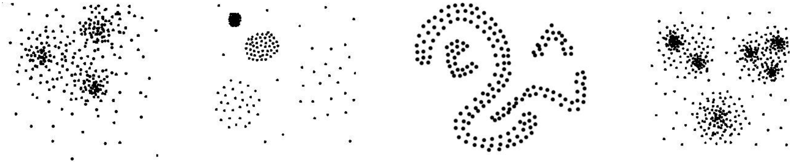
Recommended Reading

Recommended Reading:

- *The original paper on DBSCAN [Ester et al., 1996].*
- *Survey on density-based clustering [Campello et al., 2020].*
- *Tan et al. [2006], Chapters 8.4, 9.3, 9.4.5–9.4.8.*
- *Tan et al. [2020], Chapters 5.4, 8.3, 8.4.6–8.4.9.*

General Idea

- Clusters are high-density “areas” (volumes) in a d dimensional space.
- Clusters are separated by low-density “areas” (volumes).
- The high-density area defined by the set of objects forming a cluster is connected in the data space.



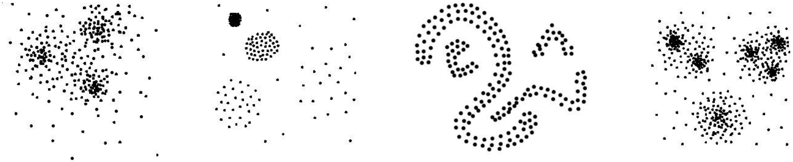
Central Assumption for Density-based Clusters

Assumption 7.1 (Density-based Cluster)

For each object in a cluster, the local density exceeds a given threshold:

$$\forall o \in C : \phi(o) \geq \theta$$

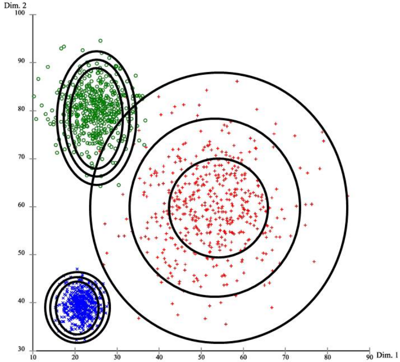
Cf. density-contour by Hartigan [1975] and discussion by Kriegel et al. [2011], Campello et al. [2015].

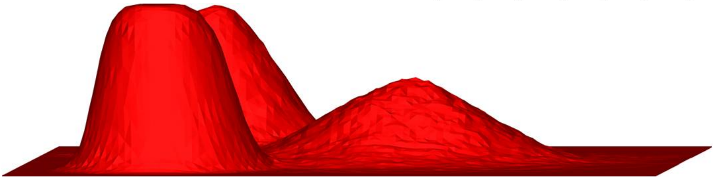


Non-Parametric Clustering: Intuition [Kriegel et al., 2011]

parametric learning approximate some density function of a given family (e.g. Gaussian)

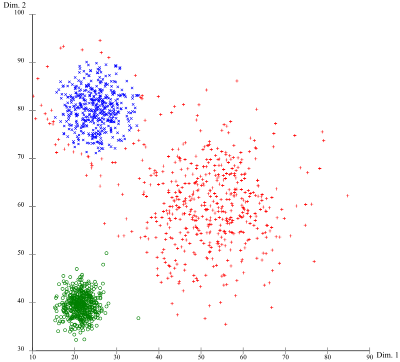
non-parametric no assumption on family of density function, identify areas of high density

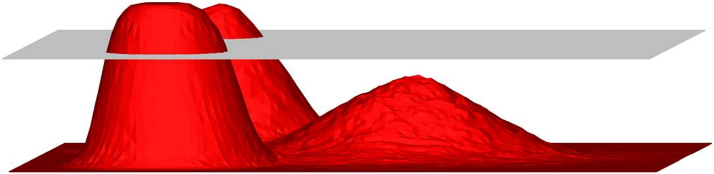




Intuition: High Density Threshold

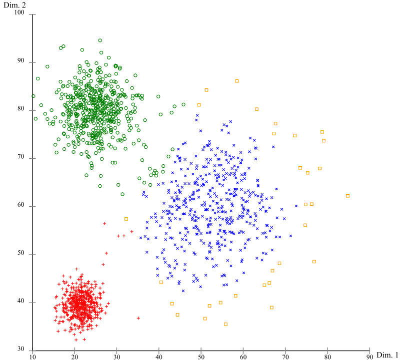
- high density threshold: cluster of lower density is regarded as noise

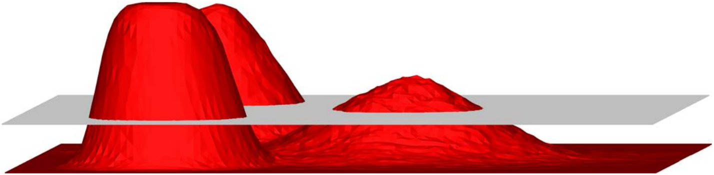




Intuition: Medium Density Threshold

- medium density threshold: three clusters are identified, some points in the tails are regarded as noise





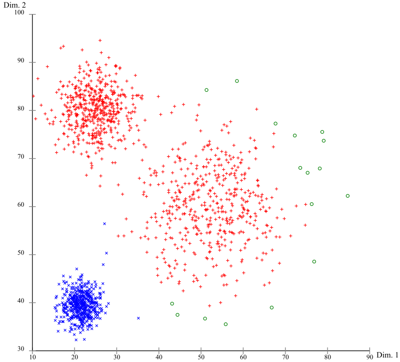
SDU SYDDANSK UNIVERSITET

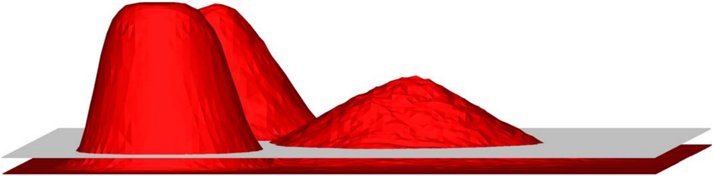
Intuition: Low Density Threshold

DM566

Melih Kandemir

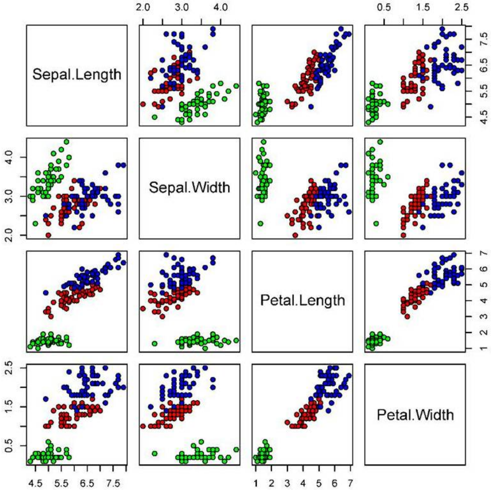
- low density threshold: two clusters are merged





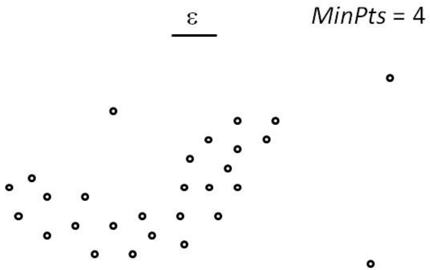
Intuition: “Natural” Clusters

- Example: different species of iris
- Within a species, we have a certain variation but no jumps



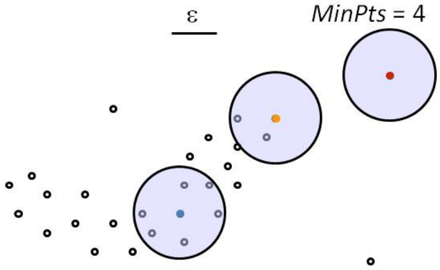
Density-based Clustering: Basic Concepts

A density threshold is defined by parameters ε and MinPts



A density threshold is defined by parameters ε and `MinPts`

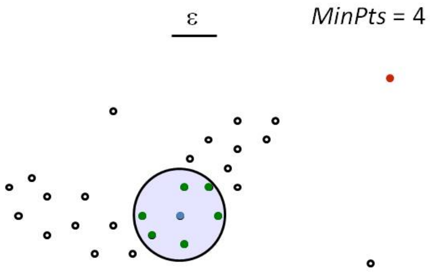
- core point



Density-based Clustering: Basic Concepts

A density threshold is defined by parameters ε and `MinPts`

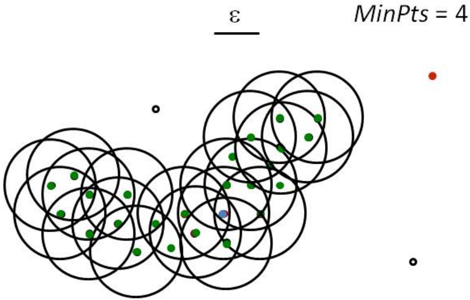
- core point
- direct density-reachability



Density-based Clustering: Basic Concepts

A density threshold is defined by parameters ε and `MinPts`

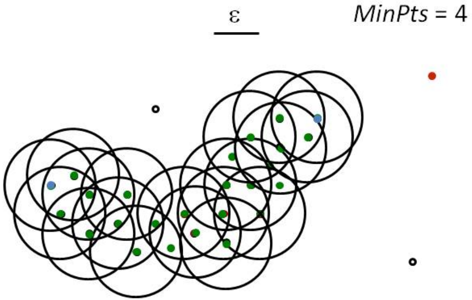
- core point
- direct density-reachability
- density-reachability



Density-based Clustering: Basic Concepts

A density threshold is defined by parameters ε and `MinPts`

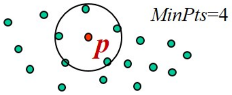
- core point
- direct density-reachability
- density-reachability
- density-connectivity



Formalization [Ester et al., 1996]

$$\mathrm{RQ}(p,\varepsilon) = \{o \in \mathcal{D} \mid \mathrm{dist}(p,o) \leq \varepsilon\}$$

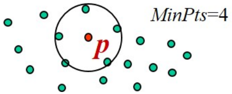
- $p \in \mathcal{D}$ is a *core point* if: $|\mathrm{RQ}(p,\varepsilon)| \geq \mathrm{MinPts}$



Formalization [Ester et al., 1996]

$$\mathrm{RQ}(p,\varepsilon) = \{o \in \mathcal{D} \,|\, \mathrm{dist}(p,o) \leq \varepsilon\}$$

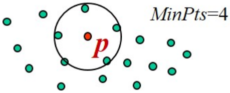
- $p \in \mathcal{D}$ is a *core point* if: $|\mathrm{RQ}(p,\varepsilon)| \geq \mathrm{MinPts}$
- $q \in \mathcal{D}$ is directly density-reachable from $p \in \mathcal{D}$ w.r.t. ε and MinPts if: $q \in \mathrm{RQ}(p,\varepsilon) \,\wedge\, |\mathrm{RQ}(p,\varepsilon)| \geq \mathrm{MinPts}$

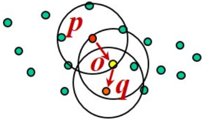


Formalization [Ester et al., 1996]

$$\mathrm{RQ}(p, \varepsilon) = \{o \in \mathcal{D} \mid \mathrm{dist}(p, o) \leq \varepsilon\}$$

- $p \in \mathcal{D}$ is a *core point* if: $|\mathrm{RQ}(p, \varepsilon)| \geq \mathrm{MinPts}$
- $q \in \mathcal{D}$ is directly density-reachable from $p \in \mathcal{D}$ w.r.t. ε and MinPts if: $q \in \mathrm{RQ}(p, \varepsilon) \wedge |\mathrm{RQ}(p, \varepsilon)| \geq \mathrm{MinPts}$
- $q \in \mathcal{D}$ is density-reachable from $p \in \mathcal{D}$ w.r.t. ε and MinPts if there is a chain of points $p_1, \ldots, p_n$, $p_1 = p$, $p_n = q$, and p_{i+1} is directly density-reachable from p_i

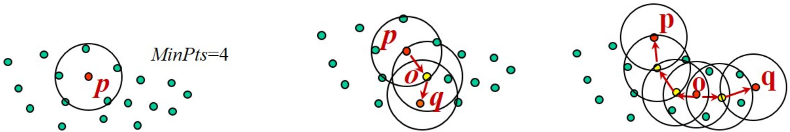




Formalization [Ester et al., 1996]

$$\mathrm{RQ}(p,\varepsilon) = \{o \in \mathcal{D} \,|\, \mathrm{dist}(p,o) \leq \varepsilon\}$$

- $p \in \mathcal{D}$ is a *core point* if: $|\mathrm{RQ}(p,\varepsilon)| \geq \mathrm{MinPts}$
- $q \in \mathcal{D}$ is directly density-reachable from $p \in \mathcal{D}$ w.r.t. ε and MinPts if: $q \in \mathrm{RQ}(p,\varepsilon) \wedge |\mathrm{RQ}(p,\varepsilon)| \geq \mathrm{MinPts}$
- $q \in \mathcal{D}$ is density-reachable from $p \in \mathcal{D}$ w.r.t. ε and MinPts if there is a chain of points $p_1, \ldots, p_n$, $p_1 = p$, $p_n = q$, and p_{i+1} is directly density-reachable from p_i
- $p \in \mathcal{D}$ is density-connected to $q \in \mathcal{D}$ w.r.t. ε and MinPts if $\exists o \in \mathcal{D}$ such that both, p and q are density-reachable from o w.r.t. ε and MinPts.



Formalization [Ester et al., 1996]

A *(density-based) cluster* C w.r.t. ε and MinPts is a non-empty subset of $\mathcal{D}$ satisfying the following conditions:

1. *maximality*: $\forall p, q :$ if $p \in C$ and q is density-reachable from p w.r.t. ε and MinPts, then $q \in C$.
2. *connectivity*: $\forall p, q \in C$: p is density-connected to q w.r.t. ε and MinPts.

Formalization [Ester et al., 1996]

A *(density-based) cluster* C w.r.t. ε and MinPts is a non-empty subset of $\mathcal{D}$ satisfying the following conditions:

1. *maximality*: $\forall p, q$: if $p \in C$ and q is density-reachable from p w.r.t. ε and MinPts, then $q \in C$.
2. *connectivity*: $\forall p, q \in C$: p is density-connected to q w.r.t. ε and MinPts.

Note that:

This fundamental property enables efficient search for clusters!

DBSCAN [Ester et al., 1996]

Algorithm 7.1 (DBSCAN [Ester et al., 1996])

```
DBSCAN(DB,Eps,MinPts)
  // DB is UNCLASSIFIED
  C_Id := nextId(NOISE);
  FOR i FROM 1 TO DB.size DO
    Point := DB.get(i);
    IF Point.C_Id = UNCLASSIFIED THEN
      IF ExpandCluster(DB, Point, C_Id, Eps, MinPts) THEN
        C_Id := nextId(C_Id)
```

DBSCAN [Ester et al., 1996]

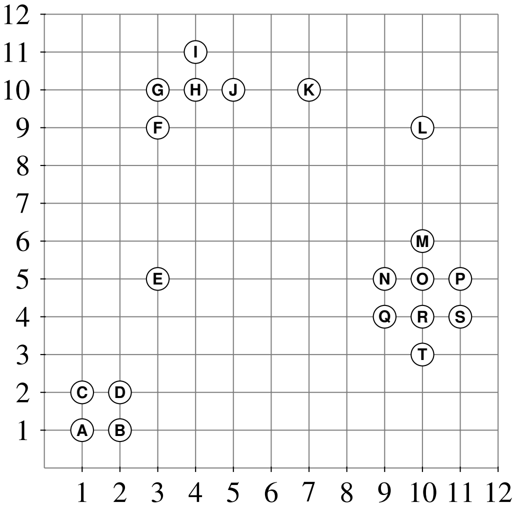
Algorithm 7.2 (DBSCAN – expandCluster)

```
ExpandCluster(DB, Point, C_Id, Eps, MinPts):  Boolean
  seeds := DB.rq(Point,Eps)
  IF seeds.size < MinPts THEN // no core point
    DB.changeC_Id(Point,NOISE);
    RETURN FALSE;
  ELSE // all points in seeds are dens-reach from Point
    DB.changeC_Ids(seeds,C_Id);
    seeds.delete(Point);
  WHILE seeds <> Empty DO
    currentP := seeds.first();
    result := DB.rq(currentP,Eps);
    IF result.size >= MinPts THEN
      FOR i FROM 1 TO result.size DO
        resultP := result.get(i);
        IF resultP.C_Id IN {UNCLASSIFIED,NOISE} THEN
          IF resultP.CiId = UNCLASSIFIED THEN
            seeds.append(resultP);
          DB.changeC_Id(resultP,C_Id);
    seeds.delete(currentP);

  RETURN TRUE;
```

Example 1

Start example Skip example



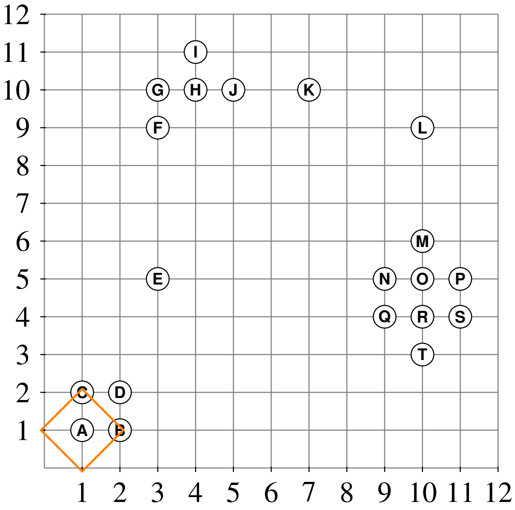
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example · Skip example



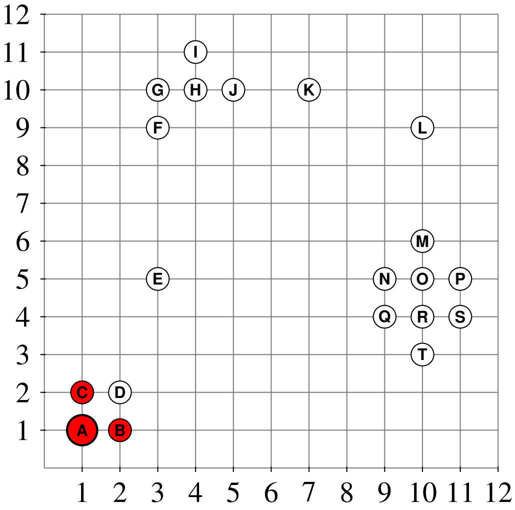
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



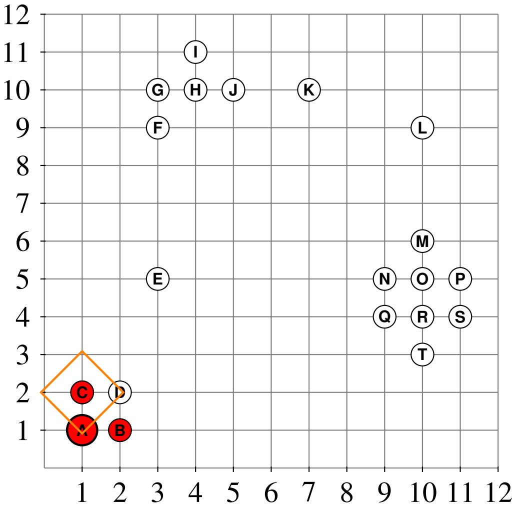
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
BC

Example 1

Start example · Skip example



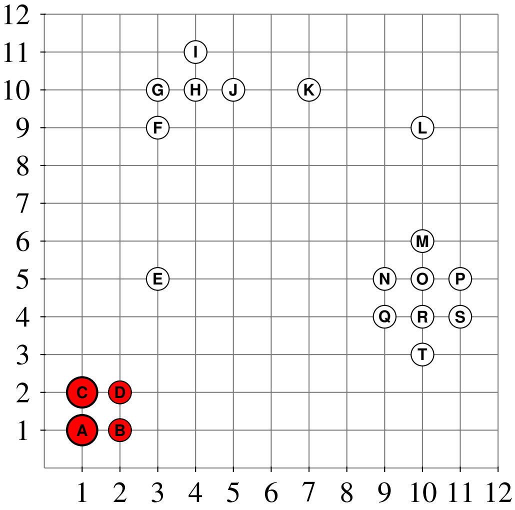
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
B

Example 1

Start example | Skip example



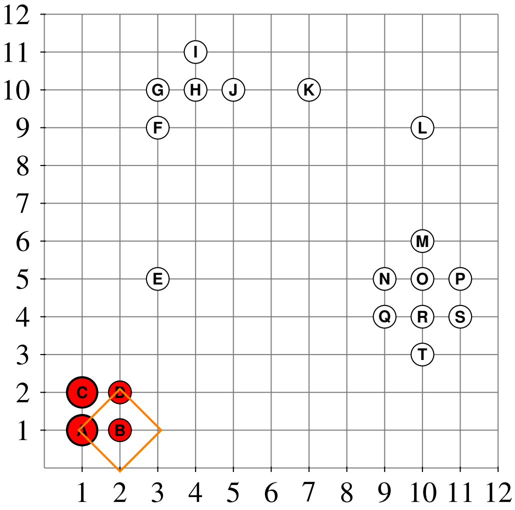
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
BD

Example 1

▸ Start example ⏩ Skip example



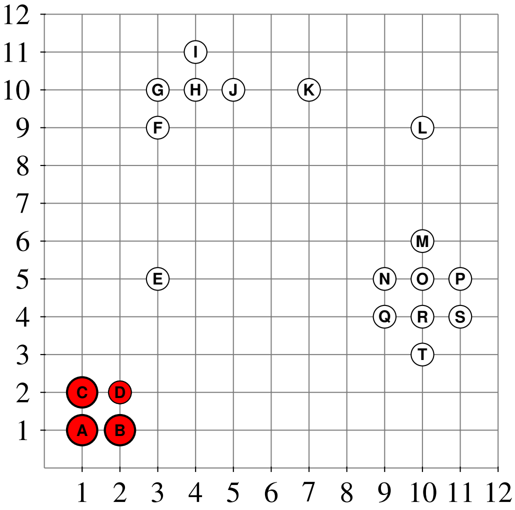
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
D

Example 1

Start example | Skip example



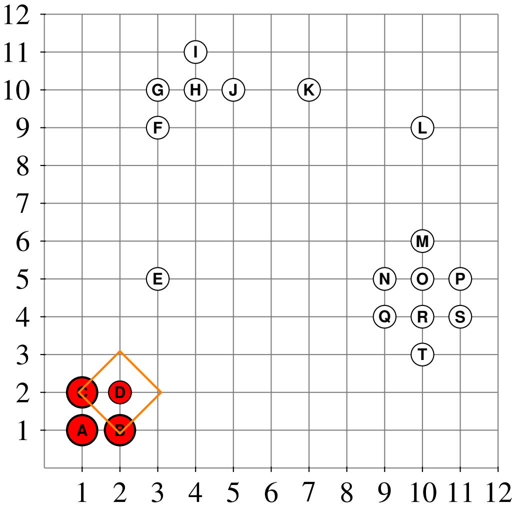
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
D

Example 1

Start example
Skip example



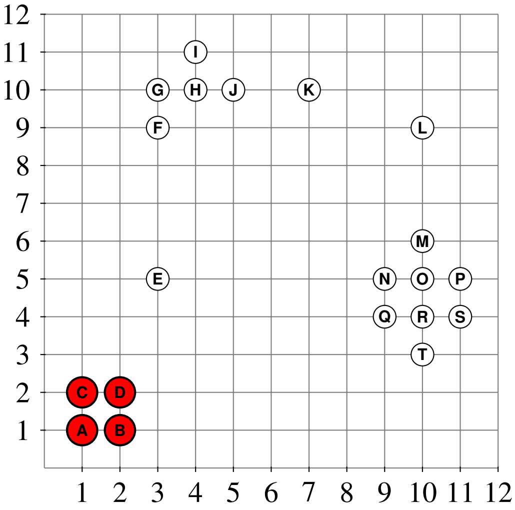
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



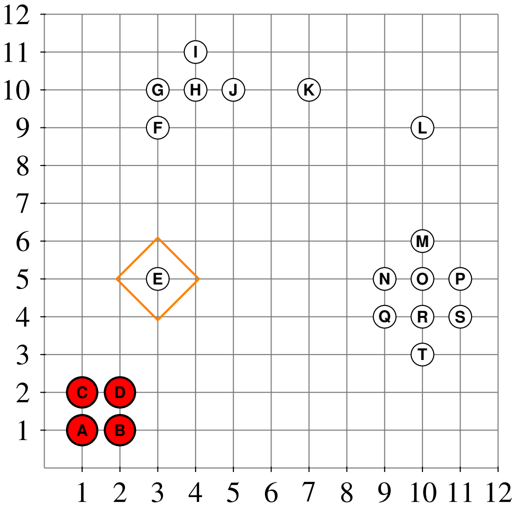
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



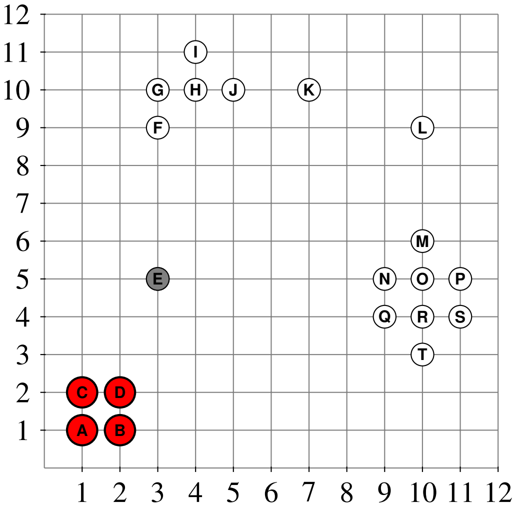
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example   Skip example



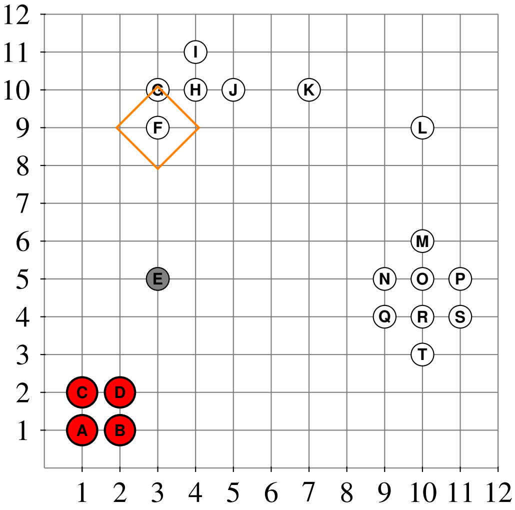
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



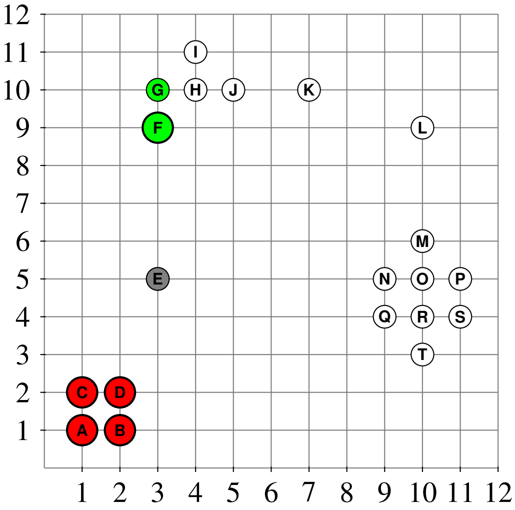
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example | Skip example



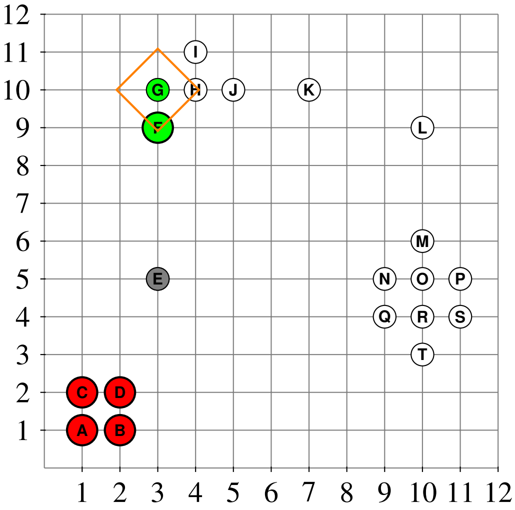
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
G

Example 1

Start example
Skip example



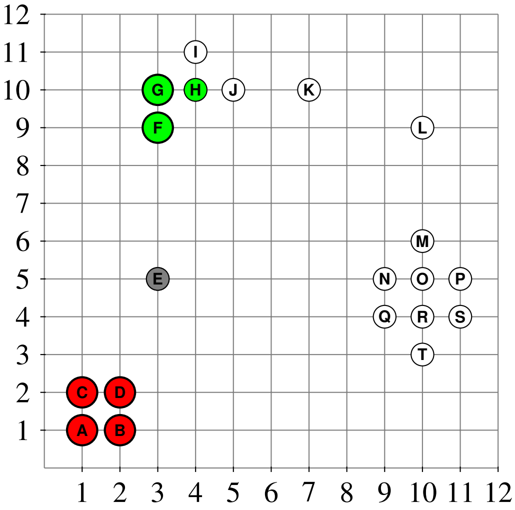
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



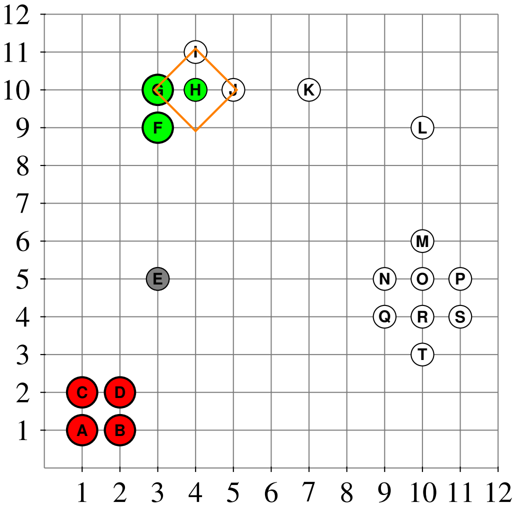
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
H

Example 1

Start example   Skip example



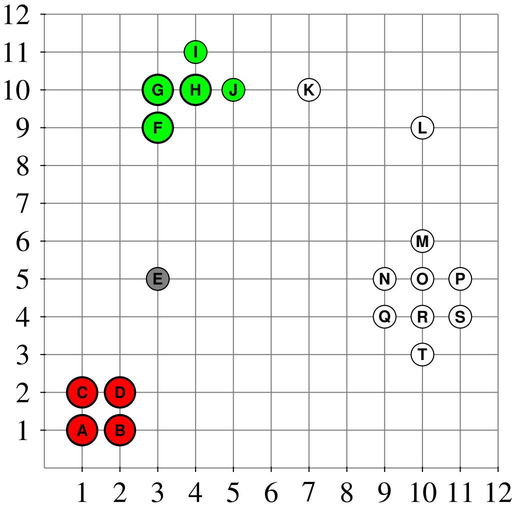
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example · Skip example

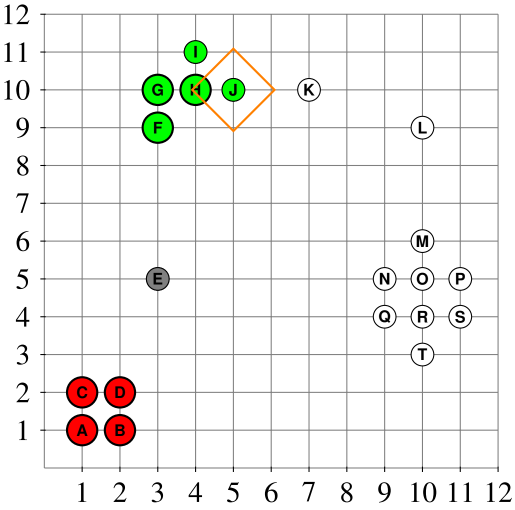


$\varepsilon = 1.1$

$minPts = 2$

Seed list:
IJ

Example 1



$\varepsilon = 1.1$

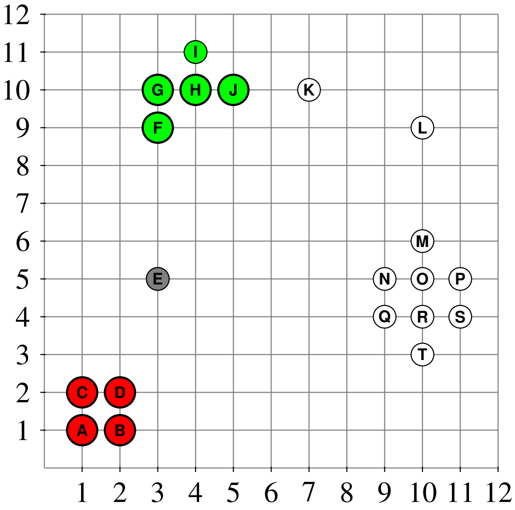
$minPts = 2$

Seed list:

I

Example 1

Start example Skip example



$\varepsilon = 1.1$

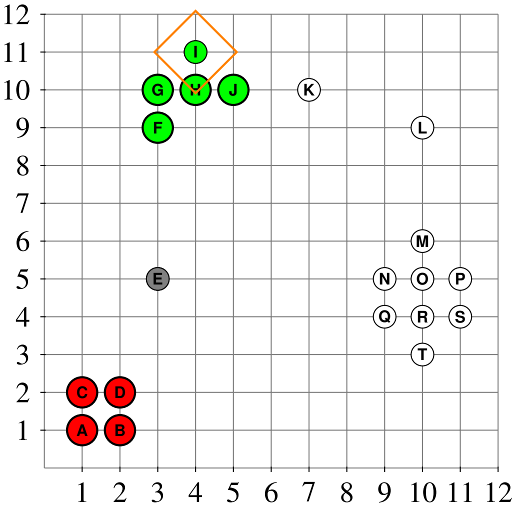
$minPts = 2$

Seed list:

I

Example 1

Start example | Skip example



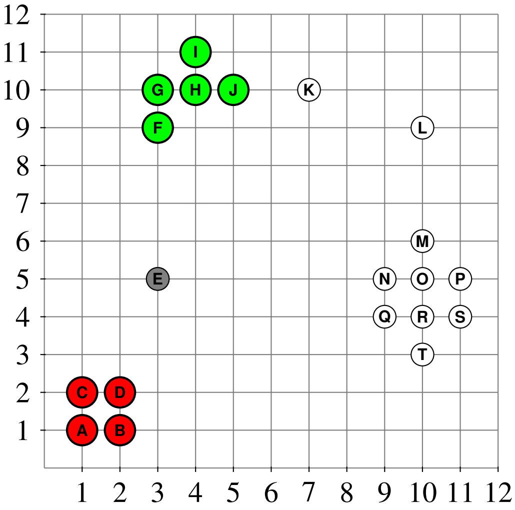
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



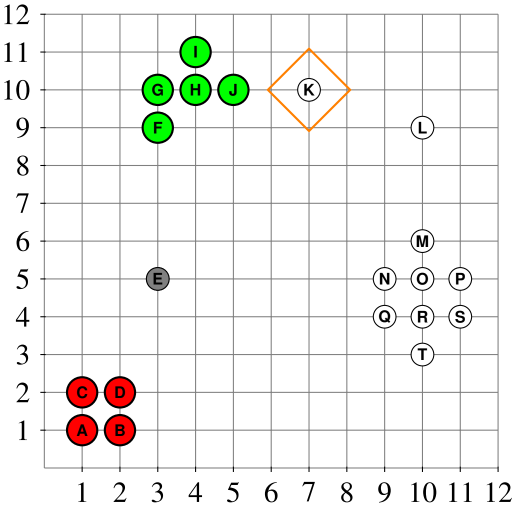
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



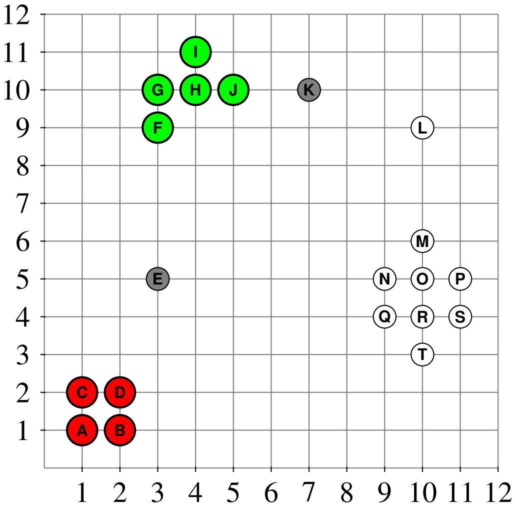
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



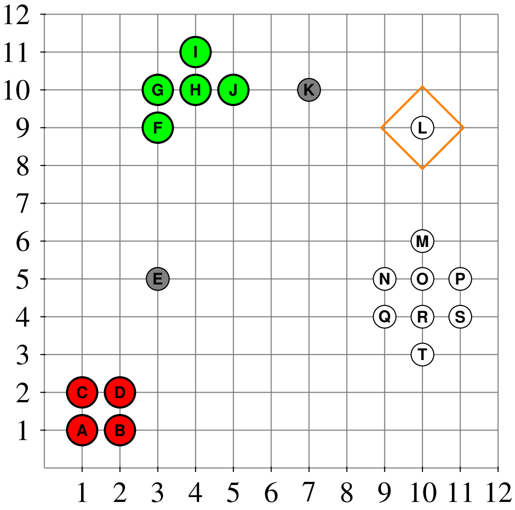
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



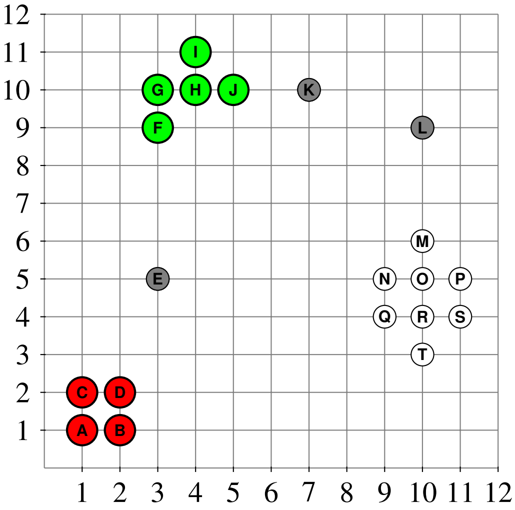
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



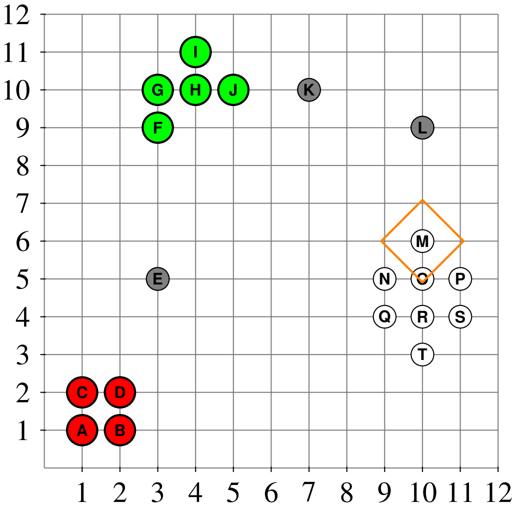
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example Skip example



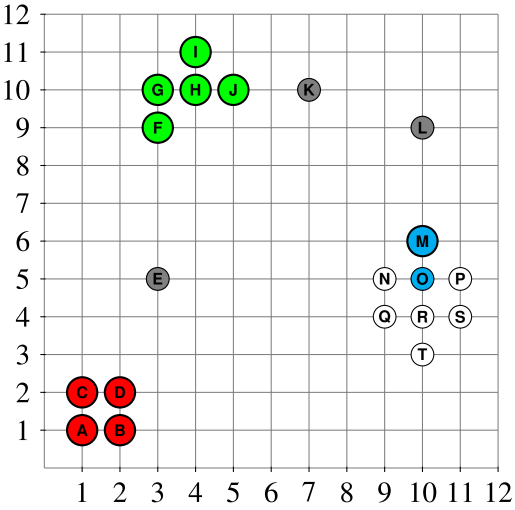
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



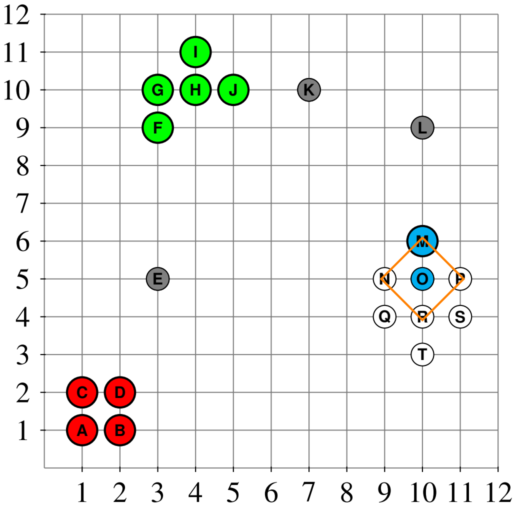
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
O

Example 1

Start example
Skip example



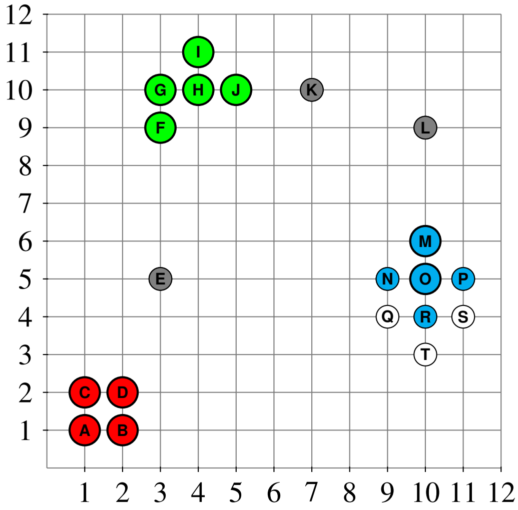
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example
Skip example



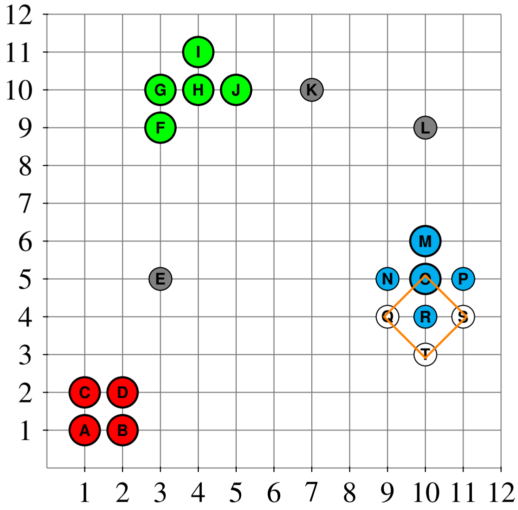
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
NPR

Example 1

Start example Skip example



$\varepsilon = 1.1$

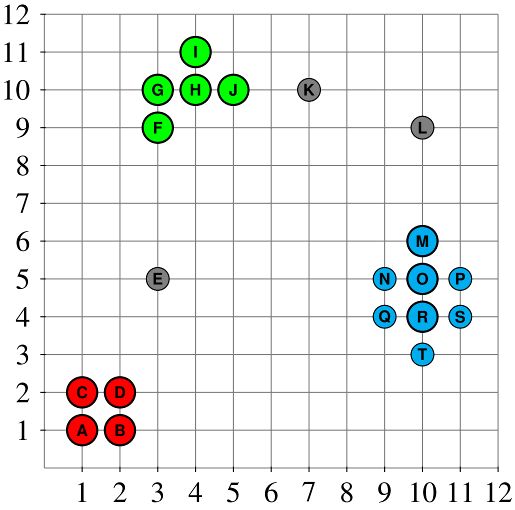
$minPts = 2$

Seed list:
NP

Example 1

Start example

Skip example



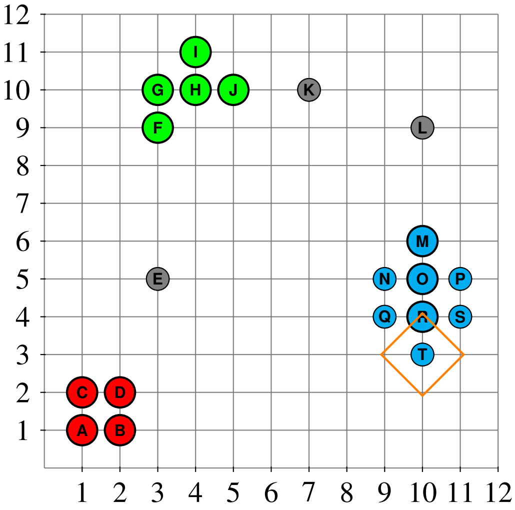
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
NPQST

Example 1

Start example | Skip example



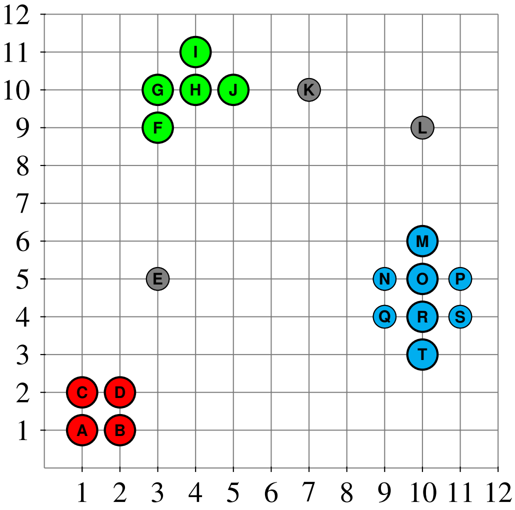
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
NPQS

Example 1

Start example | Skip example



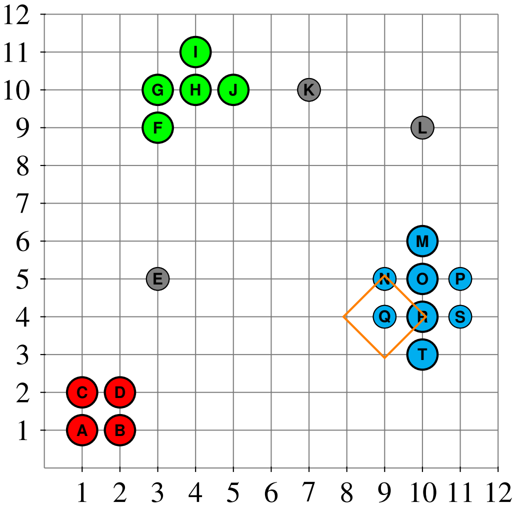
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
NPQS

Example 1

Start example · Skip example



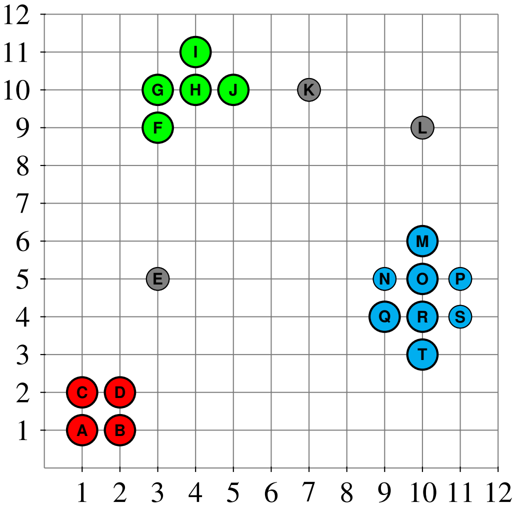
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
NPS

Example 1

Start example · Skip example



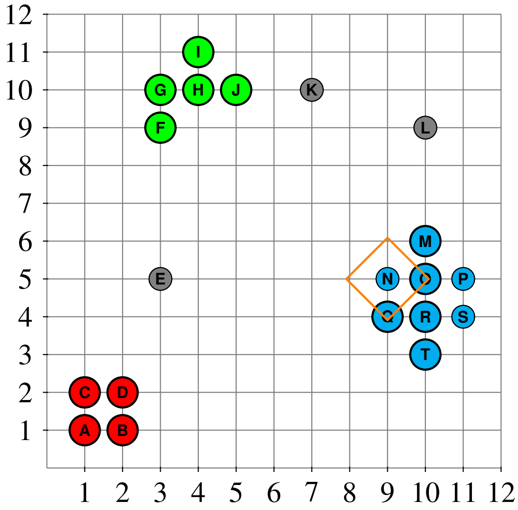
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
NPS

Example 1

Start example · Skip example



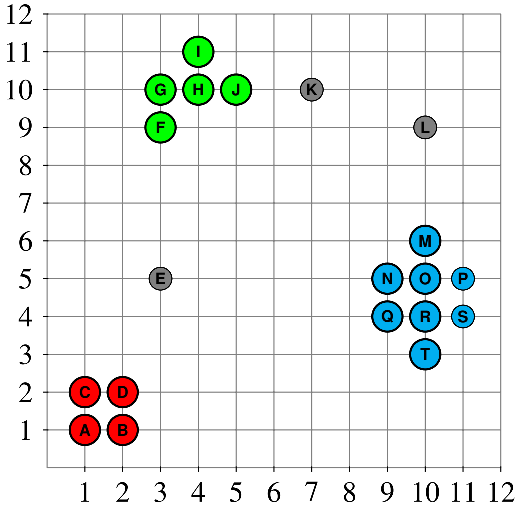
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
PS

Example 1

Start example
Skip example



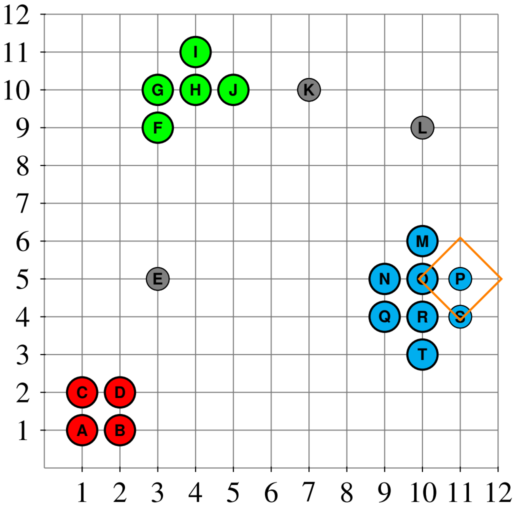
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
PS

Example 1

Start example
Skip example



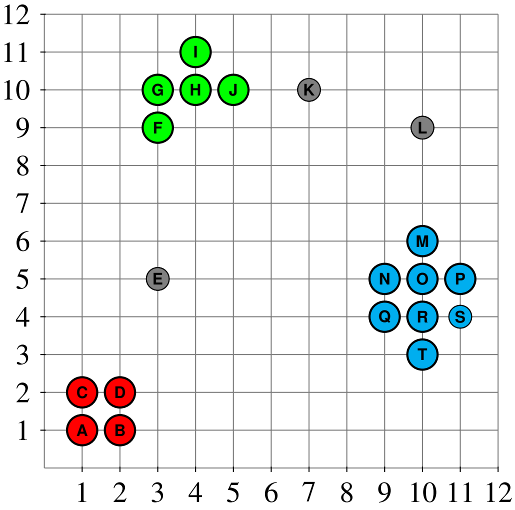
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
S

Example 1

Start example | Skip example



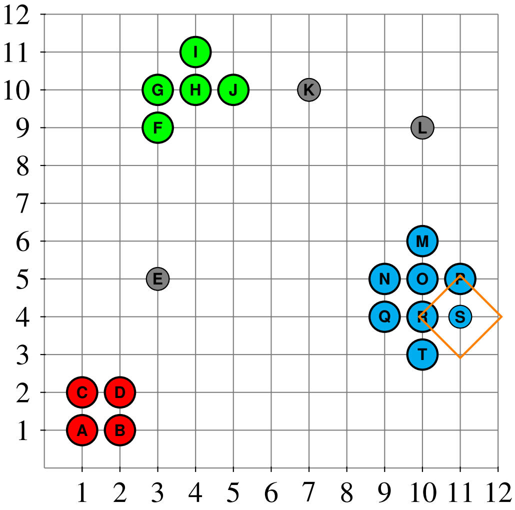
$\varepsilon = 1.1$

$minPts = 2$

Seed list:
S

Example 1

Start example Skip example



$\varepsilon = 1.1$

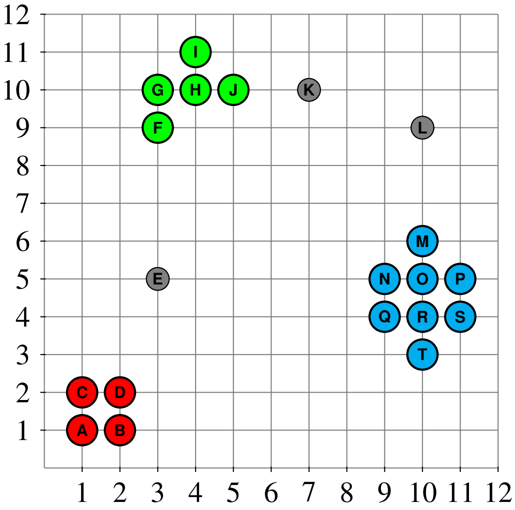
$minPts = 2$

Seed list:

SDU
SYDDANSK UNIVERSITET

Example 1

▸ Start example ▸▸ Skip example



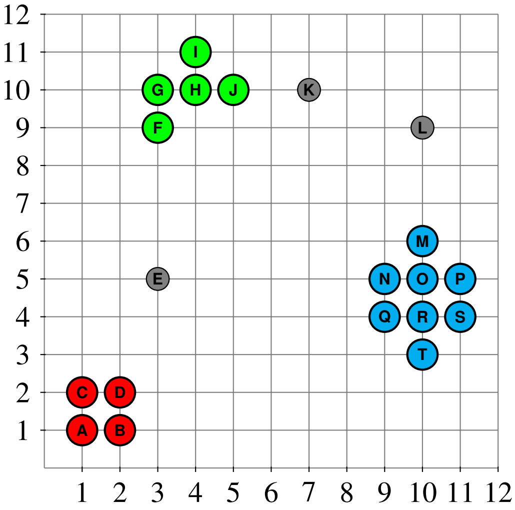
$\varepsilon = 1.1$

$minPts = 2$

Seed list:

Example 1

Start example · Skip example



$\varepsilon = 1.1$

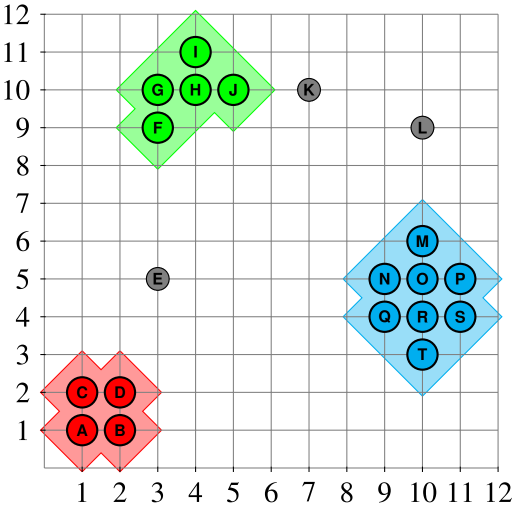
$minPts = 2$

Seed list:

No border points!

Example 1

Start example
Skip example



$\varepsilon = 1.1$

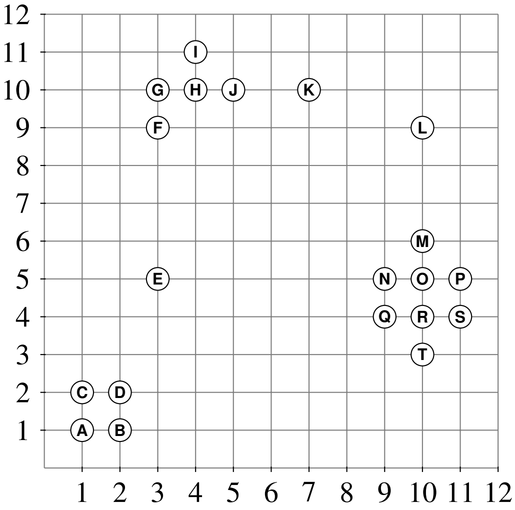
$minPts = 2$

Seed list:

No border points!

Example 2

Start example | Skip example



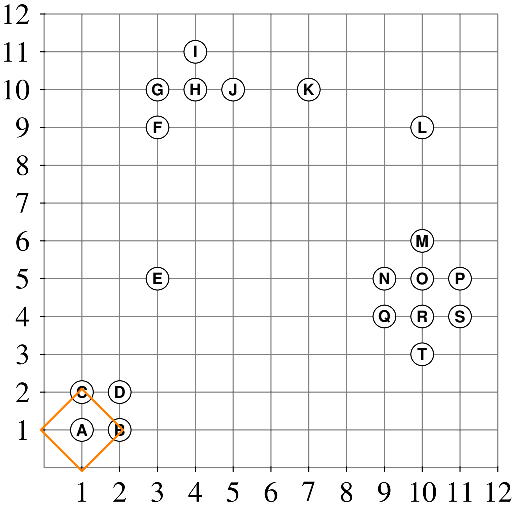
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example
Skip example



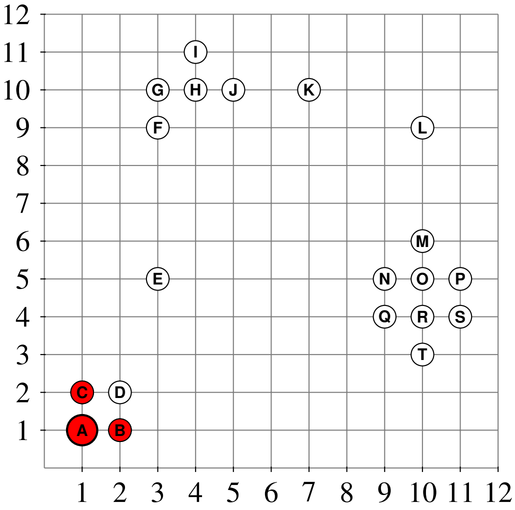
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example | Skip example



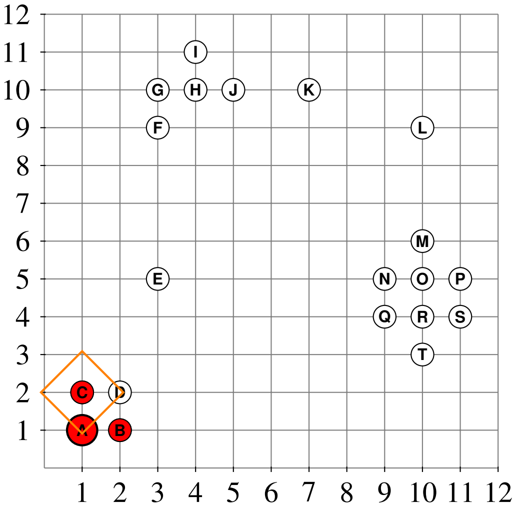
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
BC

Example 2

Start example | Skip example



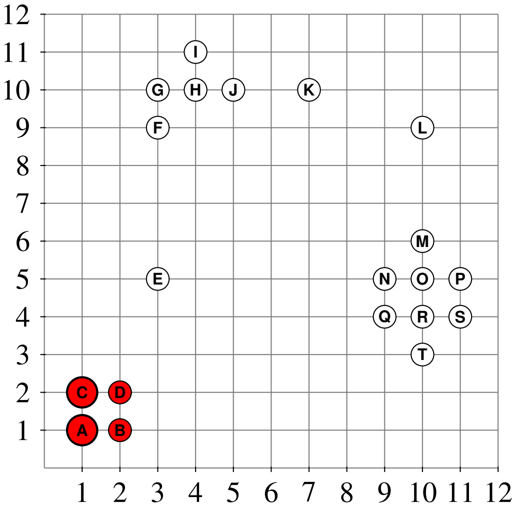
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
B

Example 2

Start example
Skip example



$\varepsilon = 1.1$

$minPts = 3$

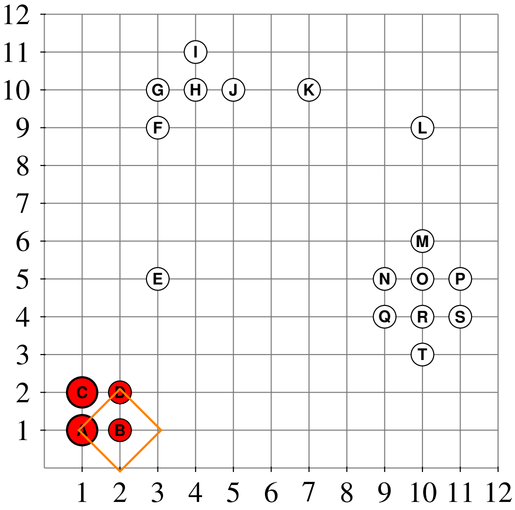
Seed list:
BD

SDU SYDDANSK UNIVERSITET

Example 2

Start example

Skip example



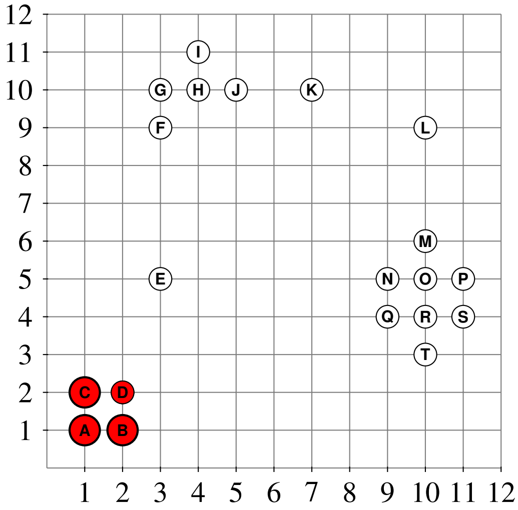
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
D

Example 2

Start example   Skip example



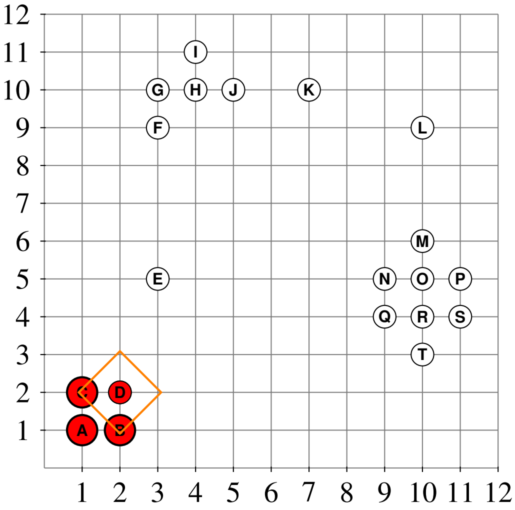
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
D

Example 2

Start example
Skip example



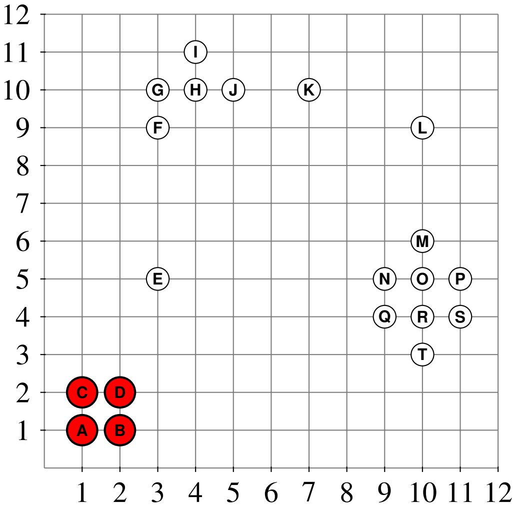
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example | Skip example



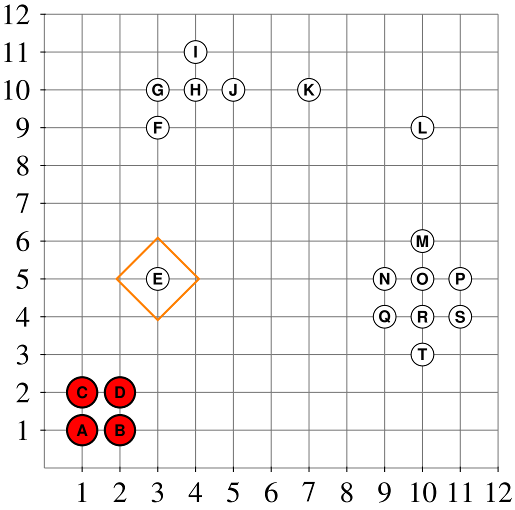
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example | Skip example



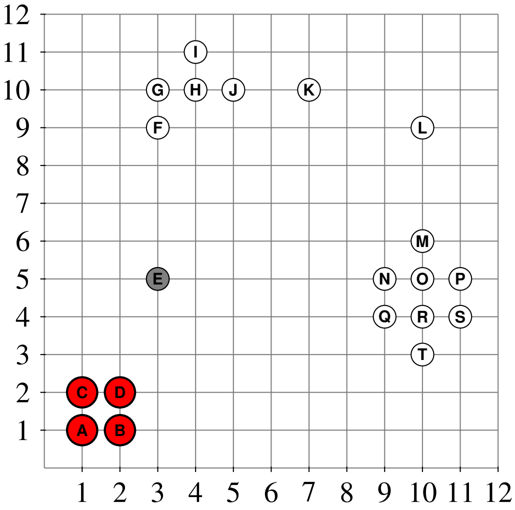
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example Skip example



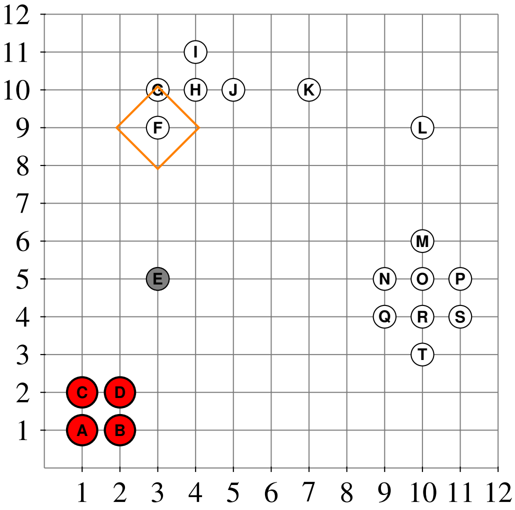
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example | Skip example



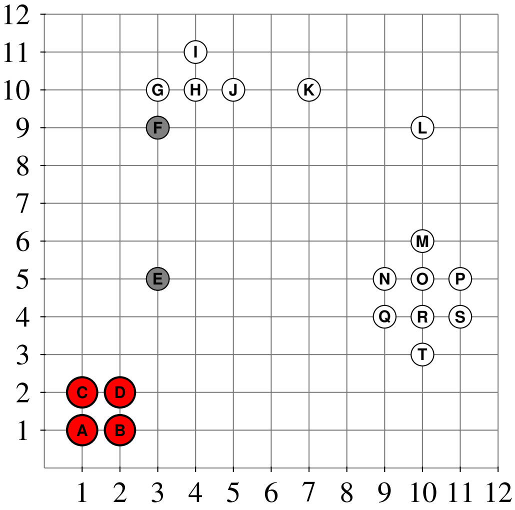
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example
Skip example



$\varepsilon = 1.1$

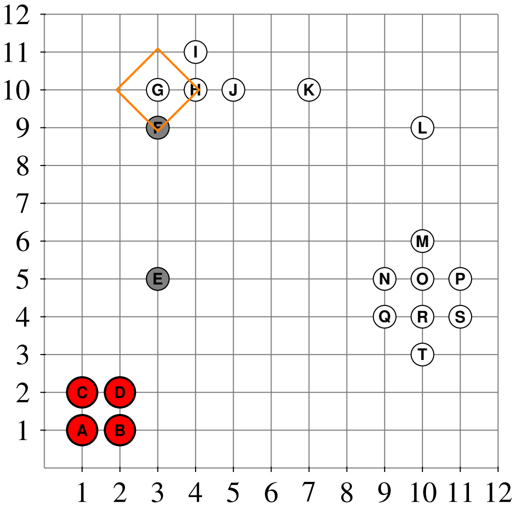
$minPts = 3$

Seed list:

Example 2

Start example

Skip example



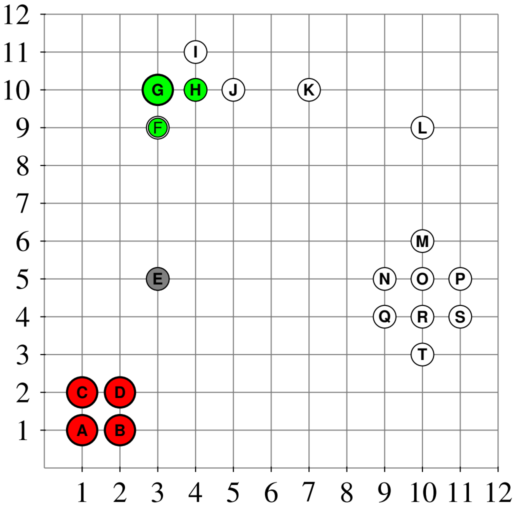
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example | Skip example



$\varepsilon = 1.1$

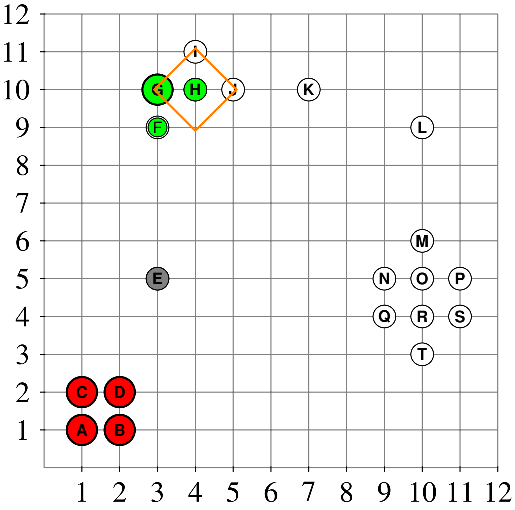
$minPts = 3$

Seed list:
H

Example 2

Start example

Skip example



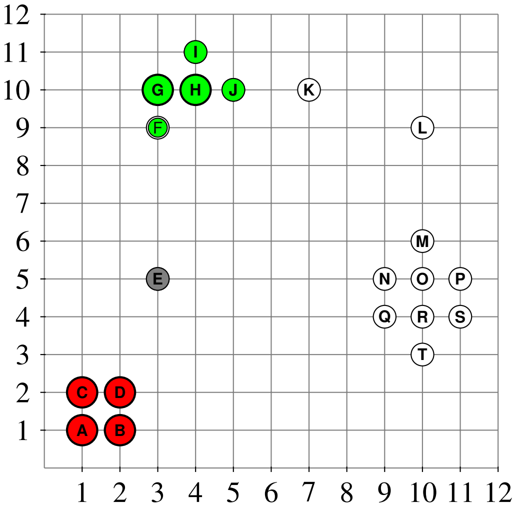
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example · Skip example



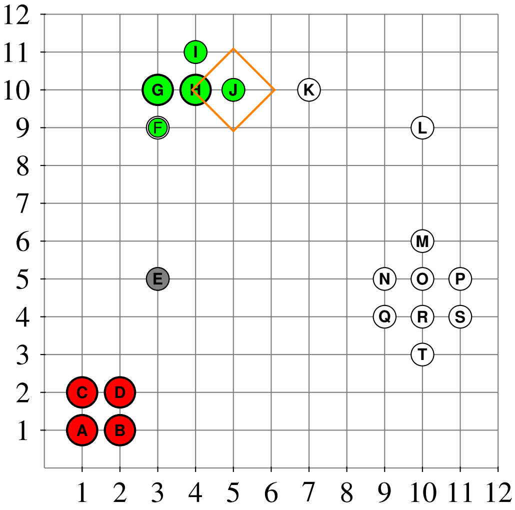
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
IJ

Example 2

Start example | Skip example



$\varepsilon = 1.1$

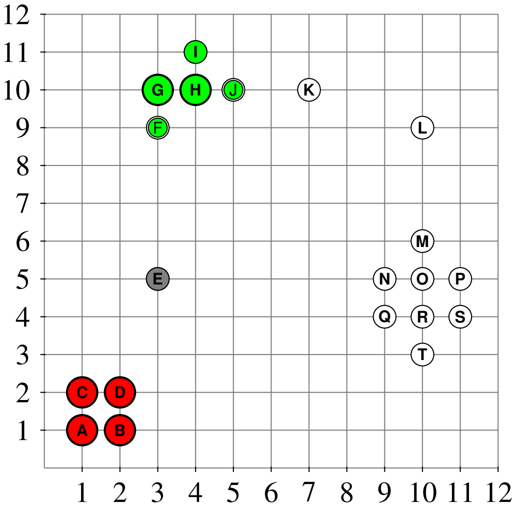
$minPts = 3$

Seed list:

I

Example 2

Start example
Skip example



$\varepsilon = 1.1$

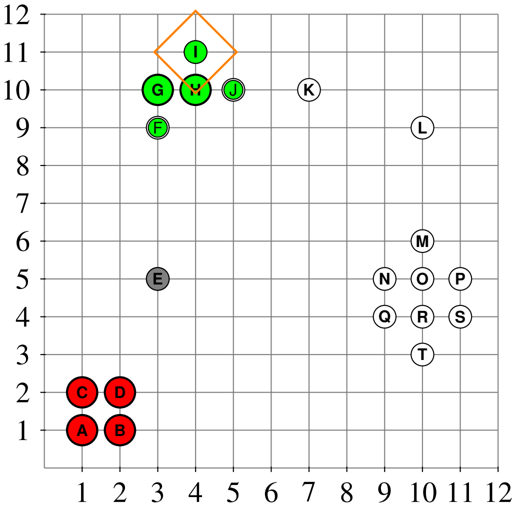
$minPts = 3$

Seed list:

I

Example 2

Start example
Skip example



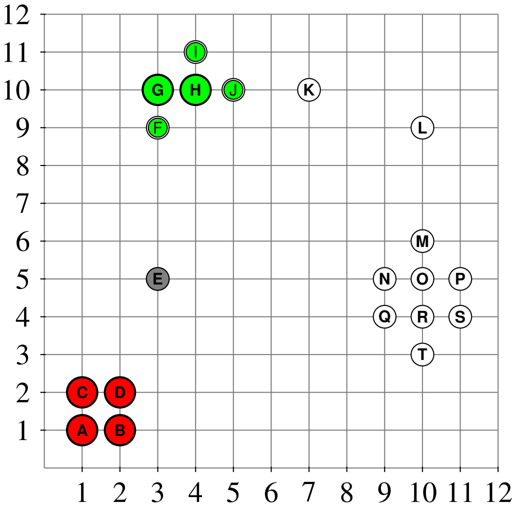
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example
Skip example



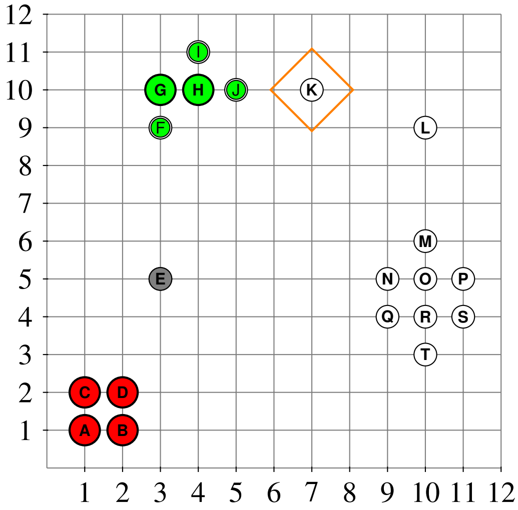
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example Skip example



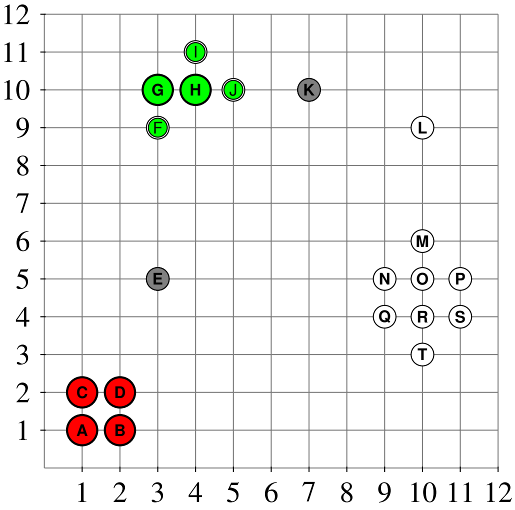
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example   Skip example



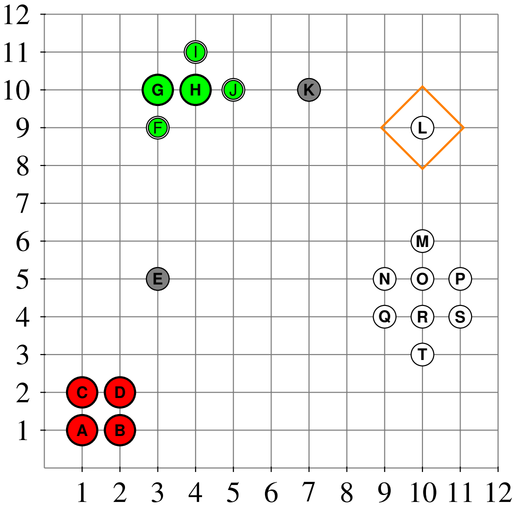
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example
Skip example



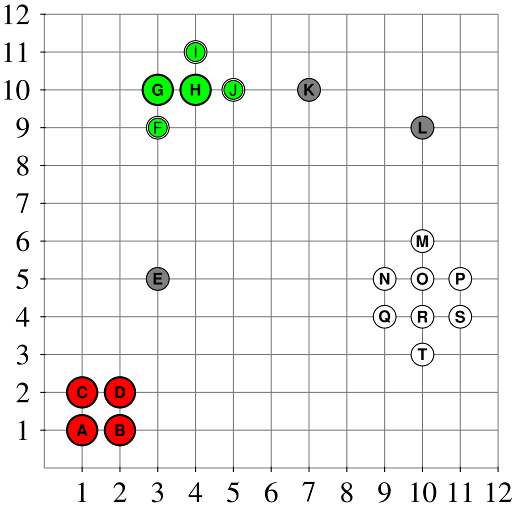
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example
Skip example



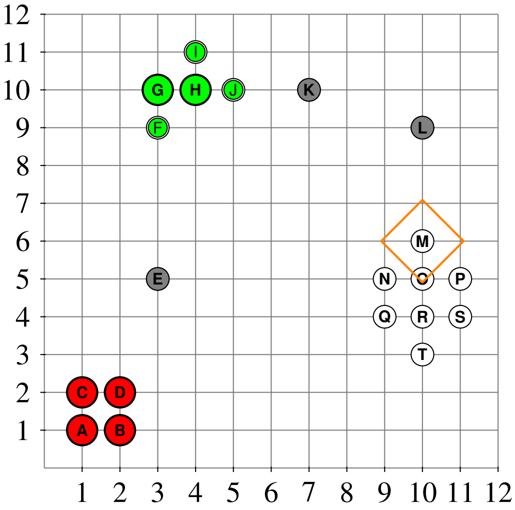
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example
Skip example



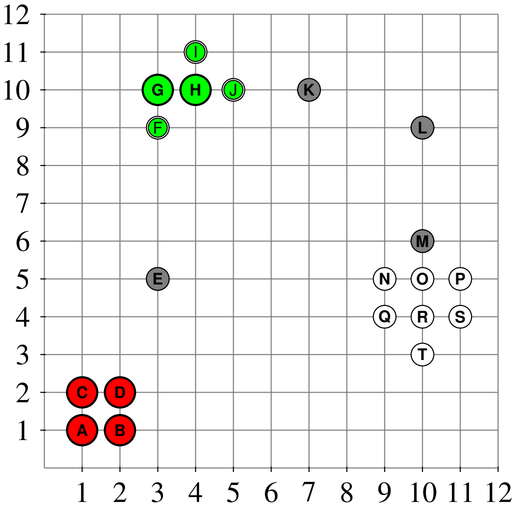
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example
Skip example



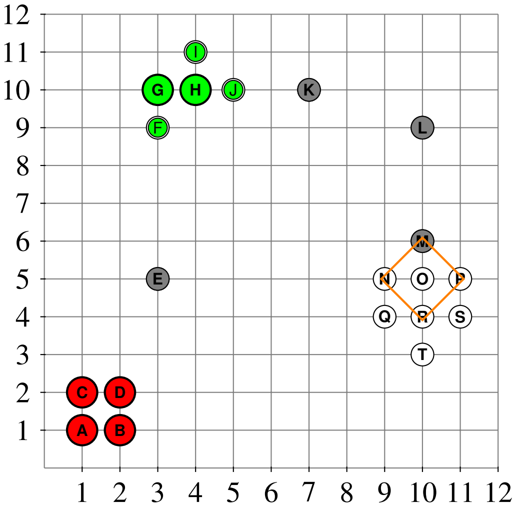
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example Skip example



$\varepsilon = 1.1$

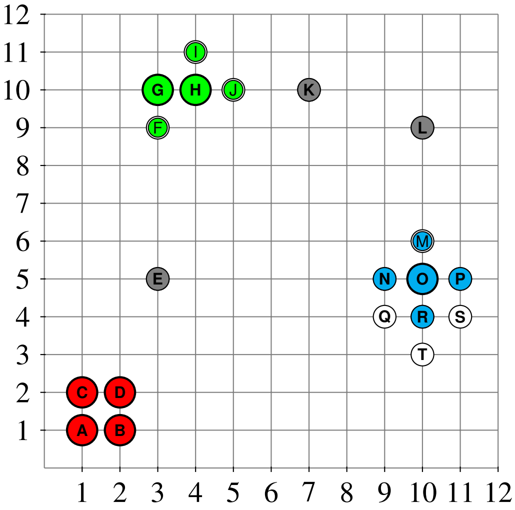
$minPts = 3$

Seed list:

Example 2

Start example

Skip example



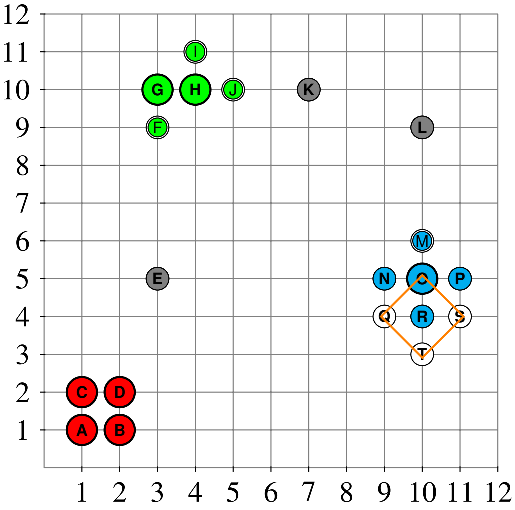
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
NPR

Example 2

Start example
Skip example



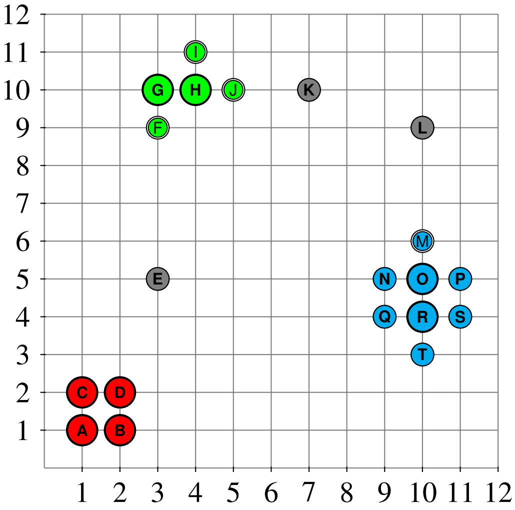
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
NP

Example 2

Start example
Skip example



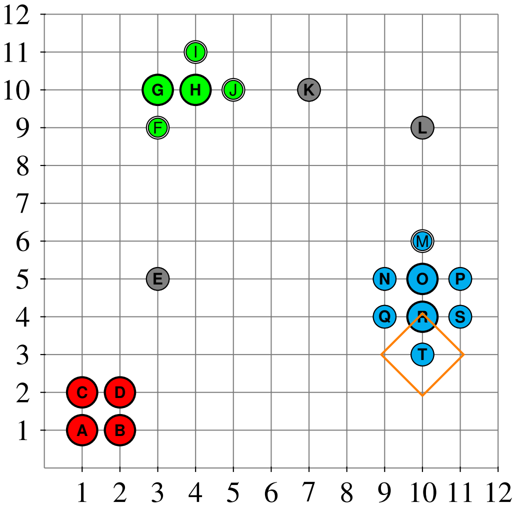
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
NPQST

Example 2

Start example | Skip example



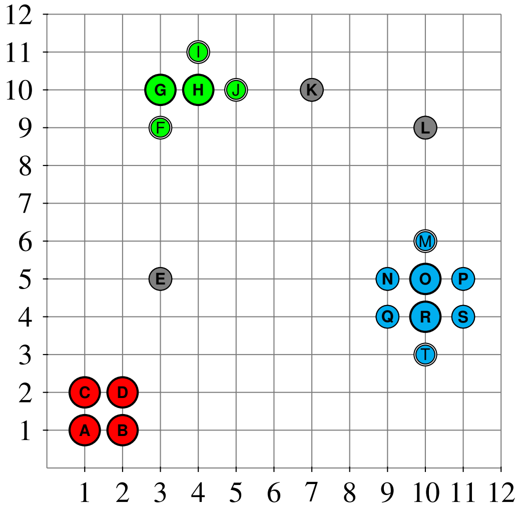
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
NPQS

Example 2

▸ Start example ▸▸ Skip example



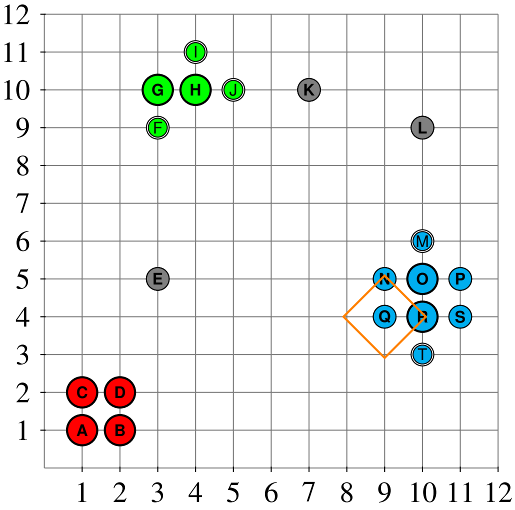
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
NPQS

Example 2

Start example Skip example



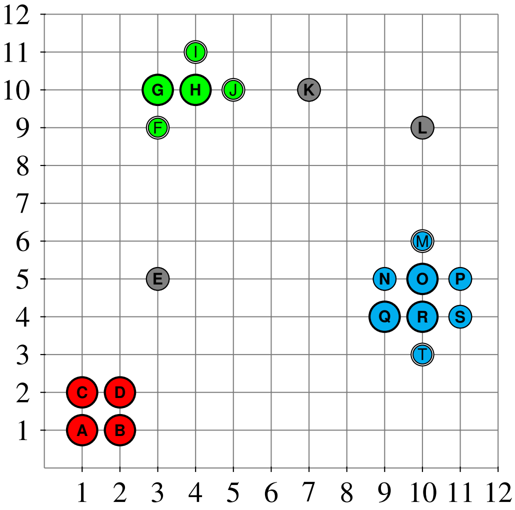
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
NPS

Example 2

Start example
Skip example



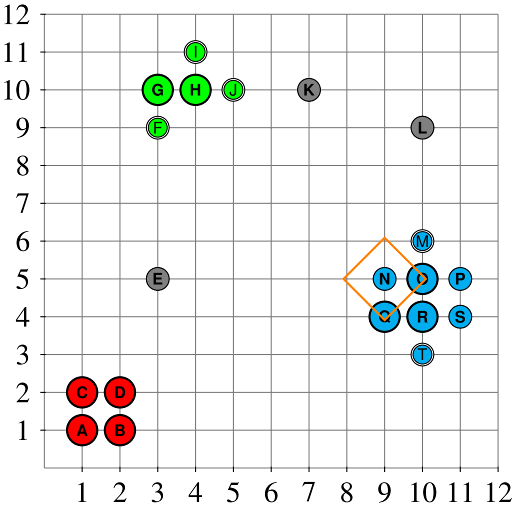
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
NPS

Example 2

▸ Start example ▸▸ Skip example



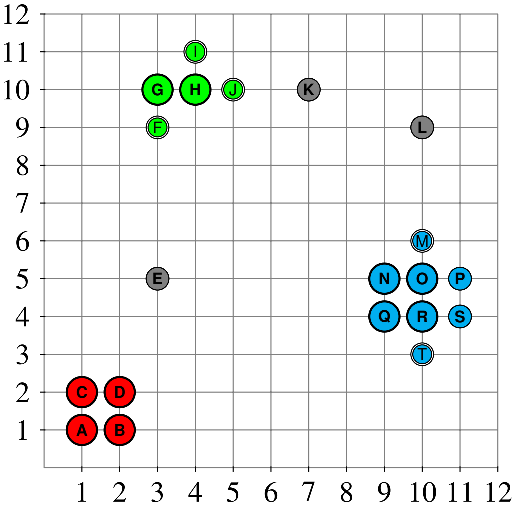
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
PS

Example 2

Start example Skip example



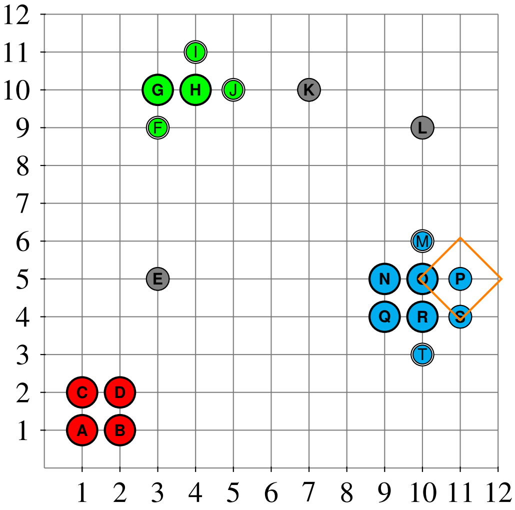
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
PS

Example 2

Start example Skip example



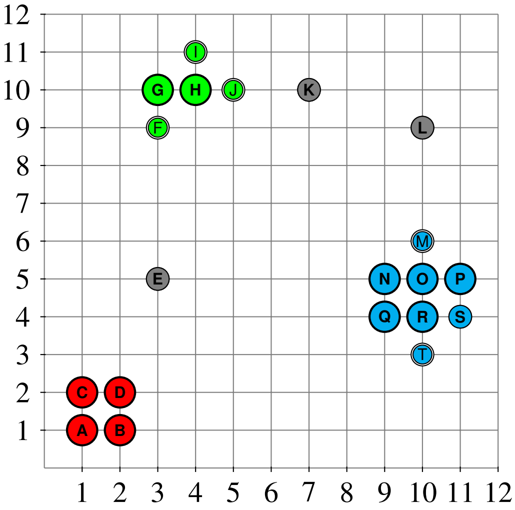
$\varepsilon = 1.1$

$minPts = 3$

Seed list:
S

Example 2

Start example
Skip example

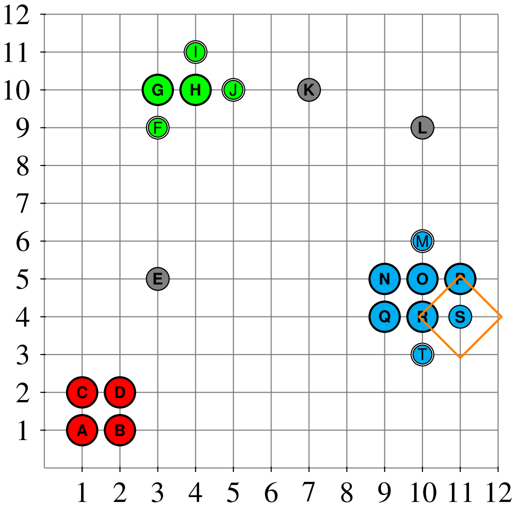


$\varepsilon = 1.1$

$minPts = 3$

Seed list:
S

Example 2 ▸ Start example ⏩ Skip example



$\varepsilon = 1.1$

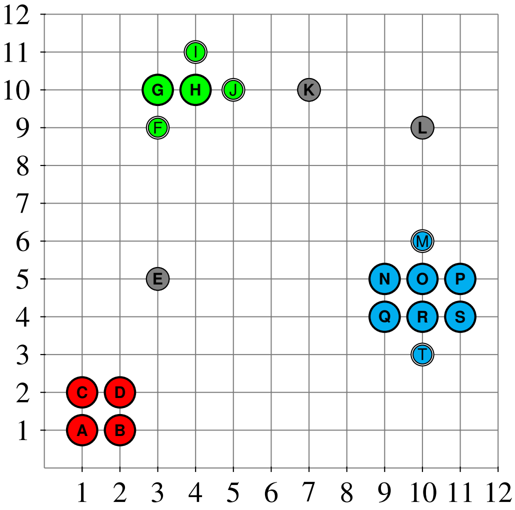
$minPts = 3$

Seed list:

Example 2

Start example

Skip example



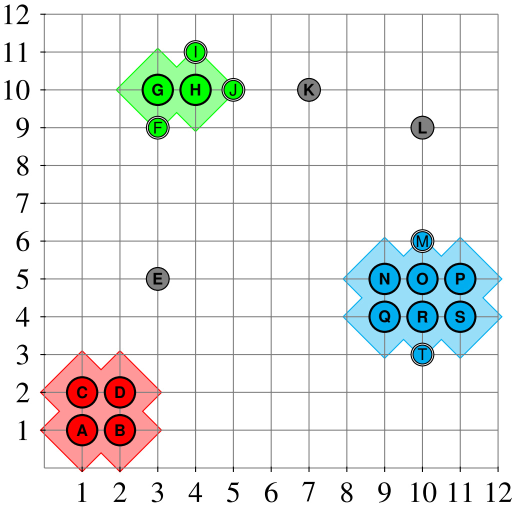
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 2

Start example   Skip example



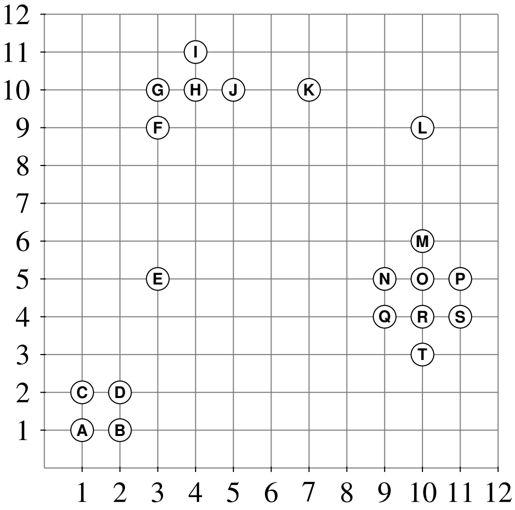
$\varepsilon = 1.1$

$minPts = 3$

Seed list:

Example 3

Start example | Skip example

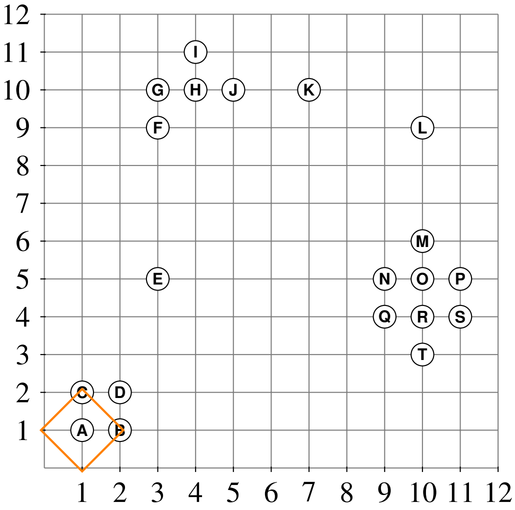


$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3



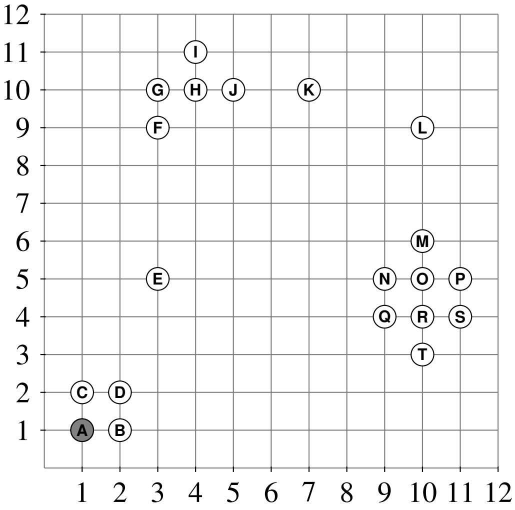
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example
Skip example



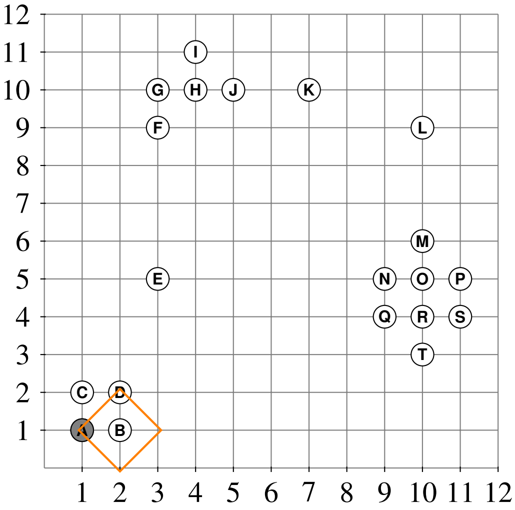
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example | Skip example



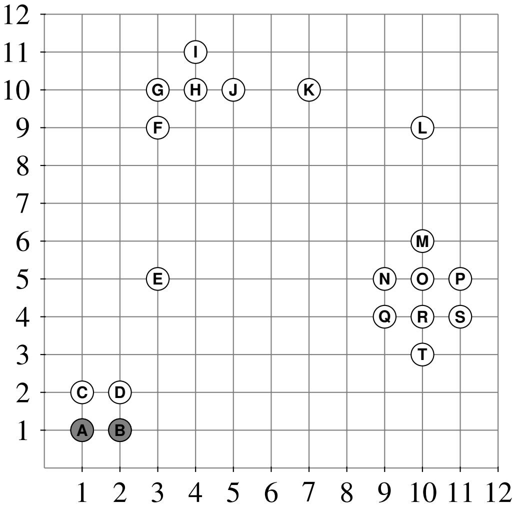
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example · Skip example



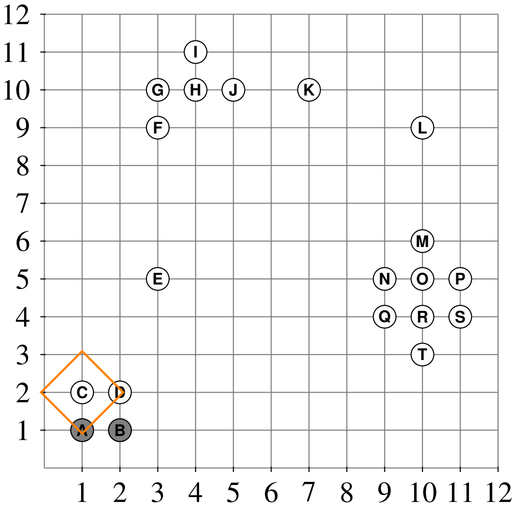
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example
Skip example



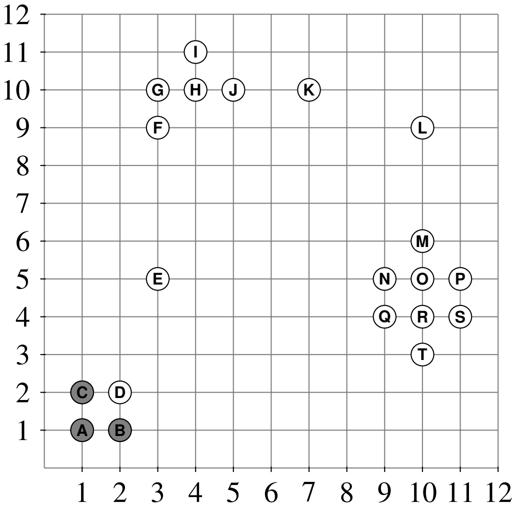
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example | Skip example



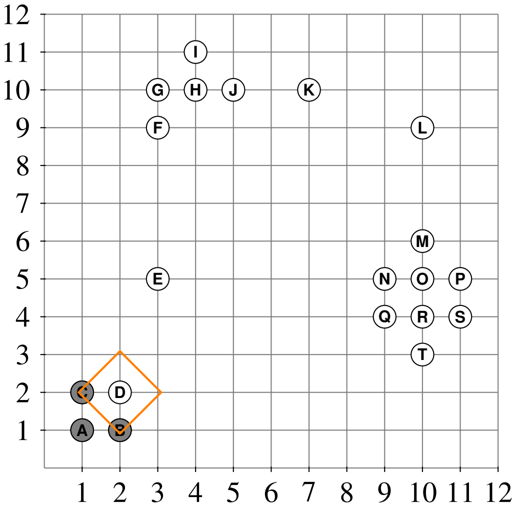
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

▸ Start example ⏩ Skip example



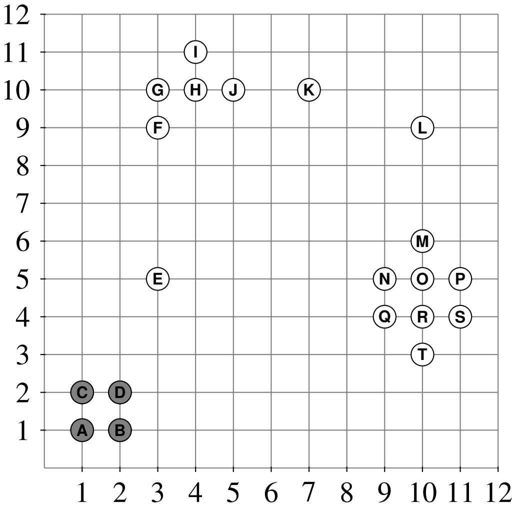
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example   Skip example



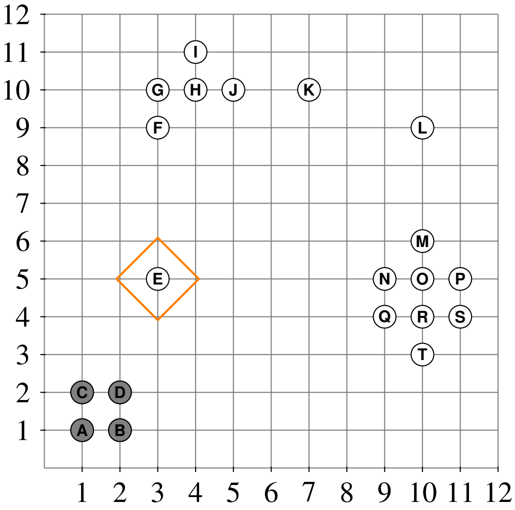
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example   Skip example



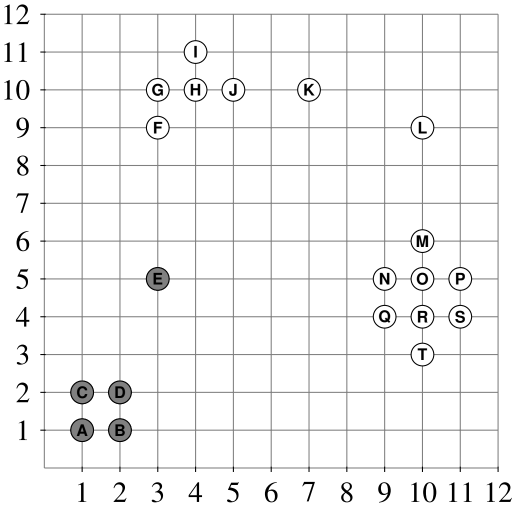
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example | Skip example



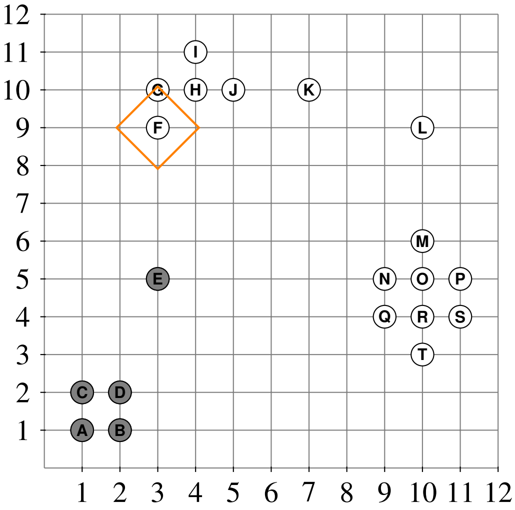
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

▸ Start example ▸▸ Skip example



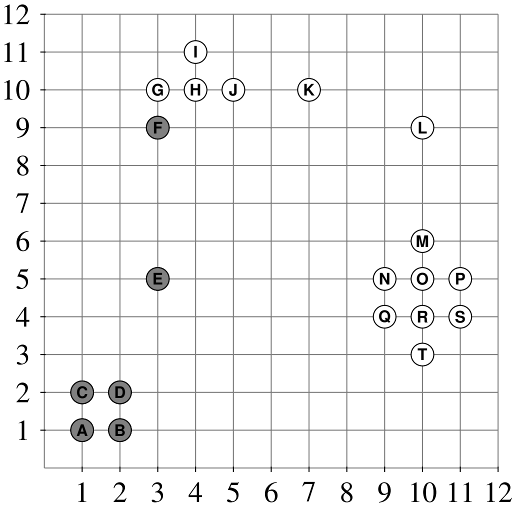
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example
Skip example



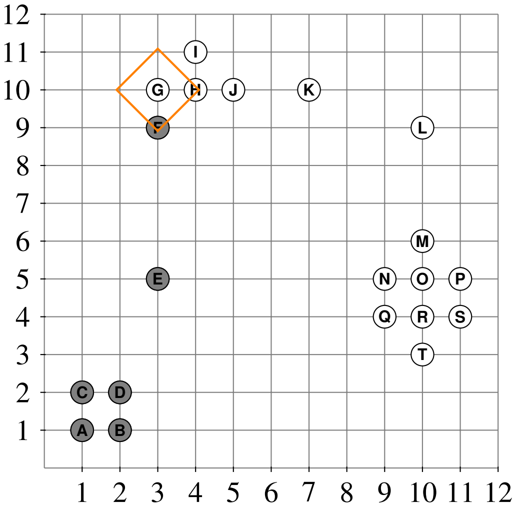
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example   Skip example



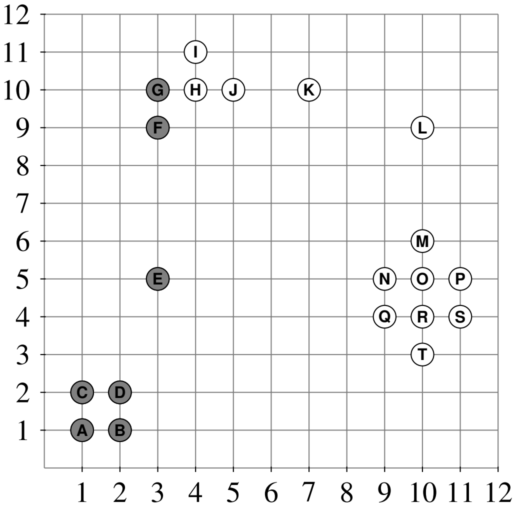
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example Skip example



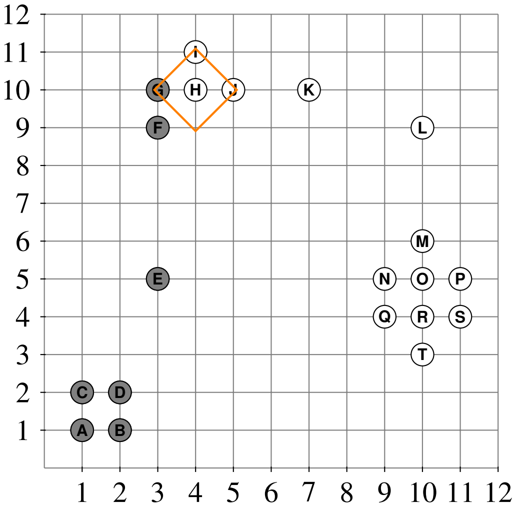
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example · Skip example



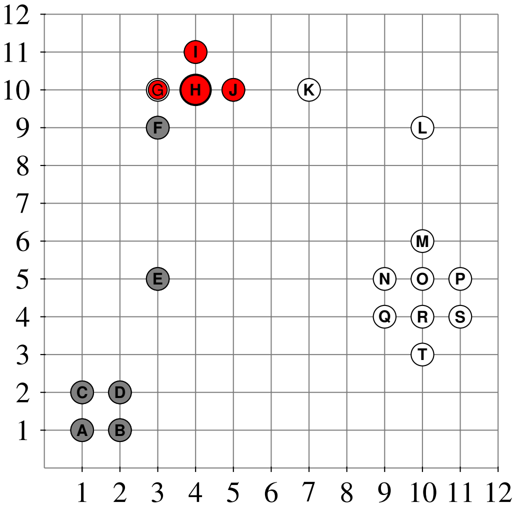
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example
Skip example



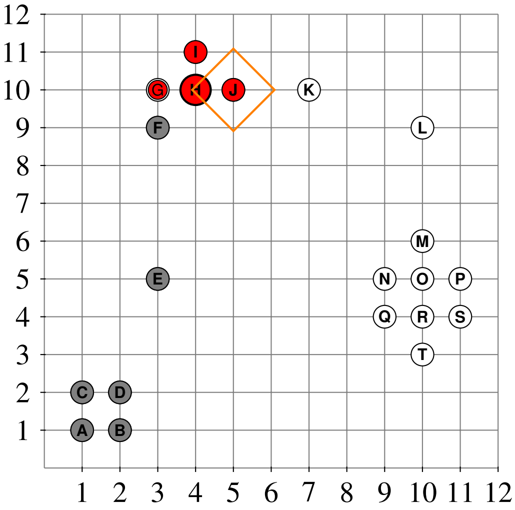
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
IJ

Example 3

Start example
Skip example



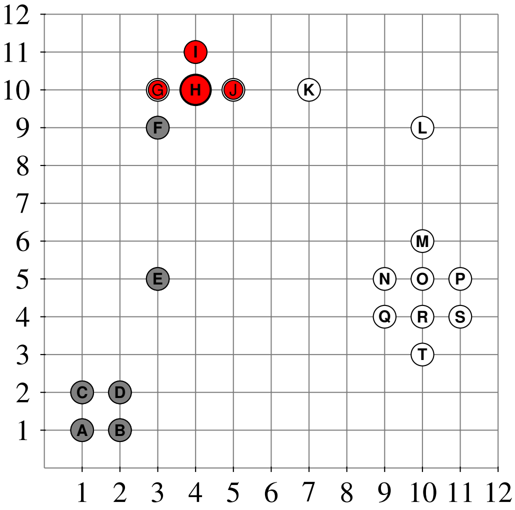
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
I

Example 3

Start example
Skip example



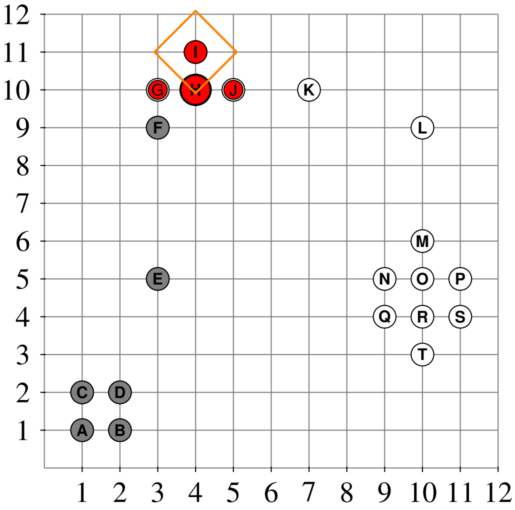
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
I

Example 3

Start example
Skip example



$\varepsilon = 1.1$

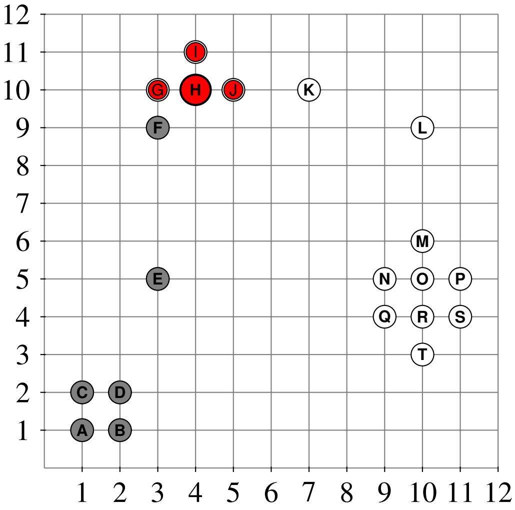
$minPts = 4$

Seed list:

Example 3

Start example

Skip example



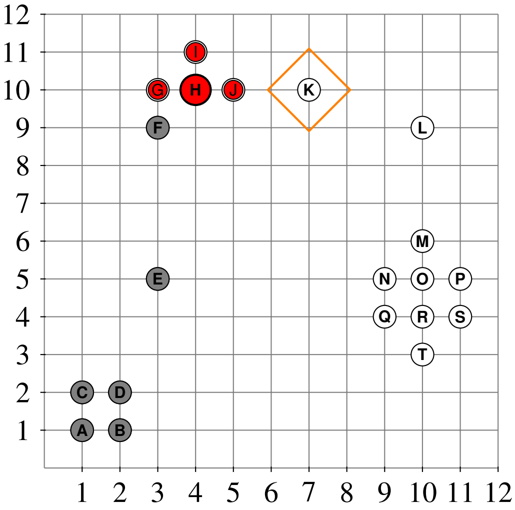
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example
Skip example



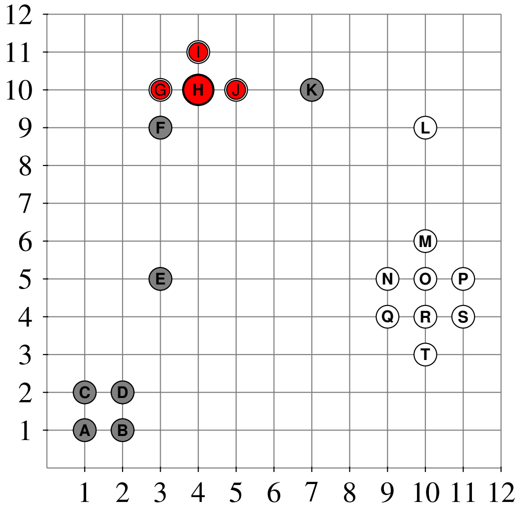
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example | Skip example



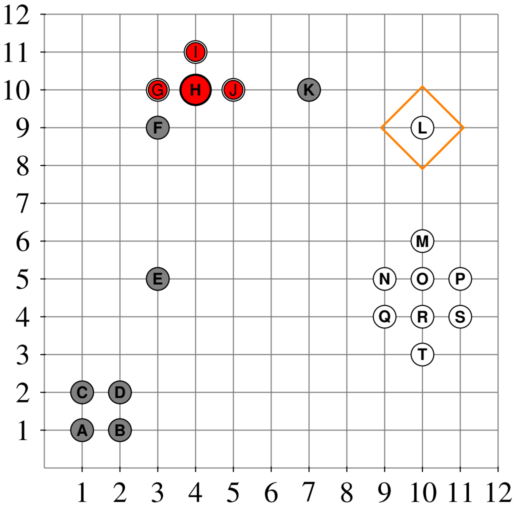
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example
Skip example



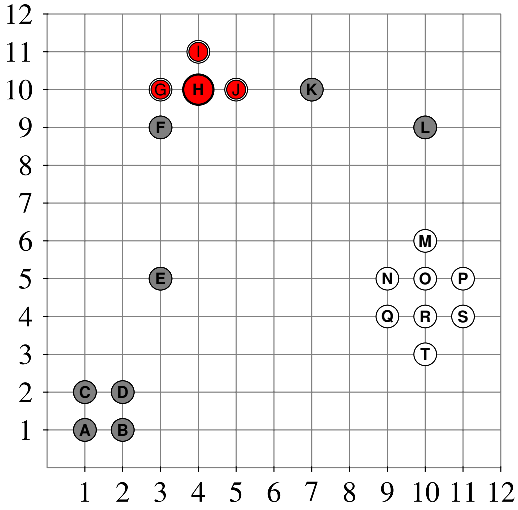
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example | Skip example



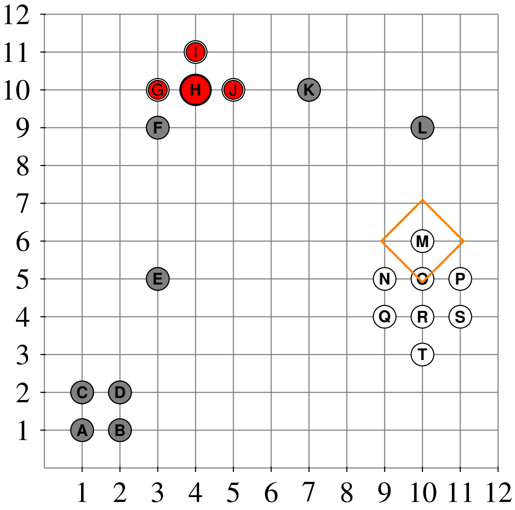
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example
Skip example



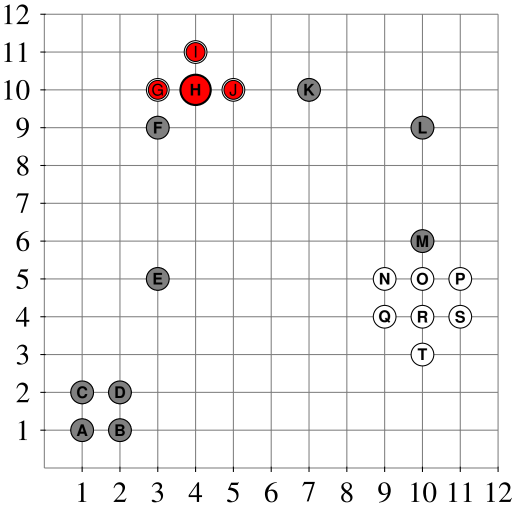
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example
Skip example



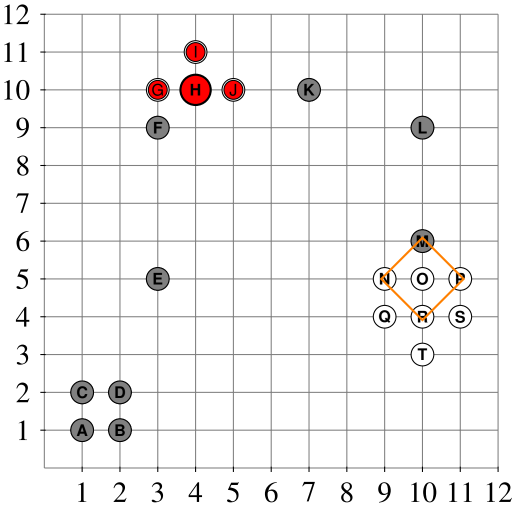
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example | Skip example



$\varepsilon = 1.1$

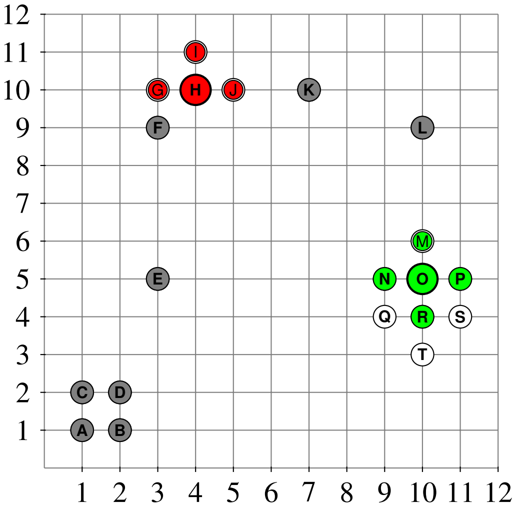
$minPts = 4$

Seed list:

Example 3

Start example

Skip example



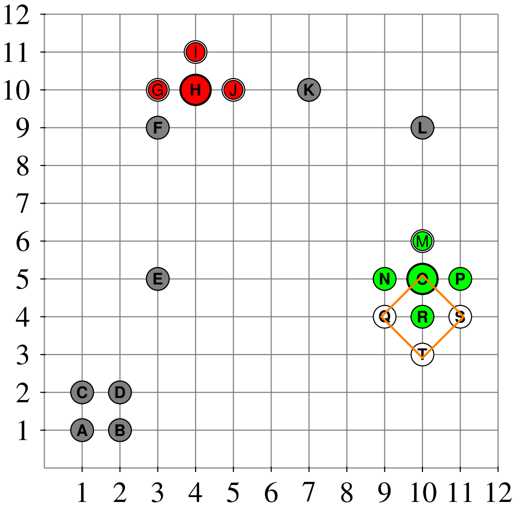
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
NPR

Example 3

Start example · Skip example



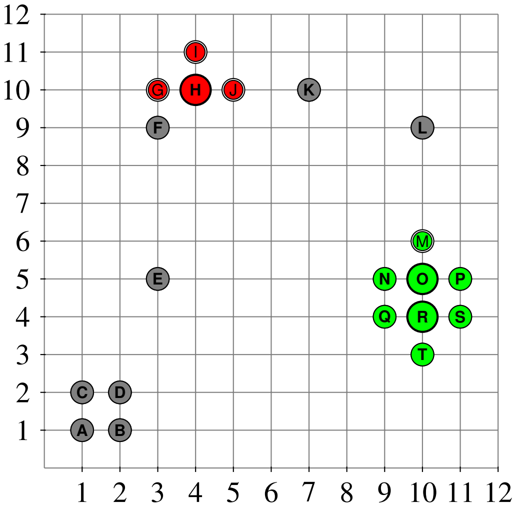
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
NP

Example 3

Start example   Skip example



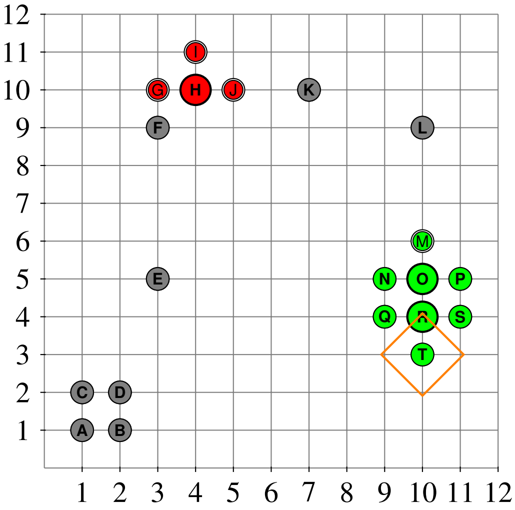
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
NPQST

Example 3

Start example Skip example

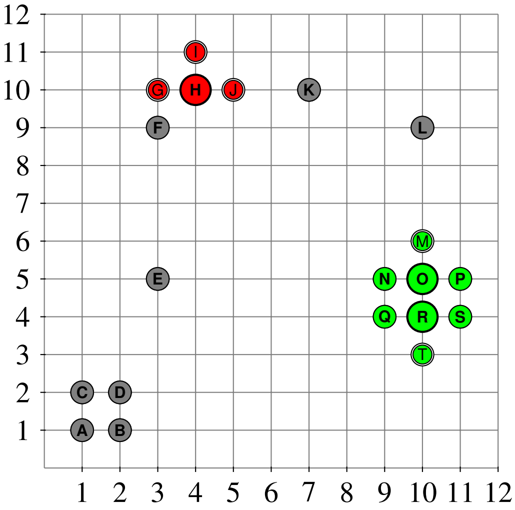


$\varepsilon = 1.1$

$minPts = 4$

Seed list:
NPQS

Example 3 ▸ Start example ⏩ Skip example



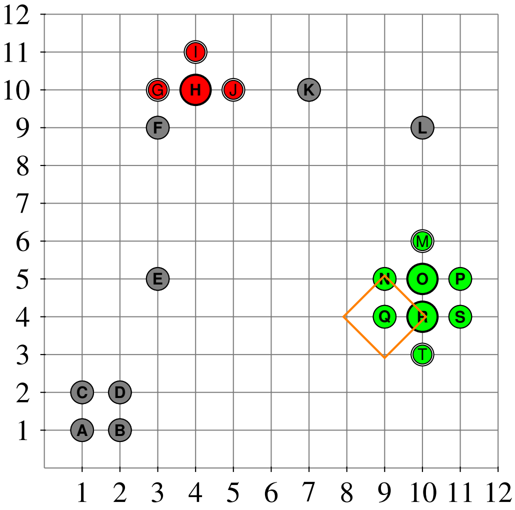
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
NPQS

Example 3

Start example · Skip example



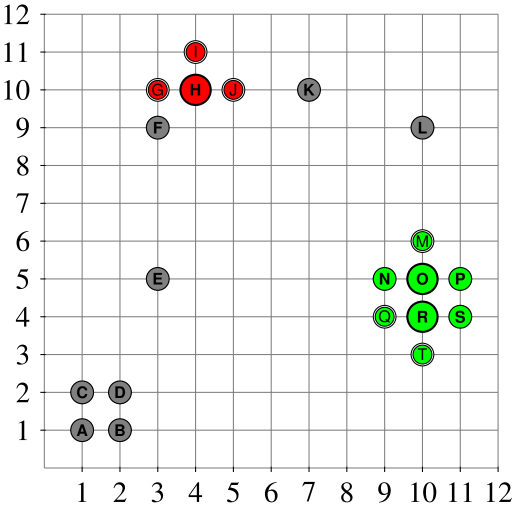
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
NPS

Example 3

Start example   Skip example



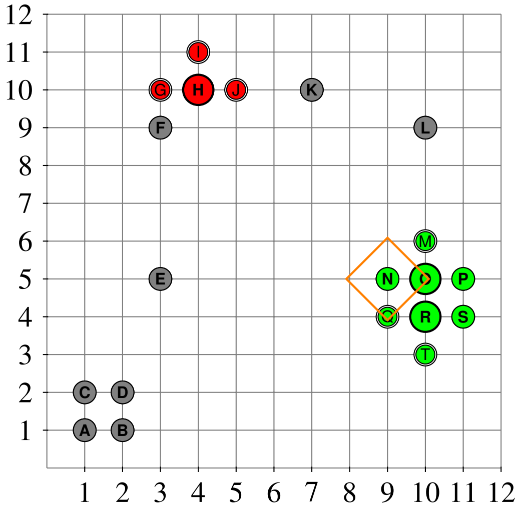
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
NPS

Example 3

Start example Skip example

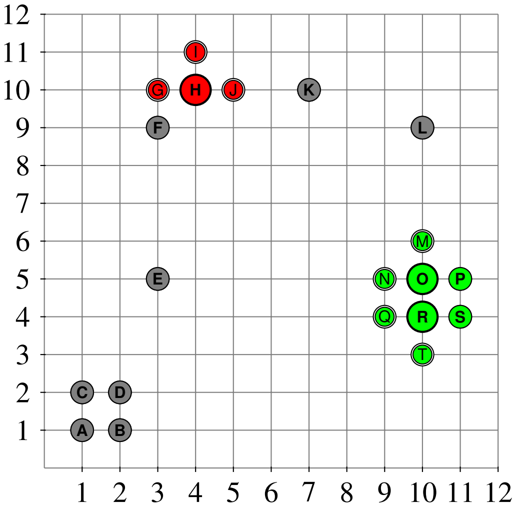


$\varepsilon = 1.1$

$minPts = 4$

Seed list:
PS

Example 3



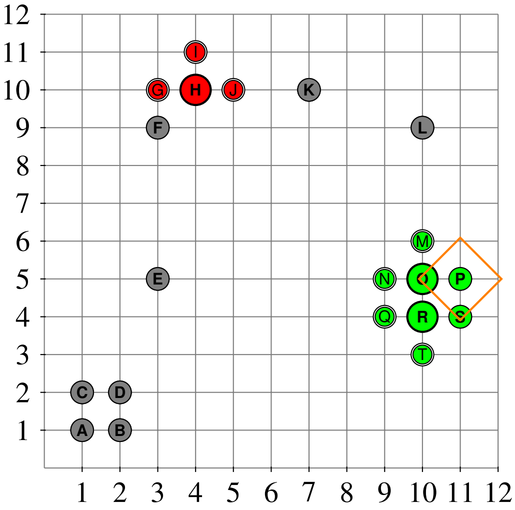
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
PS

Example 3

Start example · Skip example



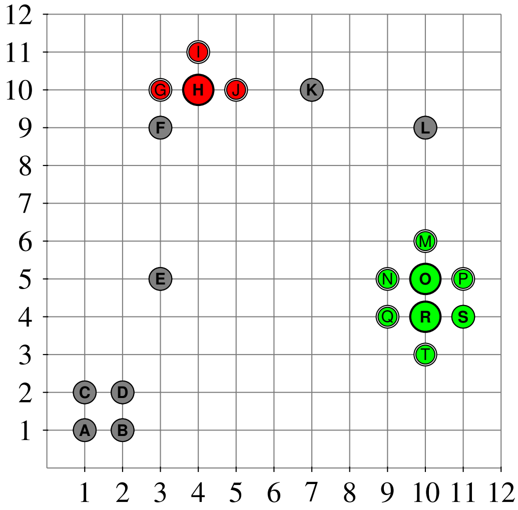
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
S

Example 3

Start example   Skip example



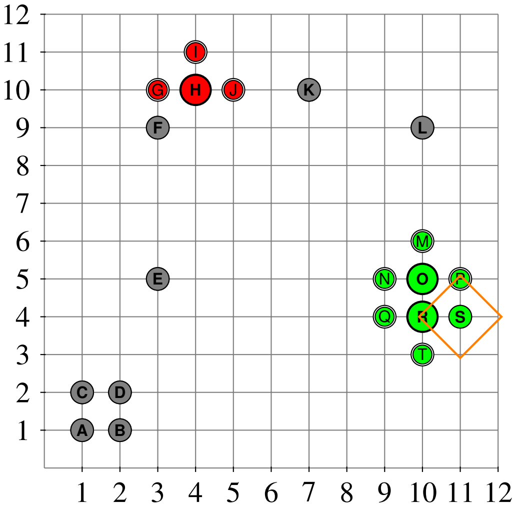
$\varepsilon = 1.1$

$minPts = 4$

Seed list:
S

Example 3

▸ Start example ▸▸ Skip example



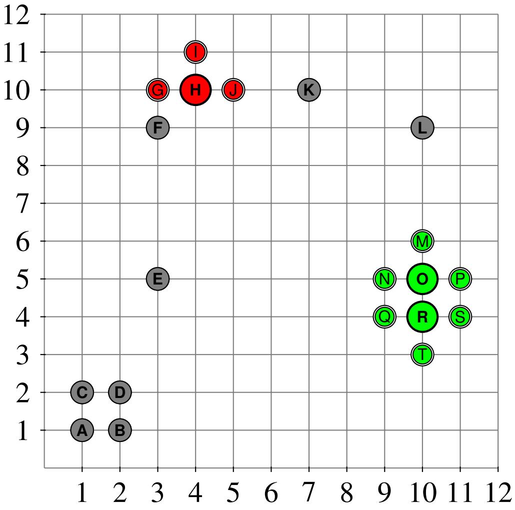
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example Skip example



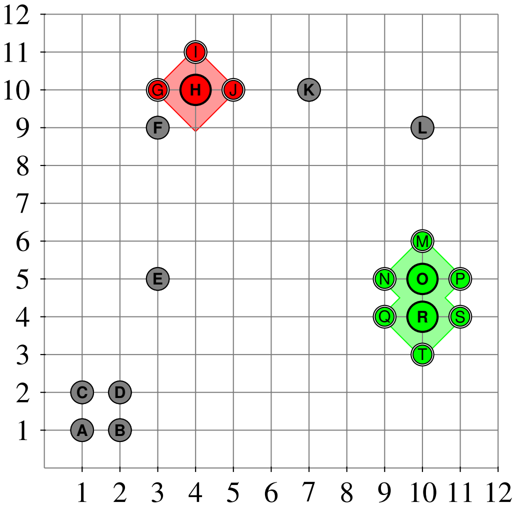
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 3

Start example | Skip example



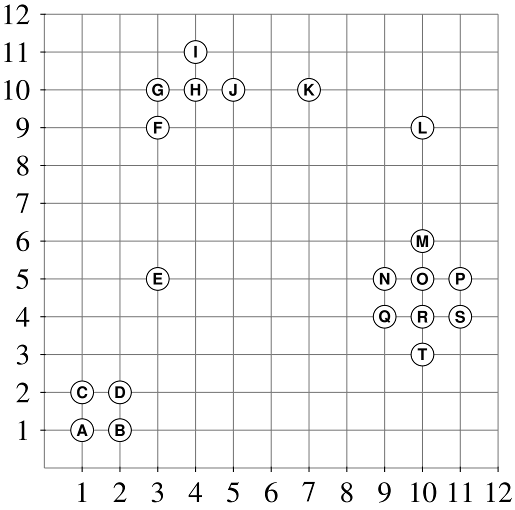
$\varepsilon = 1.1$

$minPts = 4$

Seed list:

Example 4

Start example | Skip example



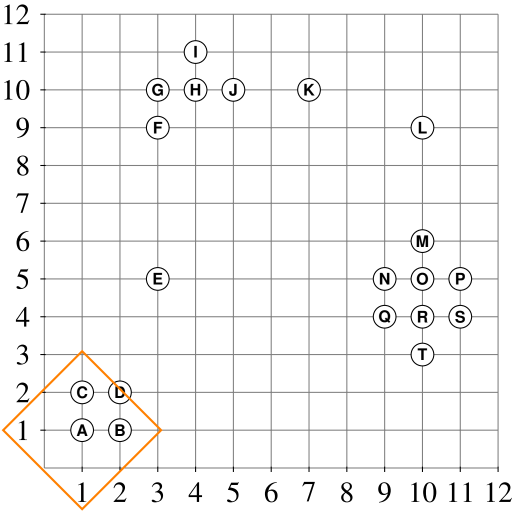
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example | Skip example



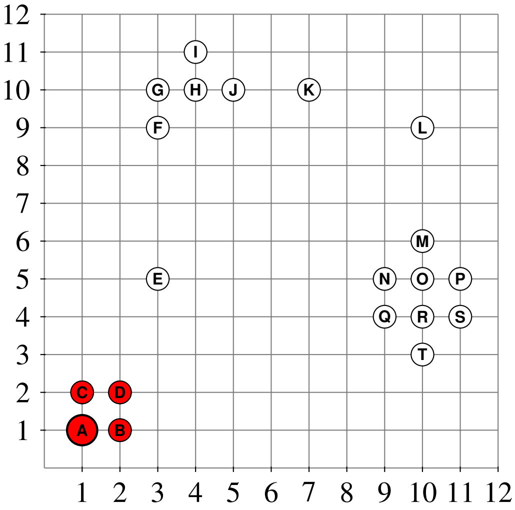
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example
Skip example



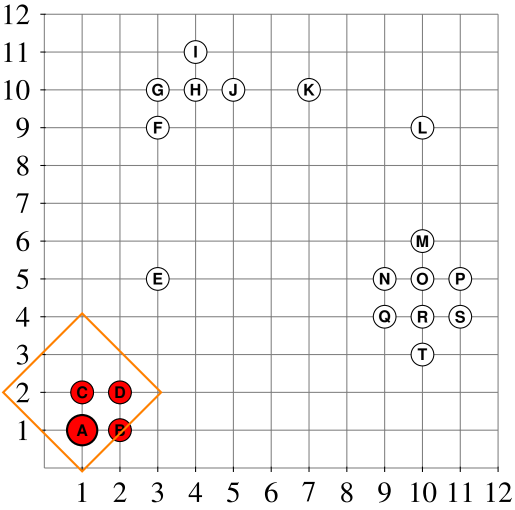
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
BCD

Example 4

Start example
Skip example



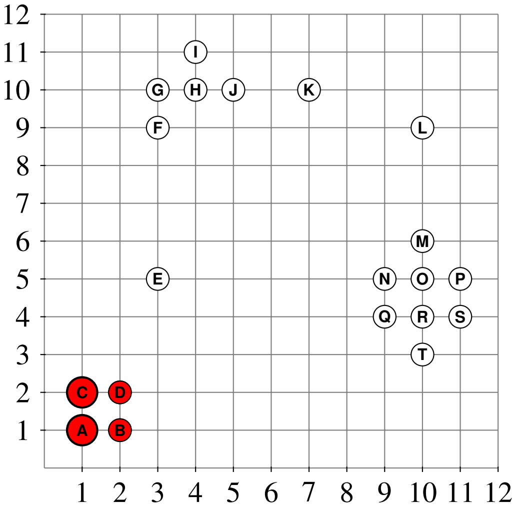
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
BD

Example 4

Start example   Skip example



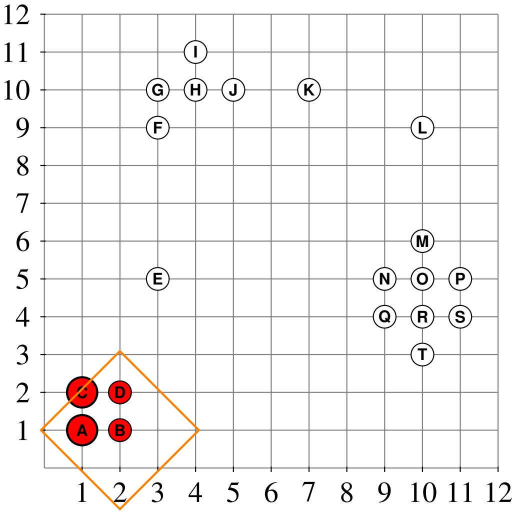
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
BD

Example 4

Start example · Skip example



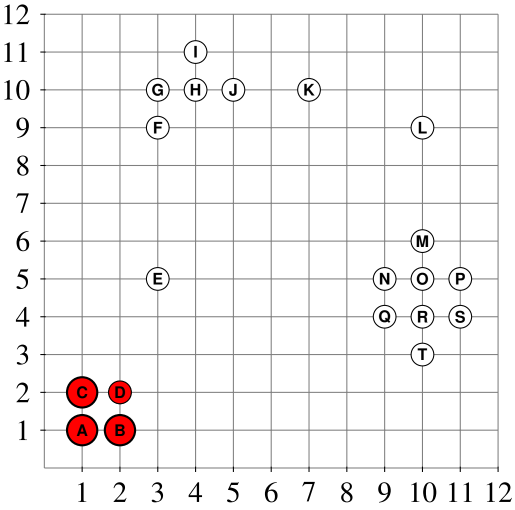
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
D

Example 4

Start example | Skip example



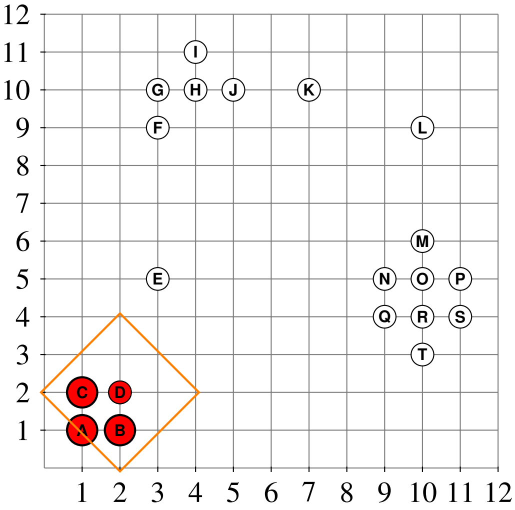
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
D

Example 4

▸ Start example ▸▸ Skip example



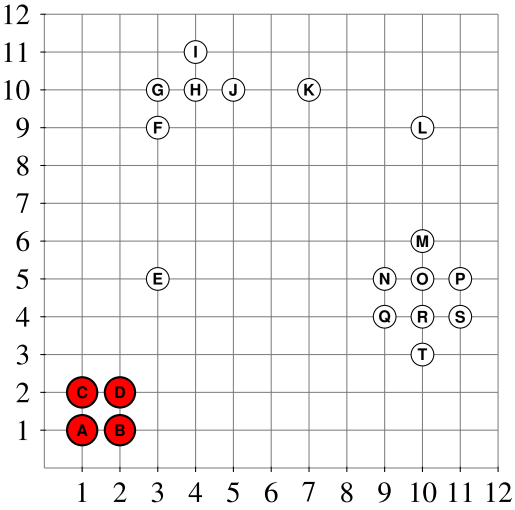
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example Skip example



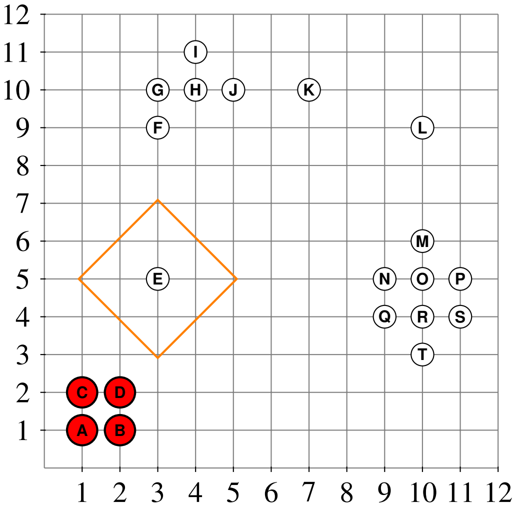
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example
Skip example



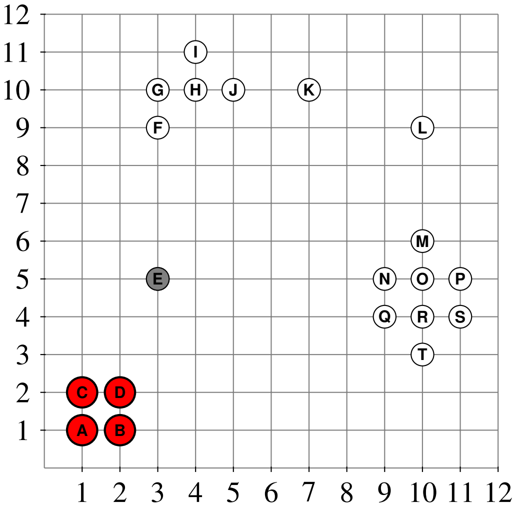
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example Skip example



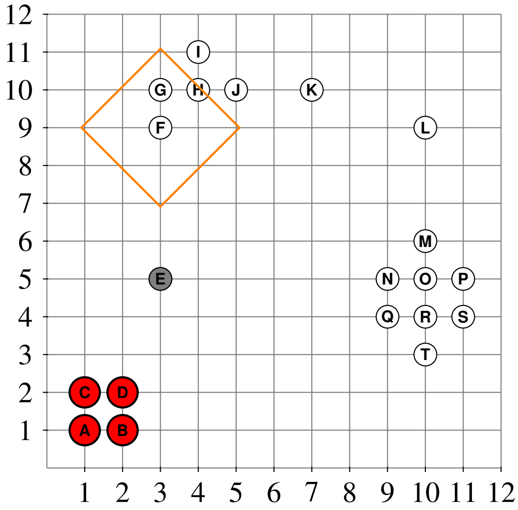
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example | Skip example



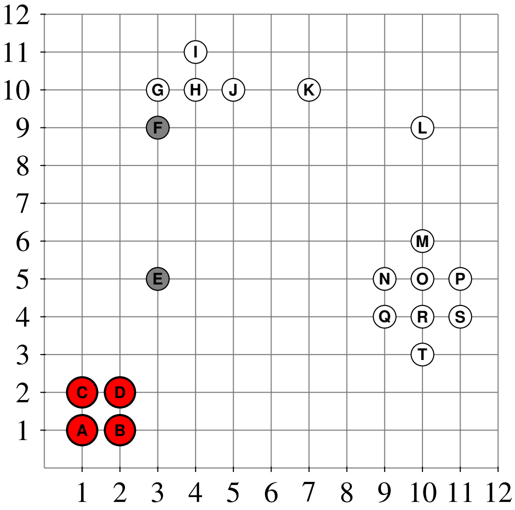
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example | Skip example



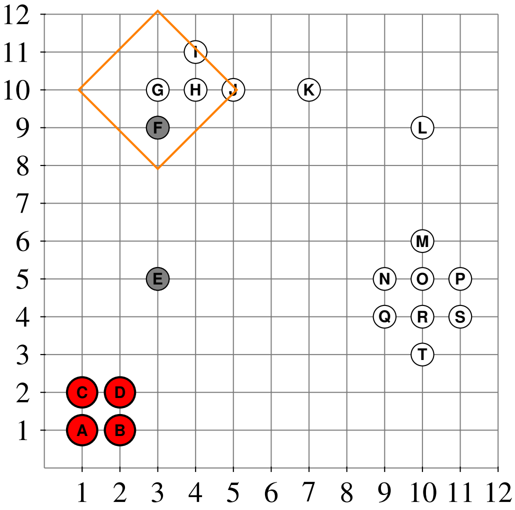
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example Skip example



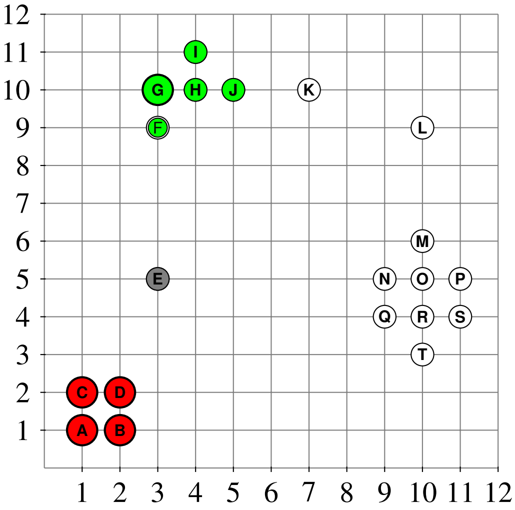
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example
Skip example



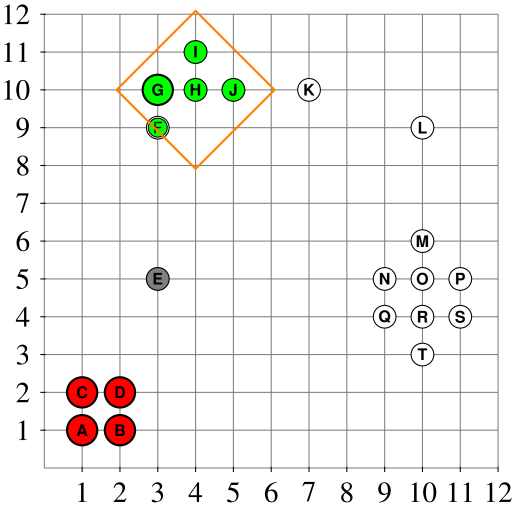
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
HIJ

Example 4

Start example Skip example



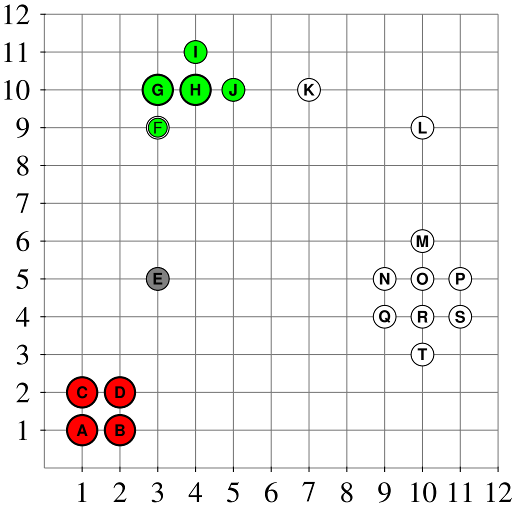
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
IJ

Example 4

Start example | Skip example



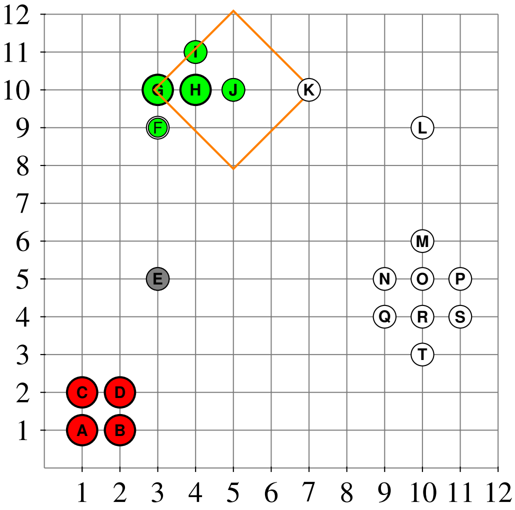
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
IJ

Example 4

Start example Skip example



$\varepsilon = 2.1$

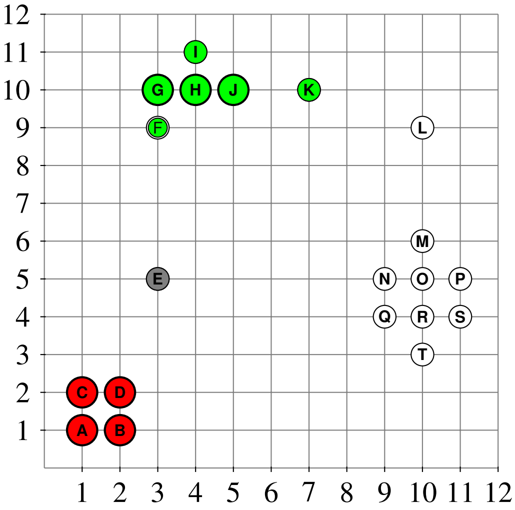
$minPts = 4$

Seed list:

I

Example 4

Start example | Skip example



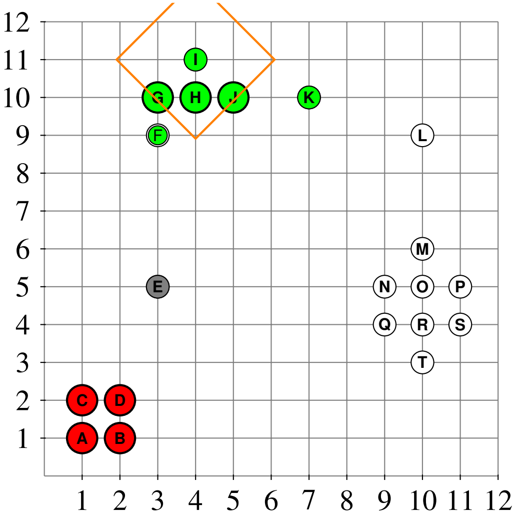
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
IK

Example 4

Start example Skip example



$\varepsilon = 2.1$

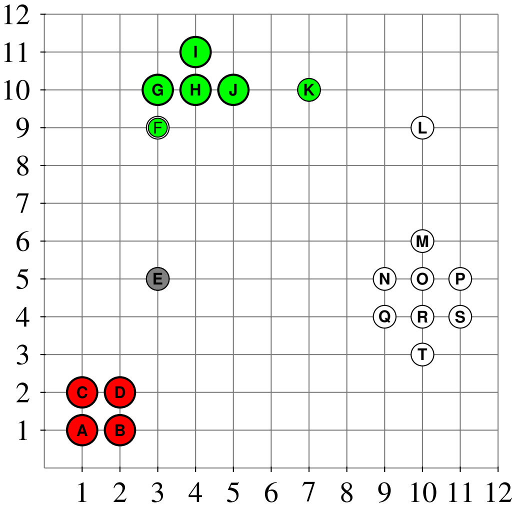
$minPts = 4$

Seed list:
K

Example 4

Start example

Skip example



$\varepsilon = 2.1$

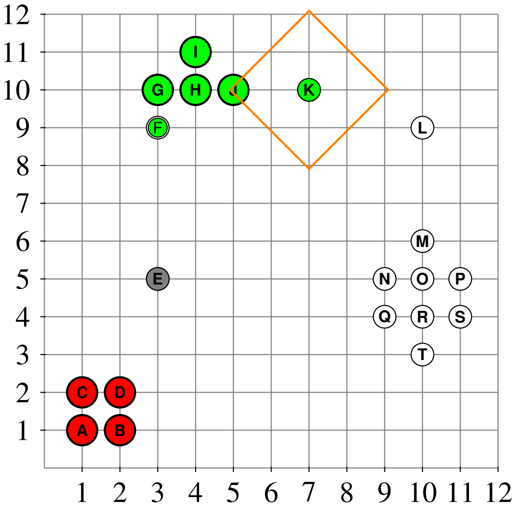
$minPts = 4$

Seed list:
K

SDU
SYDDANSK UNIVERSITET

Example 4

Start example | Skip example



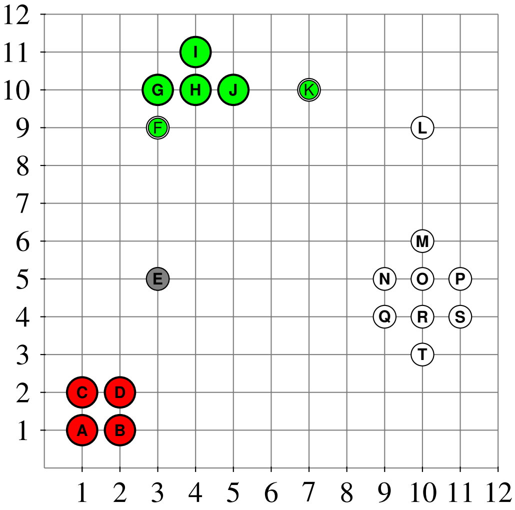
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example
Skip example



$\varepsilon = 2.1$

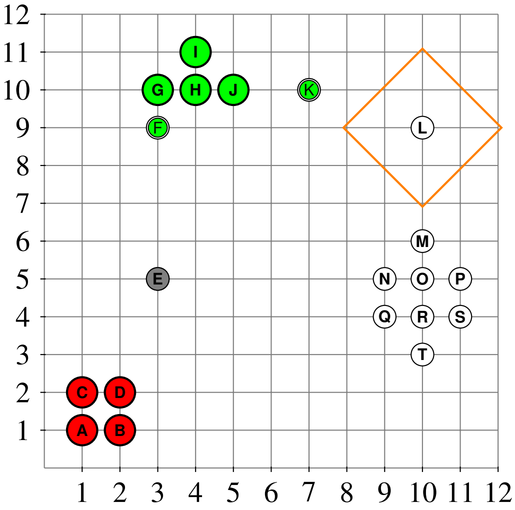
$minPts = 4$

Seed list:

Example 4

Start example

Skip example



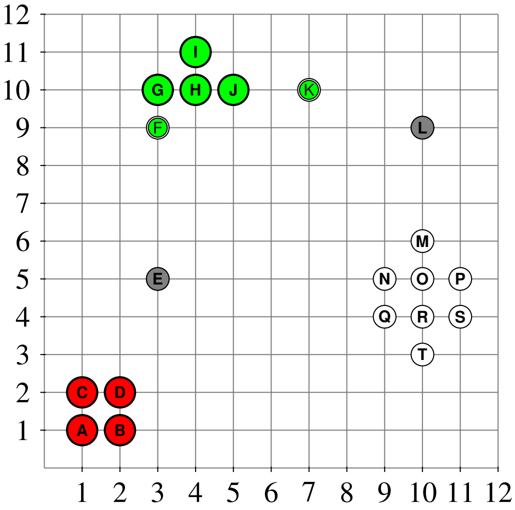
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example Skip example



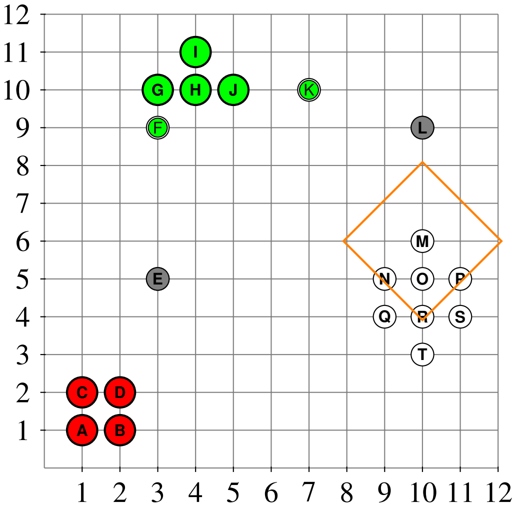
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example | Skip example



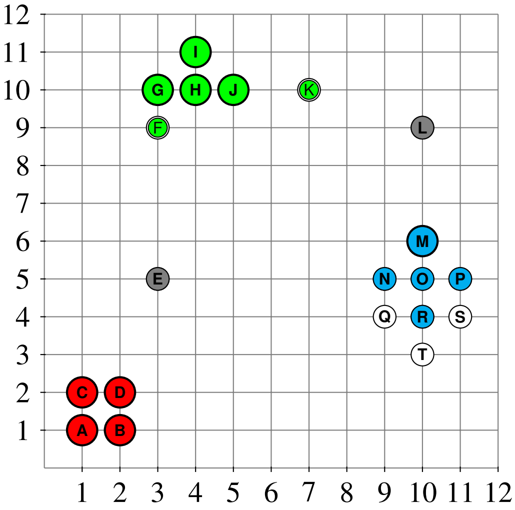
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example | Skip example



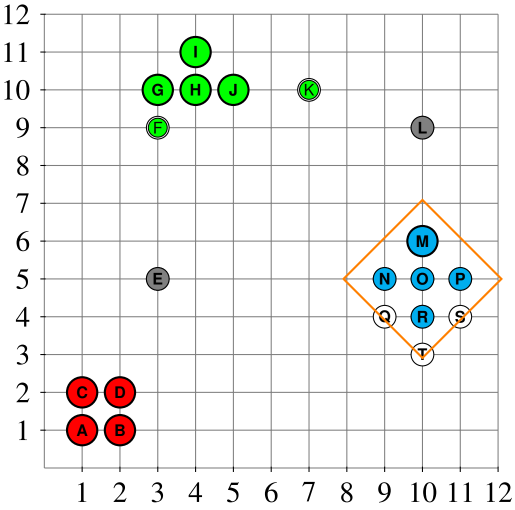
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
ONPR

Example 4

Start example
Skip example



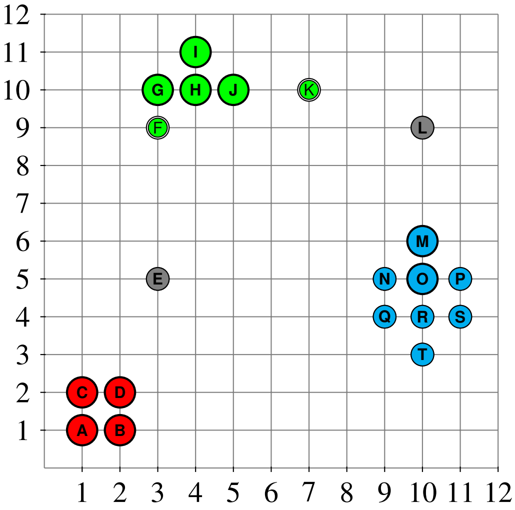
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
NPR

Example 4

Start example
Skip example



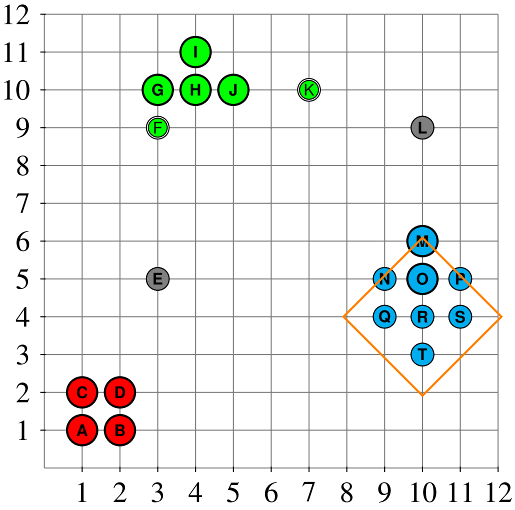
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
NPRQST

Example 4

Start example Skip example



$\varepsilon = 2.1$

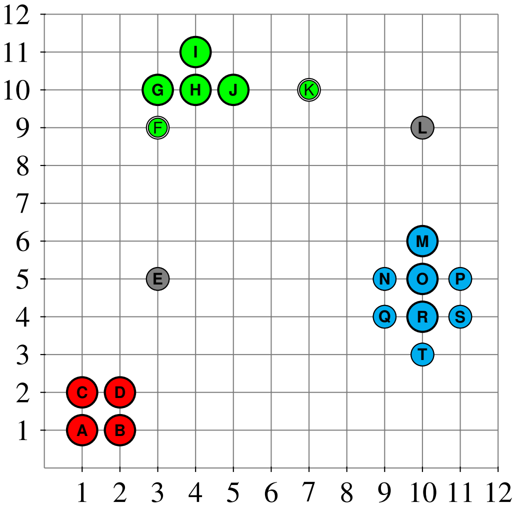
$minPts = 4$

Seed list:
NPQST

SDU
SYDDANSK UNIVERSITET

Example 4

Start example
Skip example



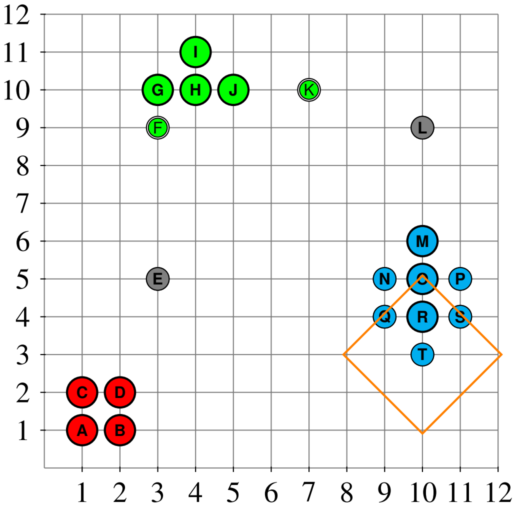
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
NPQST

Example 4

Start example
Skip example



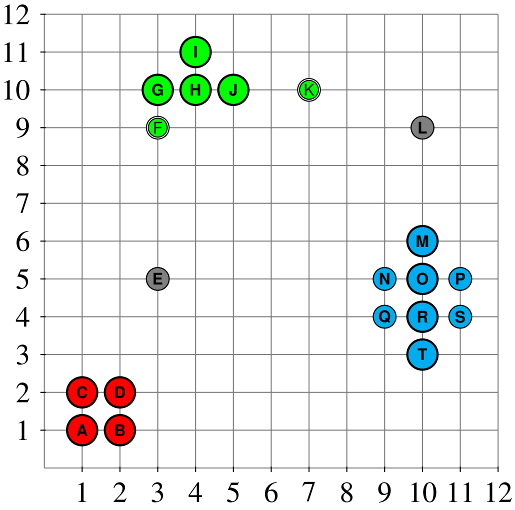
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
NPQS

Example 4

Start example Skip example



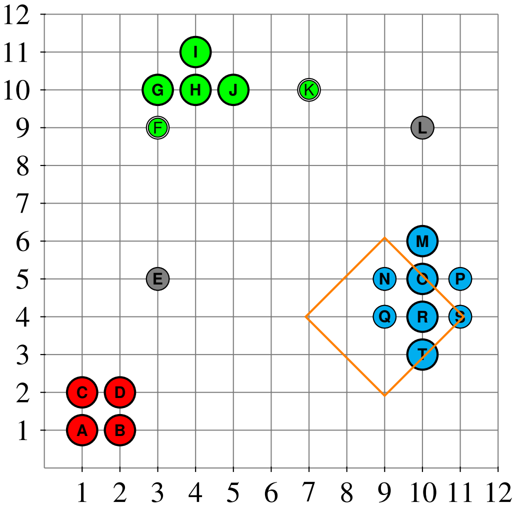
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
NPQS

Example 4

Start example
Skip example



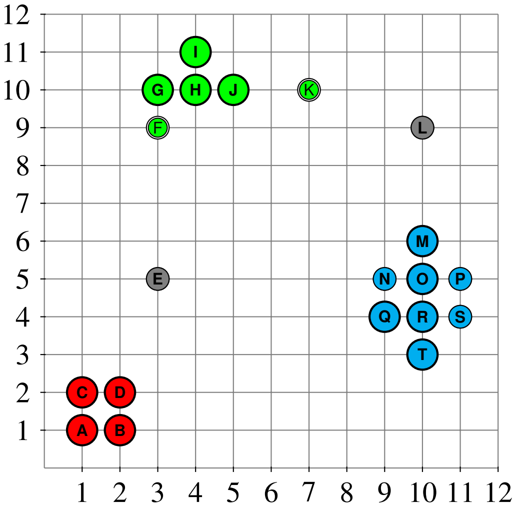
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
NPS

Example 4

Start example
Skip example



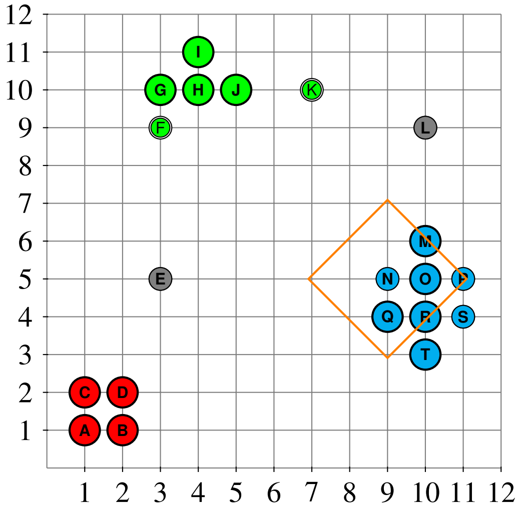
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
NPS

Example 4

Start example
Skip example



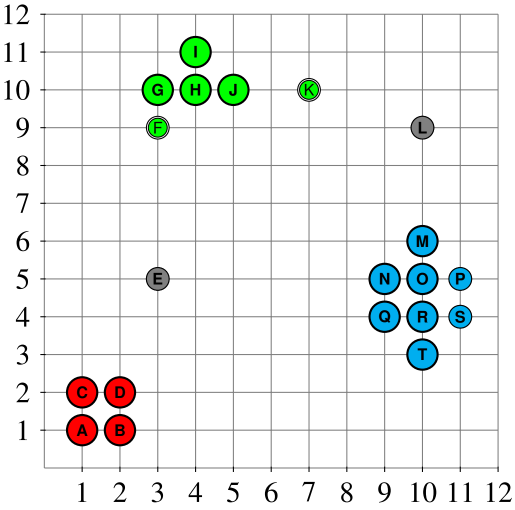
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
PS

Example 4

Start example
Skip example



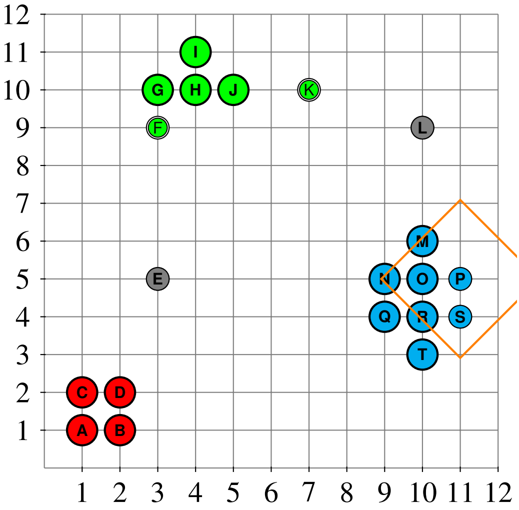
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
PS

Example 4

Start example | Skip example



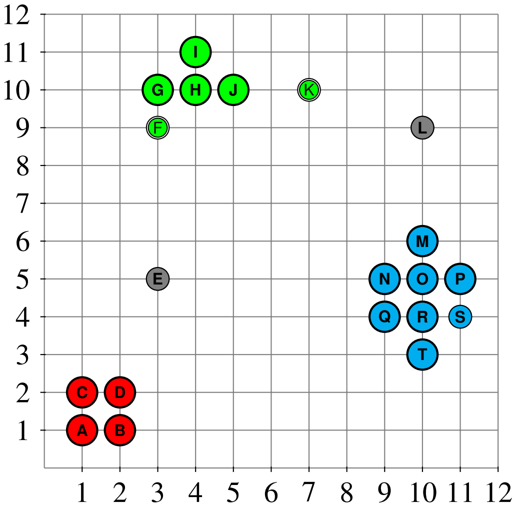
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
S

Example 4

Start example
Skip example



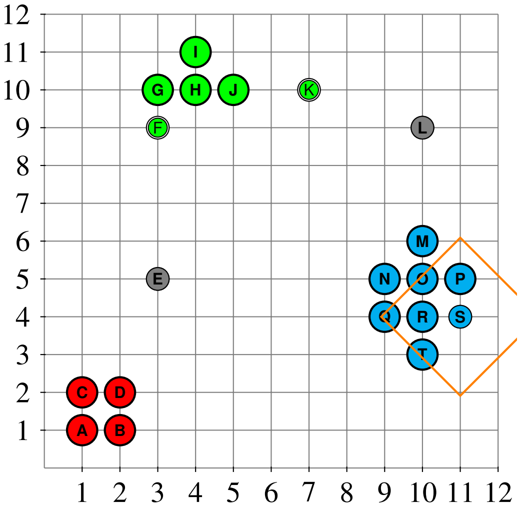
$\varepsilon = 2.1$

$minPts = 4$

Seed list:
S

Example 4

Start example
Skip example



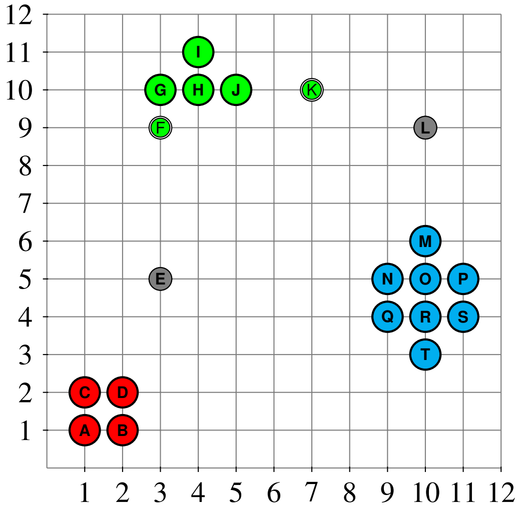
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example Skip example



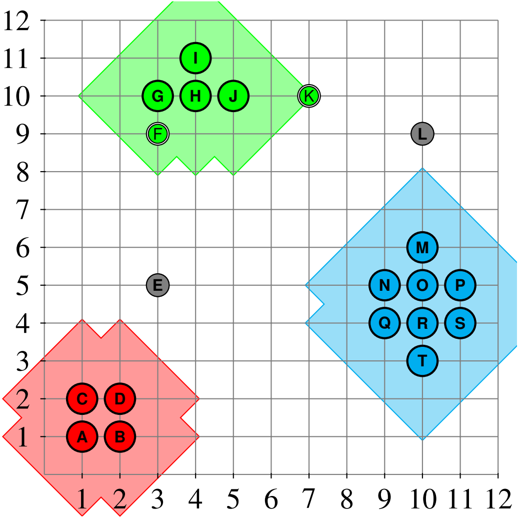
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 4

Start example Skip example



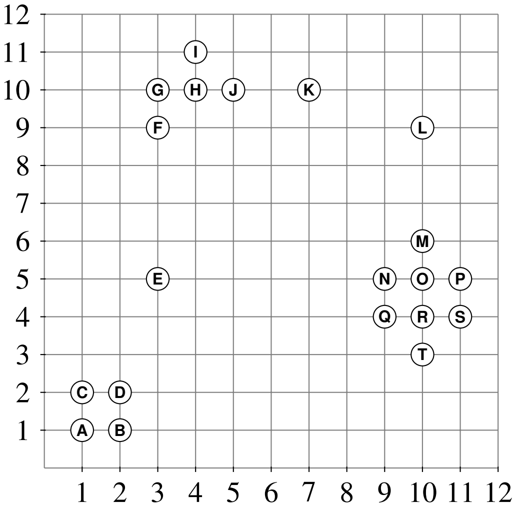
$\varepsilon = 2.1$

$minPts = 4$

Seed list:

Example 5

Start example   Skip example



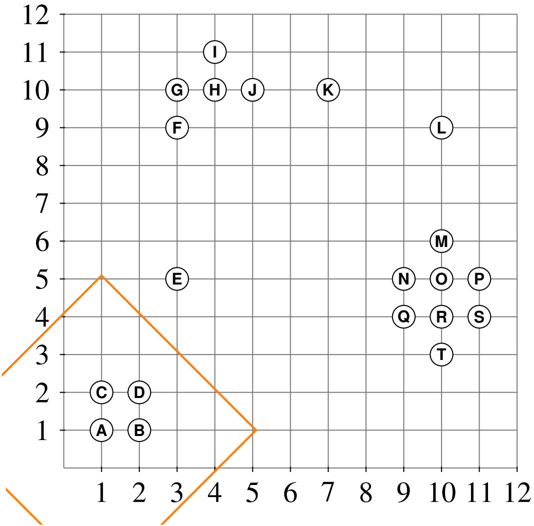
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example Skip example

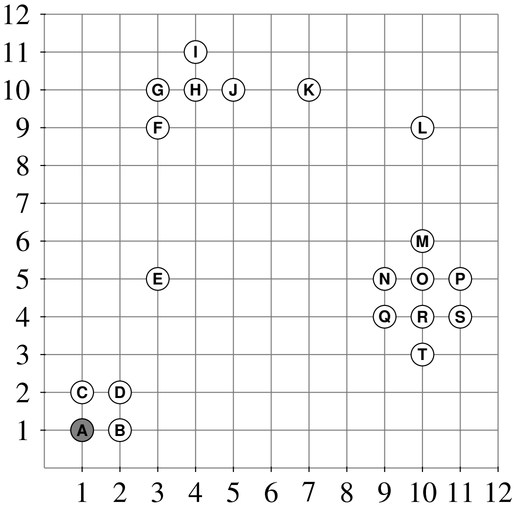


$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5



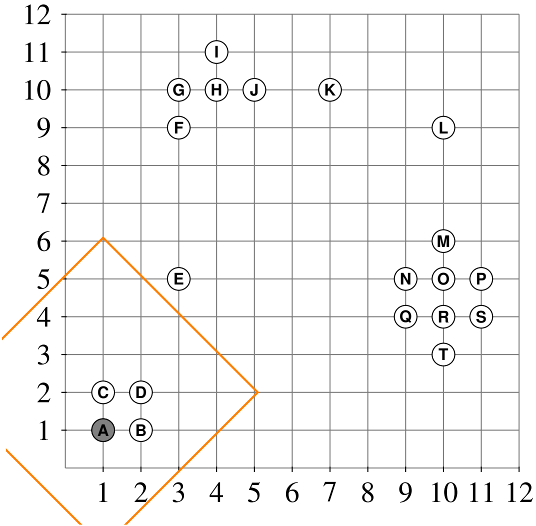
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example | Skip example



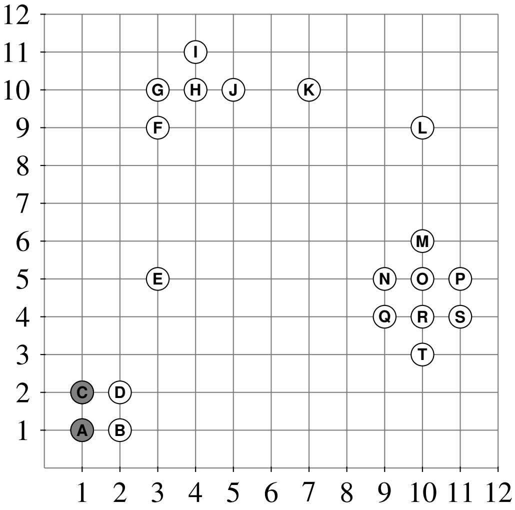
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example   Skip example



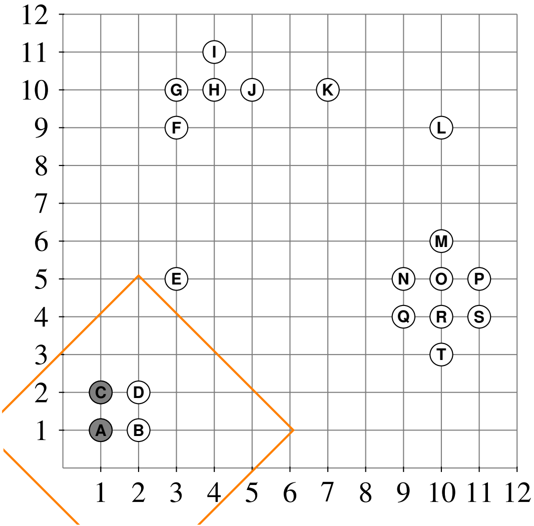
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example
Skip example



$\varepsilon = 4.1$

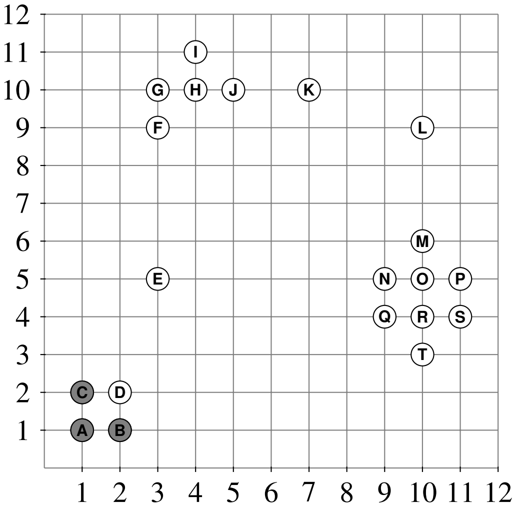
$minPts = 5$

Seed list:

Example 5

Start example

Skip example



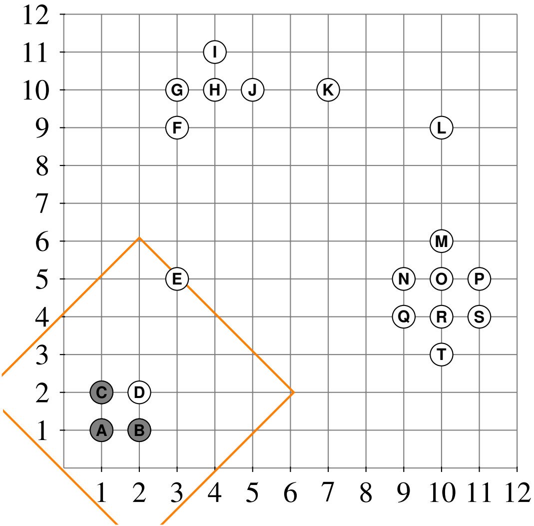
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example
Skip example



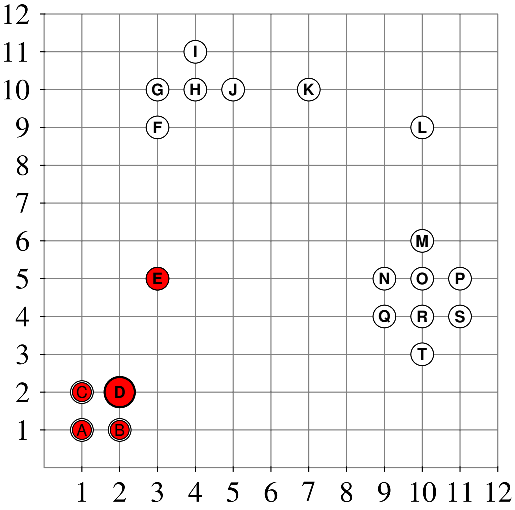
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

▸ Start example ⏩ Skip example



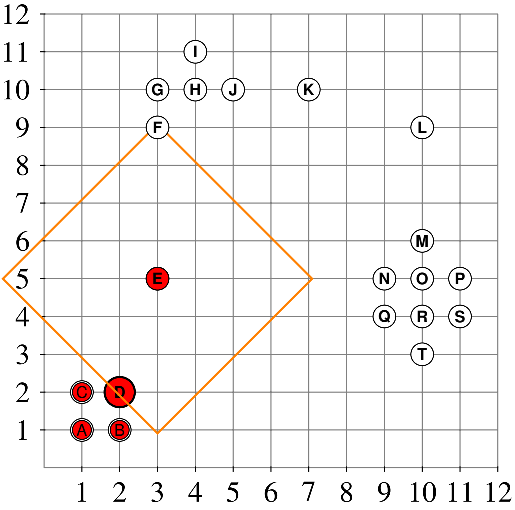
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
E

Example 5

▸ Start example ⏩ Skip example



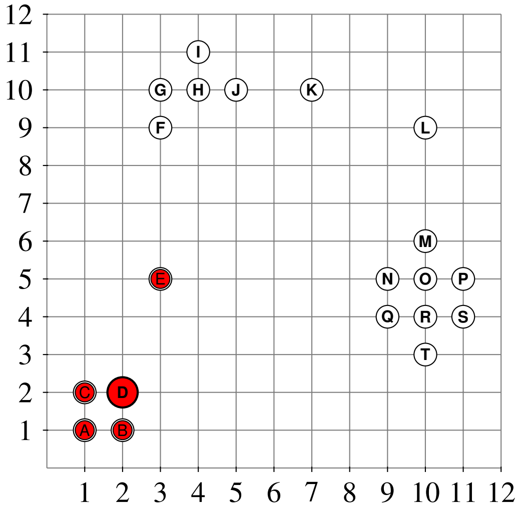
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example Skip example



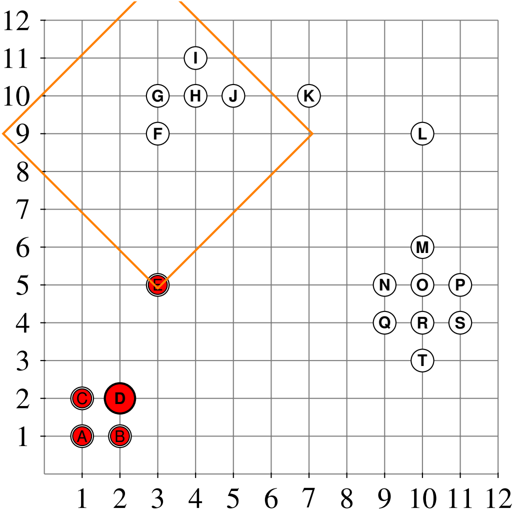
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example Skip example



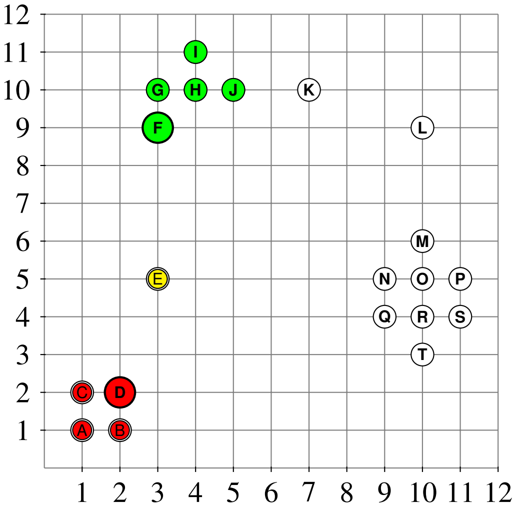
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example
Skip example



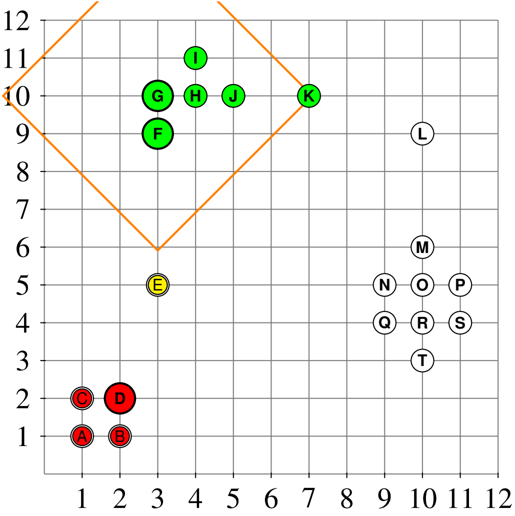
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
GHIJ

Example 5

Start example
Skip example



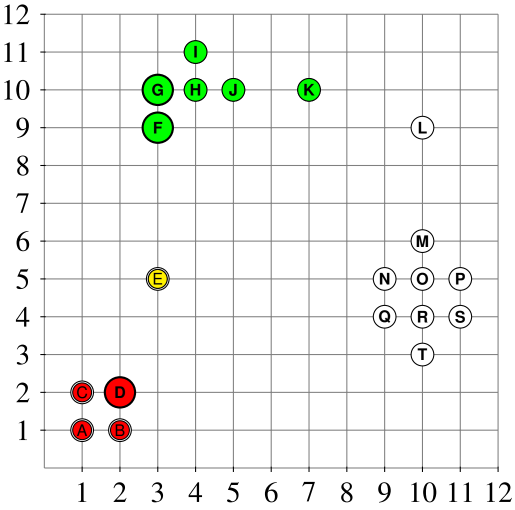
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
HIJ

Example 5

Start example
Skip example



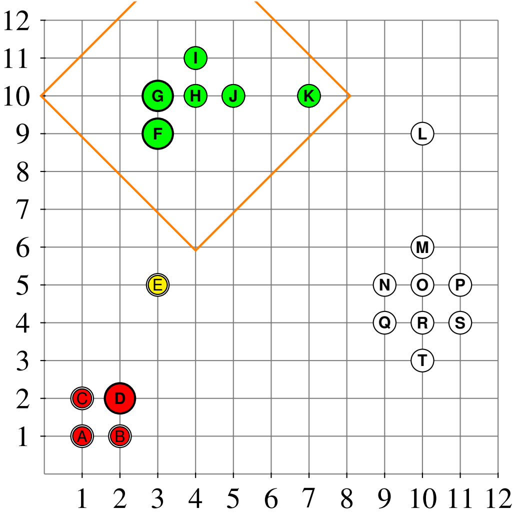
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
HIJK

Example 5

Start example | Skip example



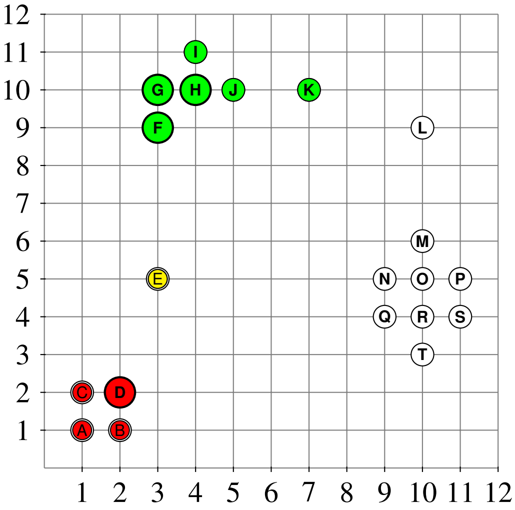
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
IJK

Example 5

Start example Skip example



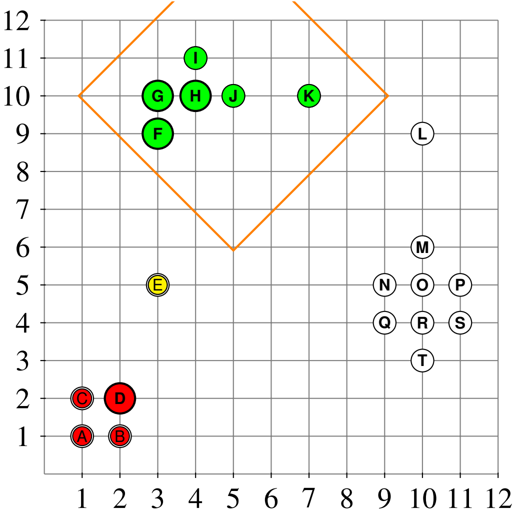
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
IJK

Example 5

Start example
Skip example



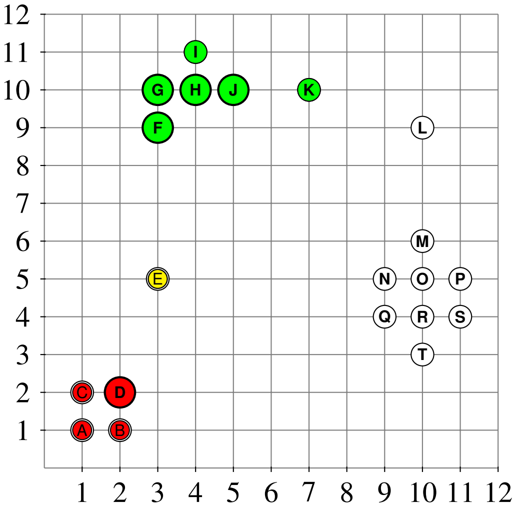
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
IK

Example 5

Start example Skip example



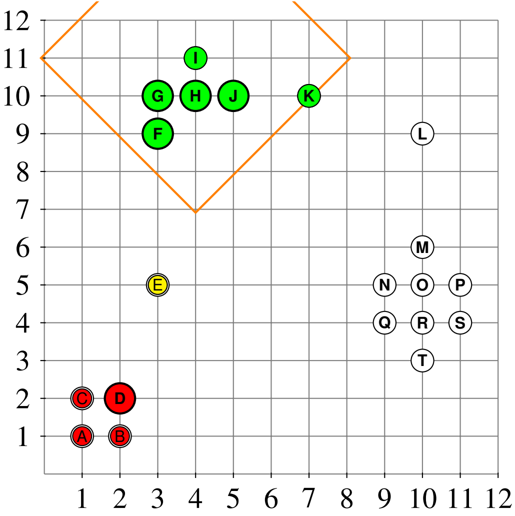
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
IK

Example 5

▸ Start example ▸▸ Skip example



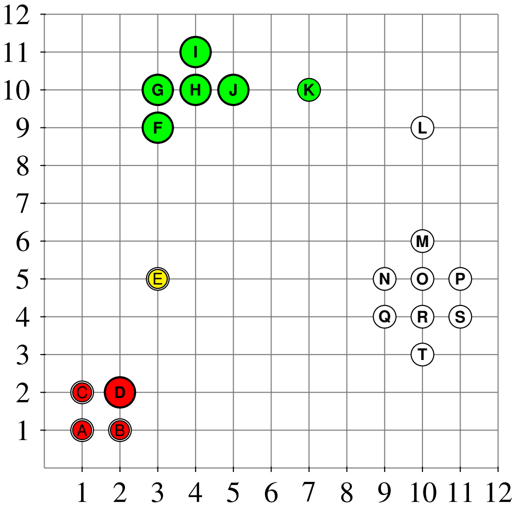
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
K

Example 5

Start example   Skip example



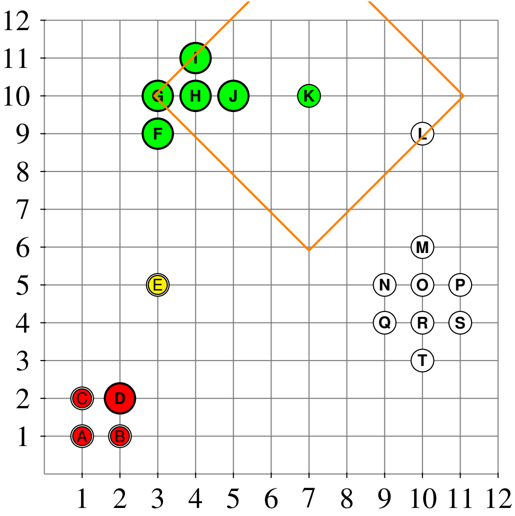
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
K

Example 5

Start example · Skip example



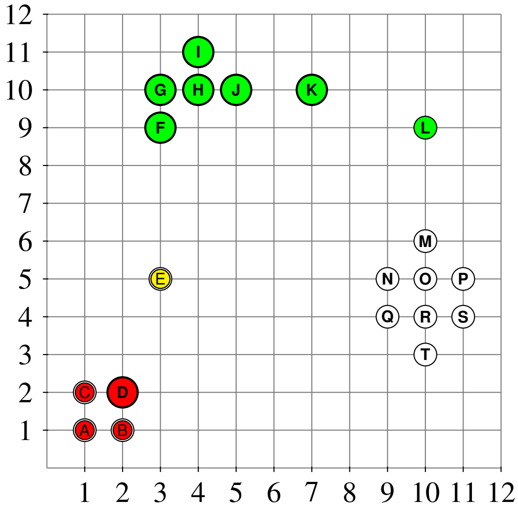
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example Skip example



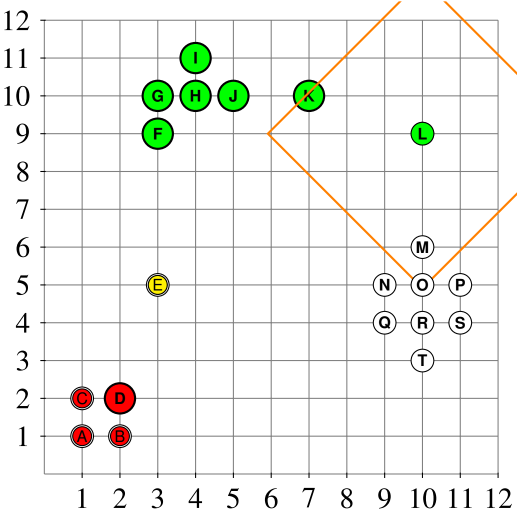
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
L

Example 5

▸ Start example ⏩ Skip example



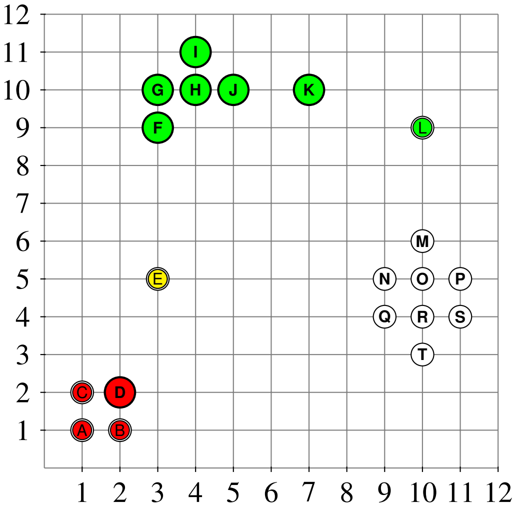
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example
Skip example



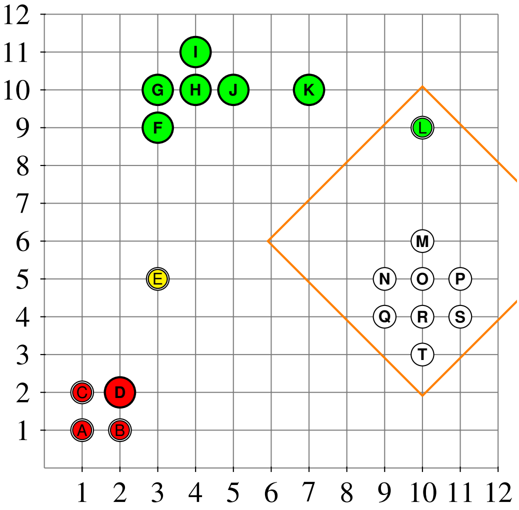
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example   Skip example



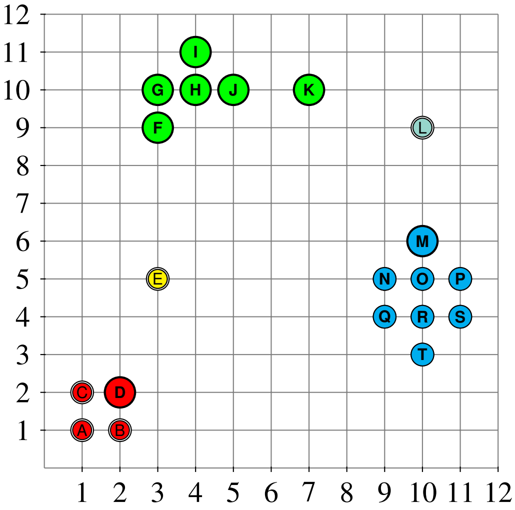
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example
Skip example



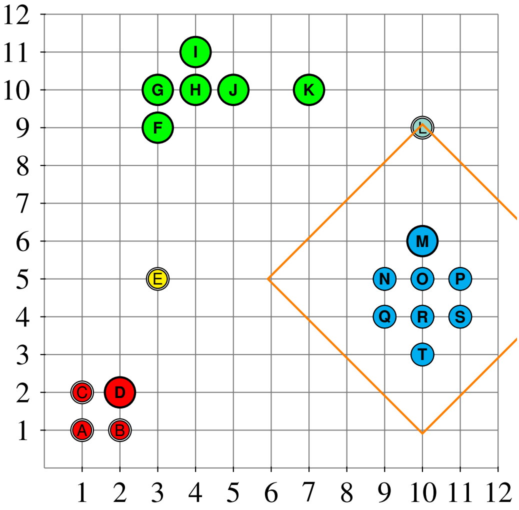
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
ONPRQST

Example 5

Start example · Skip example



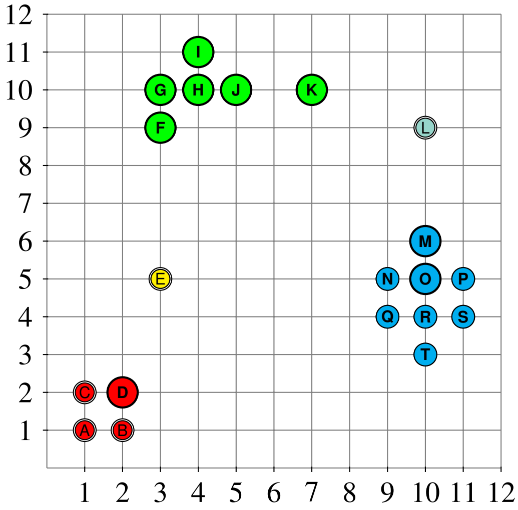
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
NPQST

Example 5

Start example
Skip example



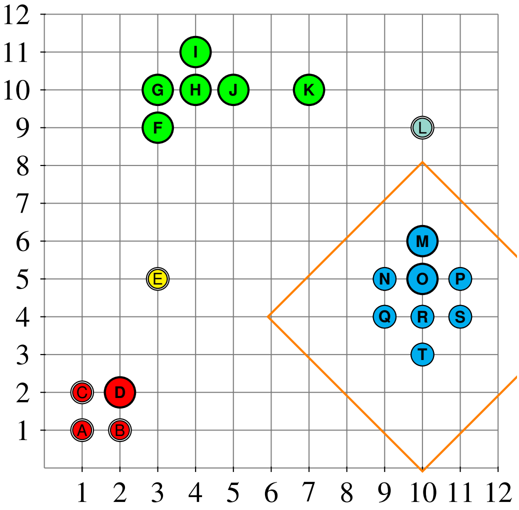
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
NPQST

Example 5

Start example | Skip example



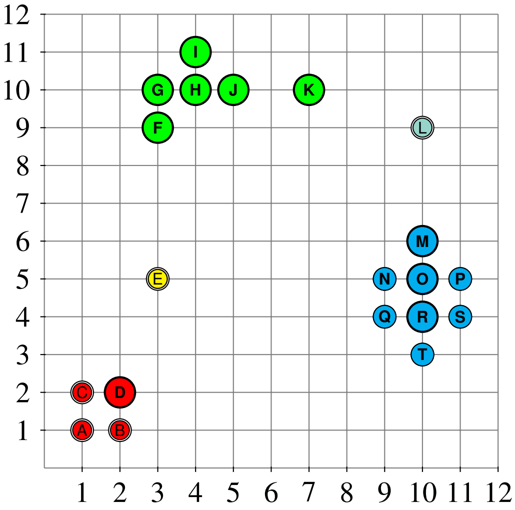
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
NPQST

Example 5

Start example   Skip example



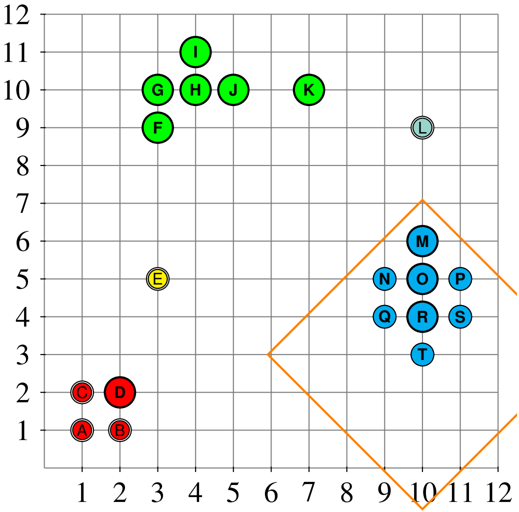
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
NPQST

Example 5

Start example · Skip example



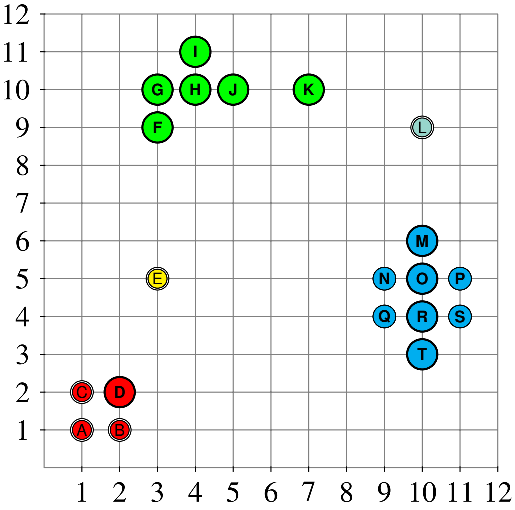
$\varepsilon = 4.1$

$minPts = 5$

Seed list: NPQS

Example 5

Start example | Skip example



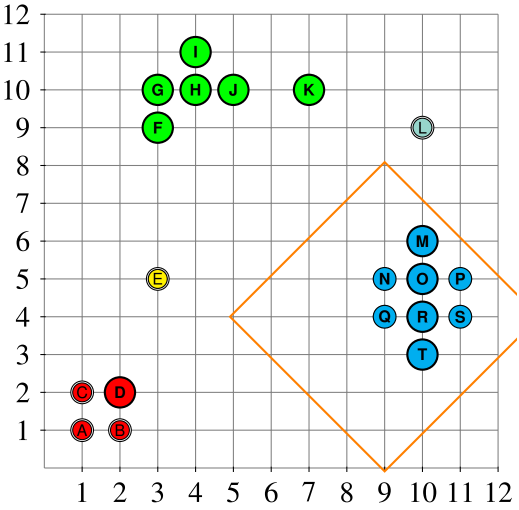
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
NPQS

Example 5

Start example
Skip example



$\varepsilon = 4.1$

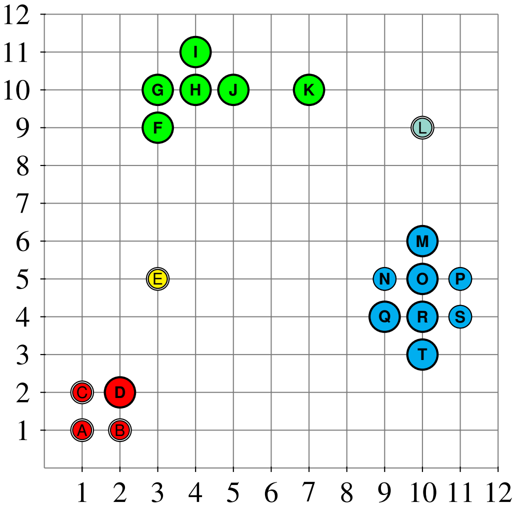
$minPts = 5$

Seed list:
NPS

Example 5

Start example

Skip example



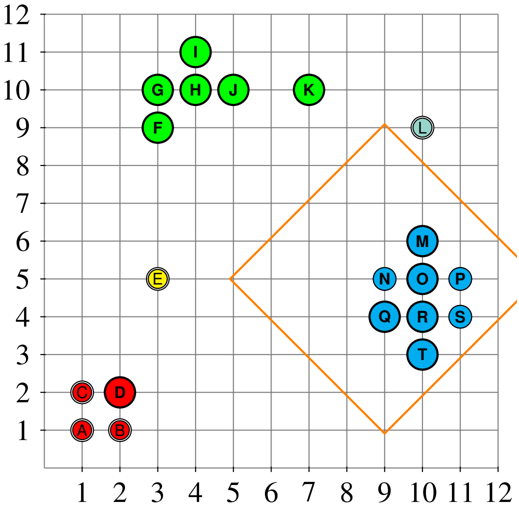
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
NPS

Example 5

▸ Start example ▸▸ Skip example



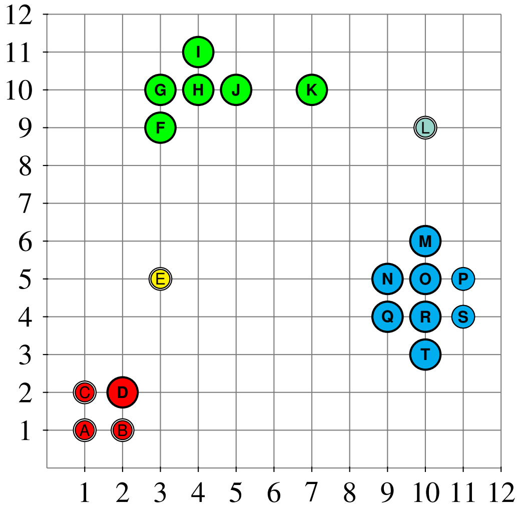
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
PS

Example 5

Start example
Skip example



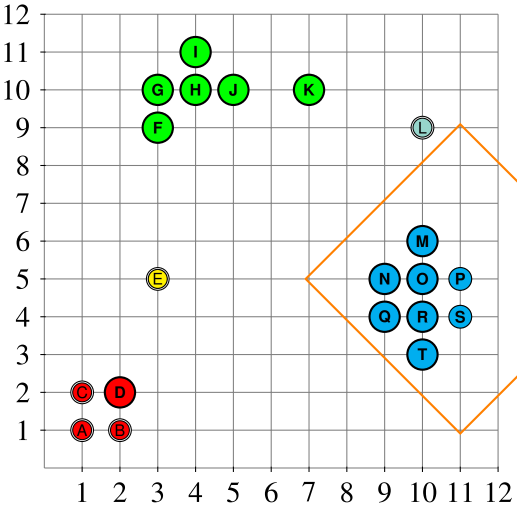
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
PS

Example 5

Start example
Skip example



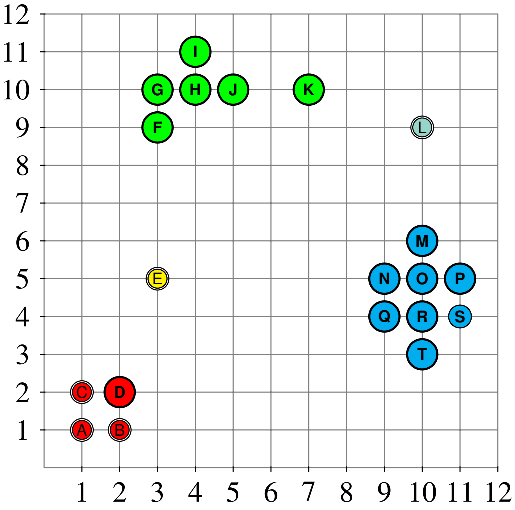
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
S

Example 5

Start example | Skip example



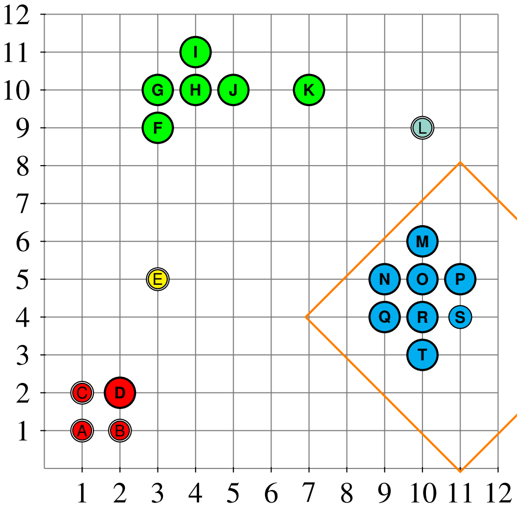
$\varepsilon = 4.1$

$minPts = 5$

Seed list:
S

Example 5

▸ Start example ⏩ Skip example



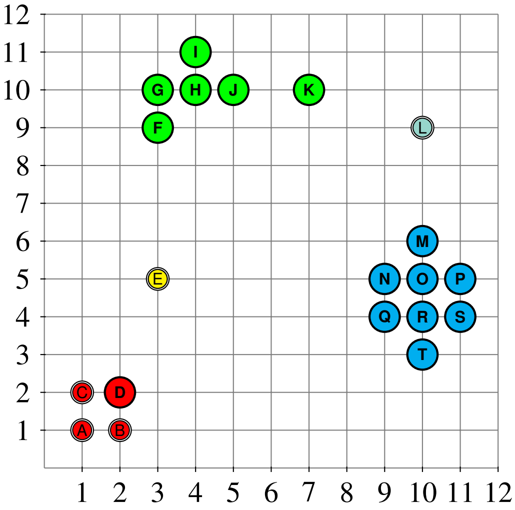
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example | Skip example



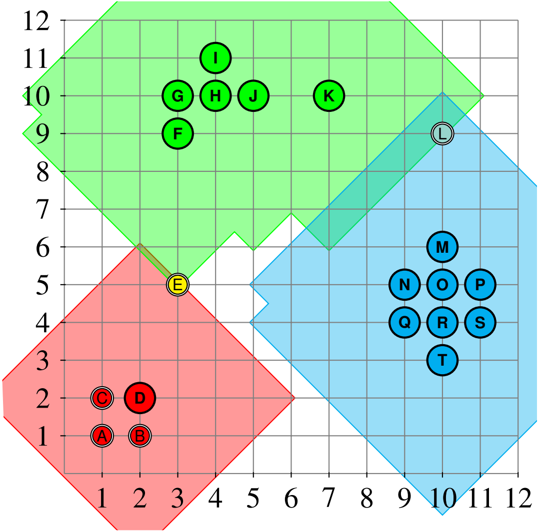
$\varepsilon = 4.1$

$minPts = 5$

Seed list:

Example 5

Start example · Skip example



$\varepsilon = 4.1$

$minPts = 5$

Seed list:

DBSCAN [Ester et al., 1996]

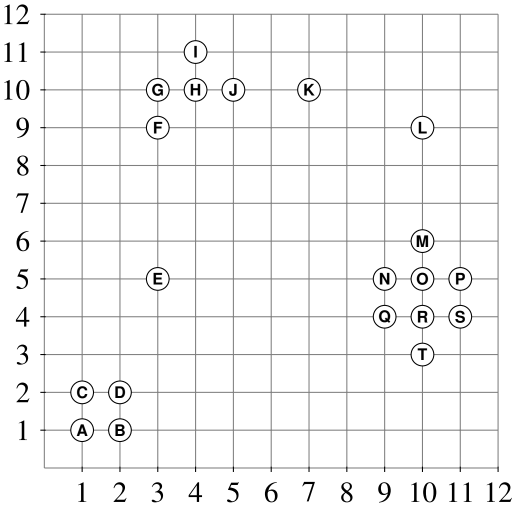
Algorithm 7.2 (DBSCAN – expandCluster)

```
ExpandCluster(DB, Point, C_Id, Eps, MinPts):  Boolean
  seeds := DB.rq(Point,Eps)
  IF seeds.size < MinPts THEN // no core point
    DB.changeC_Id(Point,NOISE);
    RETURN FALSE;
  ELSE // all points in seeds are dens-reach from Point
    DB.changeC_Ids(seeds,C_Id);
    seeds.delete(Point);
  WHILE seeds <> Empty DO
    currentP := seeds.first();
    result := DB.rq(currentP,Eps);
    IF result.size >= MinPts THEN
      FOR i FROM 1 TO result.size DO
        resultP := result.get(i);
        IF resultP.C_Id IN {UNCLASSIFIED,NOISE} THEN
          IF resultP.CiId = UNCLASSIFIED THEN
            seeds.append(resultP);
          DB.changeC_Id(resultP,C_Id);
    seeds.delete(currentP);

  RETURN TRUE;
```

Example 6

Start example | Skip example



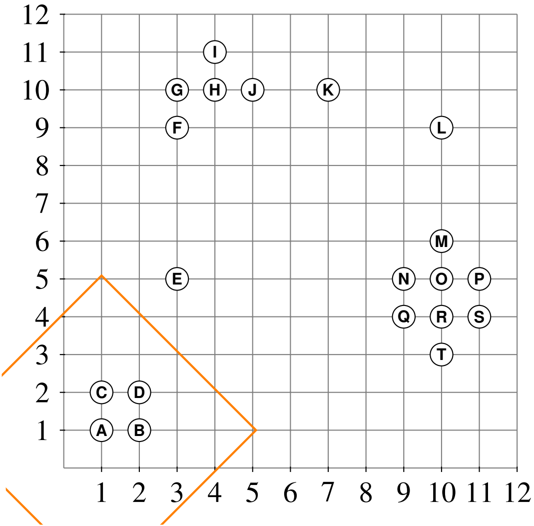
$\varepsilon = 4.1$

$minPts = 4$

Seed list:

Example 6

Start example
Skip example



$\varepsilon = 4.1$

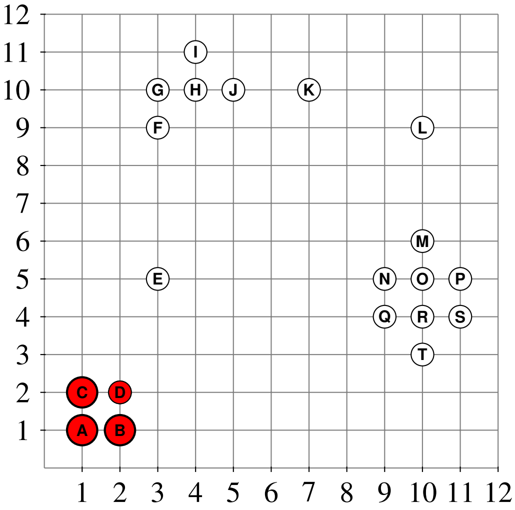
$minPts = 4$

Seed list:

Example 6

Start example

Skip example



$\varepsilon = 4.1$

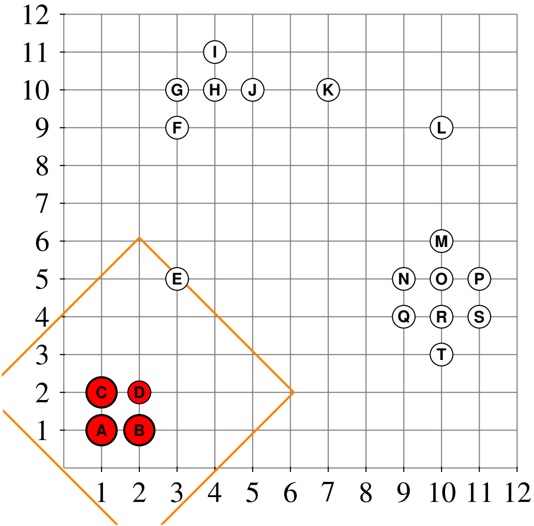
$minPts = 4$

Seed list:
D

Example 6

Start example

Skip example



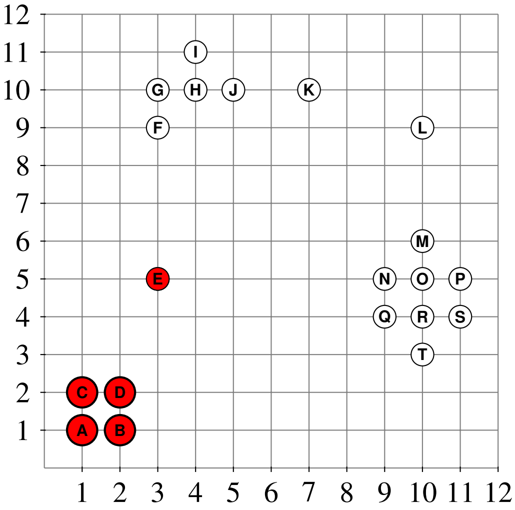
$\varepsilon = 4.1$

$minPts = 4$

Seed list:

Example 6

Start example | Skip example



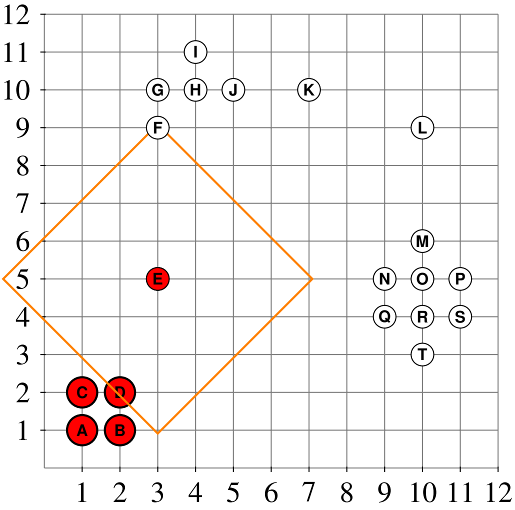
$\varepsilon = 4.1$

$minPts = 4$

Seed list:
E

Example 6

Start example Skip example



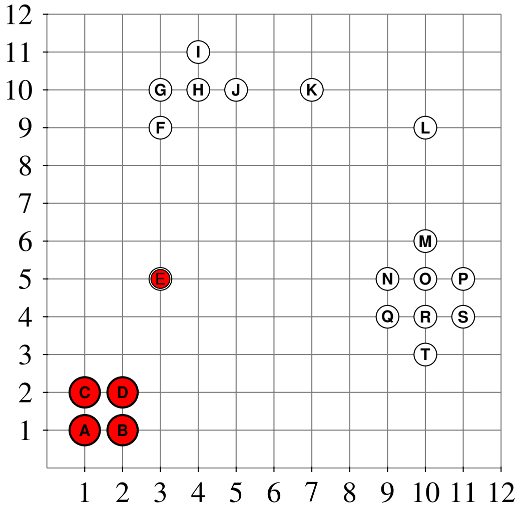
$\varepsilon = 4.1$

$minPts = 4$

Seed list:

Example 6

Start example   Skip example



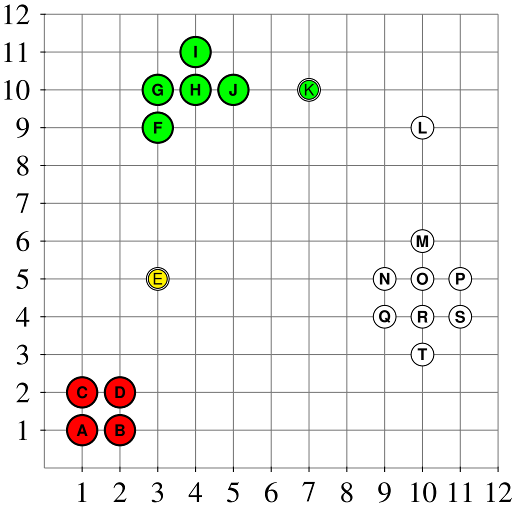
$\varepsilon = 4.1$

$minPts = 4$

Seed list:

Example 6

Start example
Skip example



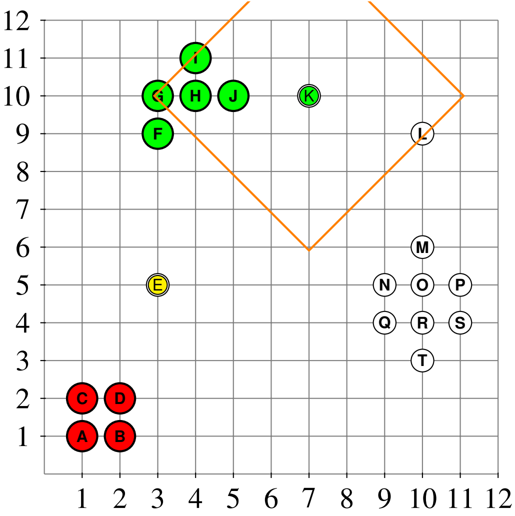
$\varepsilon = 4.1$

$minPts = 4$

Seed list:
K

Example 6

Start example Skip example



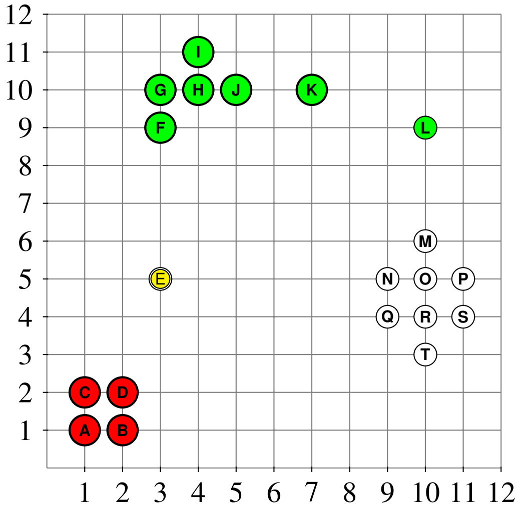
$\varepsilon = 4.1$

$minPts = 4$

Seed list:

Example 6

Start example
Skip example



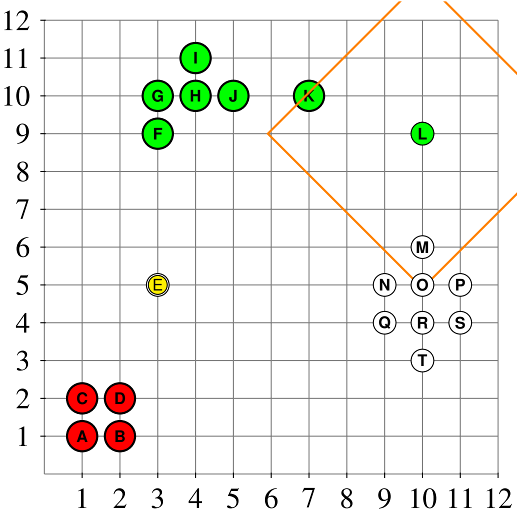
$\varepsilon = 4.1$

$minPts = 4$

Seed list:
L

Example 6

Start example · Skip example



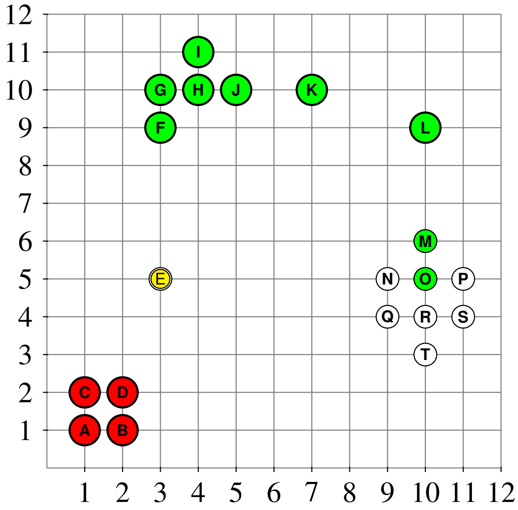
$\varepsilon = 4.1$

$minPts = 4$

Seed list:

Example 6

Start example
Skip example



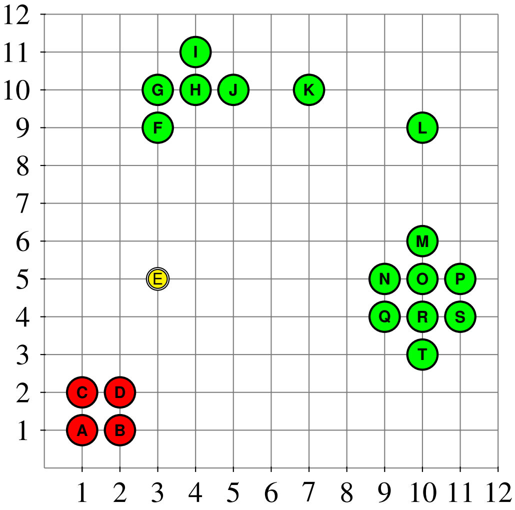
$\varepsilon = 4.1$

$minPts = 4$

Seed list:
MO

Example 6

Start example | Skip example



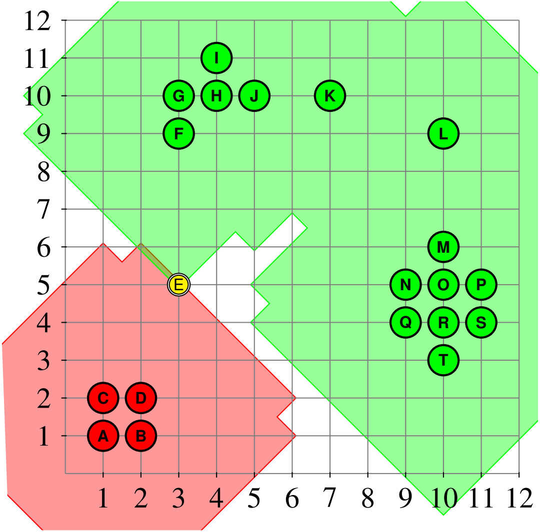
$\varepsilon = 4.1$

$minPts = 4$

Seed list:

Example 6

Start example Skip example



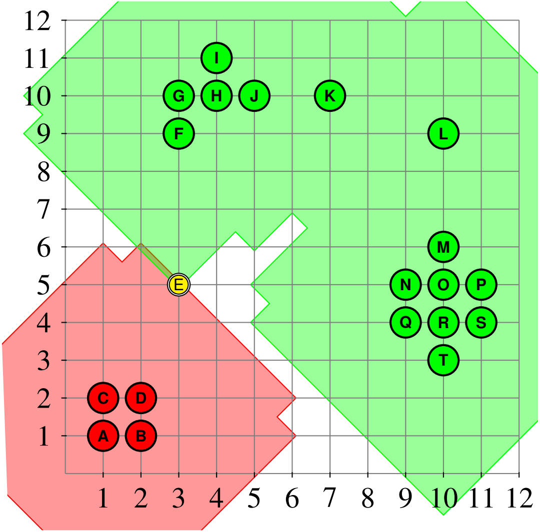
$\varepsilon = 4.1$

$minPts = 4$

Seed list:

Example 6

▸ Start example ▸▸ Skip example



$\varepsilon = 4.1$

$minPts = 4$

Seed list:

single-link like effect!

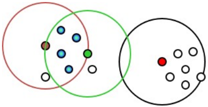
Shared Nearest Neighbor (SNN) Clustering

- DBSCAN can detect clusters of different size and shape
- but cannot find clusters for different density thresholds.

Improvement: different notion of similarity

- similarity of objects, if both are close to a reference set
- similarity is “approved” by reference set
- example: reference set is set of common neighbors, similarity is cardinality of this set
- shared nearest neighbor (SNN) similarity ($\mathrm{NN}_\mathrm{k}(o)$: set of the k nearest neighbors of o):

$$\mathrm{SNN}_k(p,q) = |\,\mathrm{NN}_\mathrm{k}(p) \cap \mathrm{NN}_\mathrm{k}(q)|$$



SNN$_6$-similarity(●,●) = 4

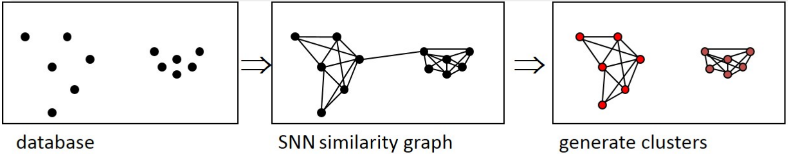
SNN$_6$-similarity(●,●) = 0

See also studies by Houle et al. [2010], Bernecker et al. [2011] on the quality improvement in similarity search by using SNN.

SNN clustering [Jarvis and Patrick, 1973]

Algorithm 7.3 (SNN clustering [Jarvis and Patrick, 1973])

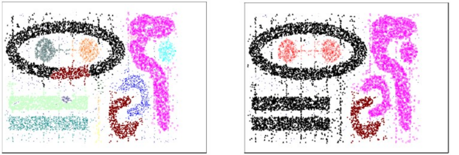
1. *compute similarity matrix and similarity graph*
   - *for all pairs $o, p \in \mathcal{D}$: compute SNN_k similarity*
   - *SNN similarity graph: nodes=objects, edges weight=SNN similarity*
   - *no edge where SNN similarity = 0*
2. *generate clusters*
   - *delete edges with weight below threshold τ*
   - *cluster: connected component in the resulting graph*



Problems of the Algorithm of Jarvis and Patrick [1973]

Problem:

- hard to find a good threshold τ
- small variations lead to very different solutions



(a) No-merging Threshold (b) No-merging Threshold + 1

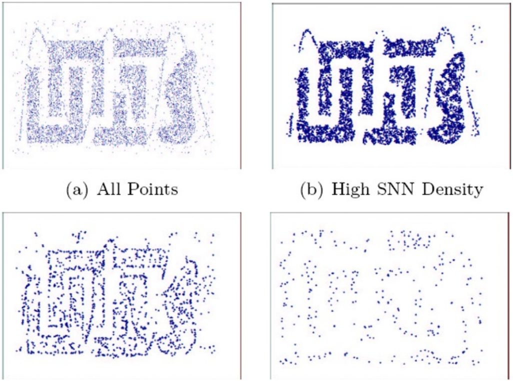
picture from Ertöz et al. [2003]

Solution of Ertöz et al. [2003]:

- combine SNN similarity with density-based concepts
- SNN density: number of points within a given radius ε w.r.t. SNN similarity

$$\mathrm{SNN}_k\text{-density}(p, \varepsilon) = |\{q | \mathrm{SNN}_k(p, q) \geq \varepsilon\}|$$

SNN clustering [Ertöz et al., 2003]



(a) All Points (b) High SNN Density (c) Medium SNN Density (d) Low SNN Density

picture from Ertöz et al. [2003]

SNN clustering [Ertöz et al., 2003]

Algorithm 7.4 (SNN clustering [Ertöz et al., 2003])

given k, ε, MinPts:

1. *compute similarity matrix and -graph (as for Alg. 7.3)*
2. *compute SNN density for each point w.r.t.* ε
3. *derive core points w.r.t.* MinPts
4. *cluster core points* p, q *if* $\mathrm{SNN}_k(p, q) \geq \varepsilon$
5. *assign a non-core point* p *to cluster* C *if there is a core point* $q \in C$ *with* $\mathrm{SNN}_k(p, q) \geq \varepsilon$
6. *all remaining non-core points are noise*

Note that:

Steps 2 to 6 are nothing but DBSCAN when using a different notion of distance!

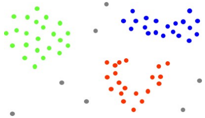
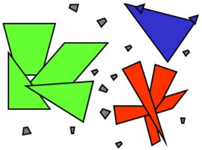
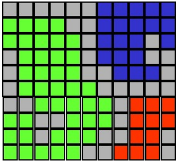
Generalization of DBSCAN [Sander et al., 1998]

basic idea of DBSCAN:
ε-neighborhood contains at least MinPts points

$\swarrow$	$\searrow$
$\mathrm{dist} \leq \varepsilon$	$\lvert \mathrm{RQ}(o, \varepsilon) \rvert \geq \mathrm{MinPts}$
$\downarrow$	$\downarrow$
$\mathrm{NPred}(o, p)$	$\mathrm{MinWeight}(N)$
reflexive, symmetric for pairs of objects	arbitrary predicate for set of objects
$\downarrow$	$\downarrow$
generalized neighborhood	generalized minimal cardinality
$N_{\mathrm{NPred}}(o) = \{p \mid \mathrm{NPred}(o, p)\}$	$\mathrm{MinWeight}(N_{\mathrm{NPred}}(o))$
$\searrow$	$\swarrow$

NPred neighborhood has at least “weight” MinWeight.

Examples

			
NPred	$\mathrm{dist}(p, q) \leq \varepsilon$	intersect(p, q)	neighbor cell and similar color
$\mathrm{MinWeight}$	cardinality$(\ldots)$ $\geq \mathrm{MinPts}$	sum of area $\geq$ 5% of total area	`true`

Algorithm GDBSCAN [Sander et al., 1998]

- same algorithmic schema as DBSCAN
- instead of $\mathrm{RQ}(o, \varepsilon)$: N_{NPred}-query
- instead of condition $|\mathrm{RQ}(o, \varepsilon)| \geq \mathrm{MinPts}$: evaluate $\mathrm{MinWeight}$ predicate
- almost arbitrary NPred possible (reflexive and symmetric)

Note that:

Complexity depends mainly on the complexity of the N_{NPred}-query.

Outline

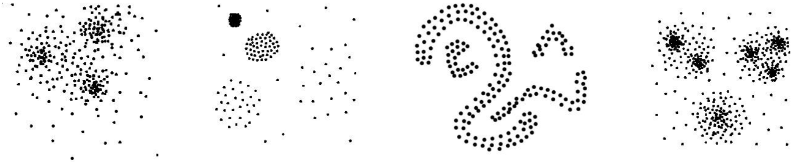
Recommended Reading

Recommended Reading:

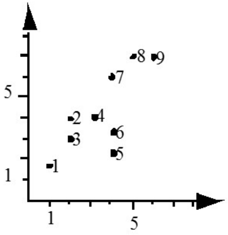
- *Tan et al. [2006], Chapter 8.3.*
- *Tan et al. [2020], Chapter 5.3.*
- *Zaki and Meira Jr. [2014], Chapter 14.*
- *Discussion of the field by Campello et al. [2020, 2015].*

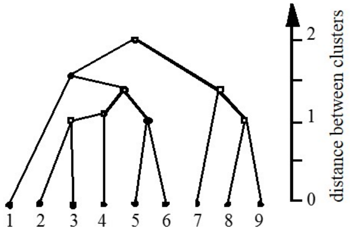
General Idea

- Instead of deriving a partition of the dataset into a flat set of clusters, derive a hierarchy (dendrogram) of clusters.
- A cluster at a higher level can contain several smaller clusters at a lower level.
- A tree represents a hierarchical clustering:
  - root: represents the complete database
  - leaf: represents a single object
  - inner node: represents a cluster containing all the objects of the subtree rooted at this node



Example





categories of hierarchical clustering algorithms (typically best-first heuristic):

- bottom-up construction of the dendrogram (agglomerative), at each node merging two clusters
- top-down construction of the dendrogram (divisive) at each node splitting a cluster into two parts

Agglomerative Algorithm

Algorithm 7.5 (Agglomerative Hierarchical Clustering)

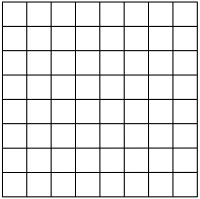
1. *Initial clusters consist of one object each.*
2. *Compute distances between all pairs of clusters.*
3. *Merge those two clusters with smallest distance.*
4. *Compute distance between the new cluster and all remaining clusters.*
5. *If only one cluster remains: terminate; otherwise: repeat from step 3.*

Complexity

initialization:

similarity matrix $\mathcal{O}(n^2)$

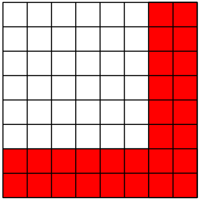
(pairwise similarity between n objects)



iteration i:

build a new cluster $\mathcal{O}((n-i+1)^2)$

compute new similarities $\mathcal{O}(n-i+1)$



n iterations $\Rightarrow \mathcal{O}(n^3)$

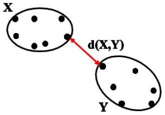
Distances Between Clusters

- Different approaches are based on different distance measures.
- Using a particular distance measure can allow for algorithmic enhancements.
- Some common distance measures are:
  - single link
  - complete link
  - average link (a.k.a.: UPGMA – unweighted pair group method with arithmetic mean)

Single Link

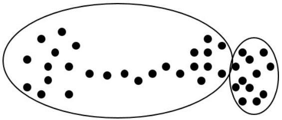
Let $\mathrm{dist} : \mathcal{D} \times \mathcal{D} \to \mathbb{R}_0^+$ be a distance measure for pairs of objects, let $X, Y \subseteq \mathcal{D}$ designate sets of objects.

$$\mathrm{dist}_{\text{single link}}(X, Y) = \min_{x \in X, y \in Y} \mathrm{dist}(x, y)$$



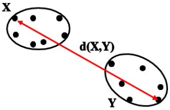
Single-Link

- efficient algorithm: SLINK [Sibson, 1973]: $\mathcal{O}(n^2)$
- single-link-effect: chain-like clusters, due to the merging of clusters through some chain of objects, yields:
  - clusters with high variance
  - clusters with an extended structure



Complete Link

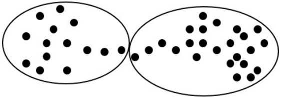
Let $\mathrm{dist} : \mathcal{D} \times \mathcal{D} \to \mathbb{R}_0^+$ be a distance measure for pairs of objects, let $X, Y \subseteq \mathcal{D}$ designate sets of objects.



Complete-Link

$$\mathrm{dist}_{\text{complete link}}(X, Y) = \max_{x \in X, y \in Y} \mathrm{dist}(x, y)$$

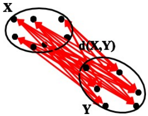
- efficient algorithm: CLINK [Defays, 1977]: $\mathcal{O}(n^2)$
- complete-link-effect:
  - small, strongly delimited clusters
  - convex clusters with similar diameters



Average Link

Let $\mathrm{dist} : \mathcal{D} \times \mathcal{D} \to \mathbb{R}_0^+$ be a distance measure for pairs of objects, let $X, Y \subseteq \mathcal{D}$ designate sets of objects.

$$\mathrm{dist}_{\text{average link}}(X, Y) = \frac{1}{|X| \cdot |Y|} \cdot \sum_{x \in X, y \in Y} \mathrm{dist}(x, y)$$



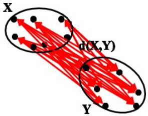
Average-Link

- sort of a compromise between single link and complete link
- no efficient algorithm known, i.e., $\mathcal{O}(n^3)$

Average Link

Let $\mathrm{dist} : \mathcal{D} \times \mathcal{D} \to \mathbb{R}_0^+$ be a distance measure for pairs of objects, let $X, Y \subseteq \mathcal{D}$ designate sets of objects.

$$\mathrm{dist}_{\text{average link}}(X, Y) = \frac{1}{|X| \cdot |Y|} \cdot \sum_{x \in X, y \in Y} \mathrm{dist}(x, y)$$



Average-Link

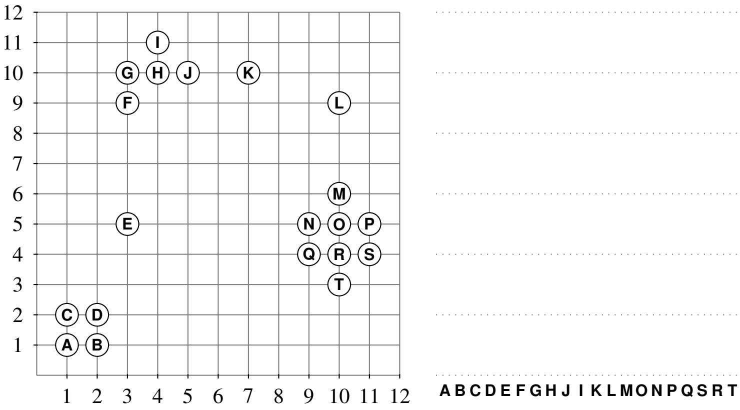
- sort of a compromise between single link and complete link
- no efficient algorithm known, i.e., $\mathcal{O}(n^3)$

Note that:

Fundamental properties of distance measures do not hold for some of these distance measures – think about it!

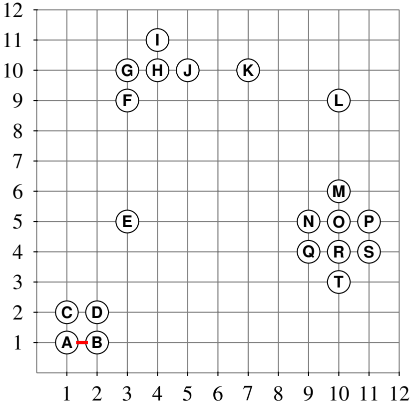
Example: Single Link Clustering

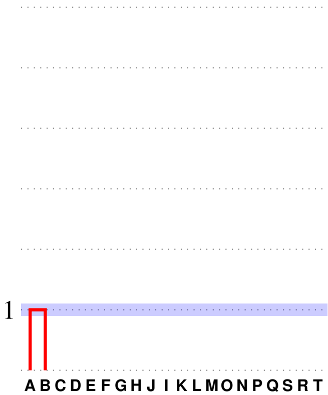
Start example Skip example



Example: Single Link Clustering

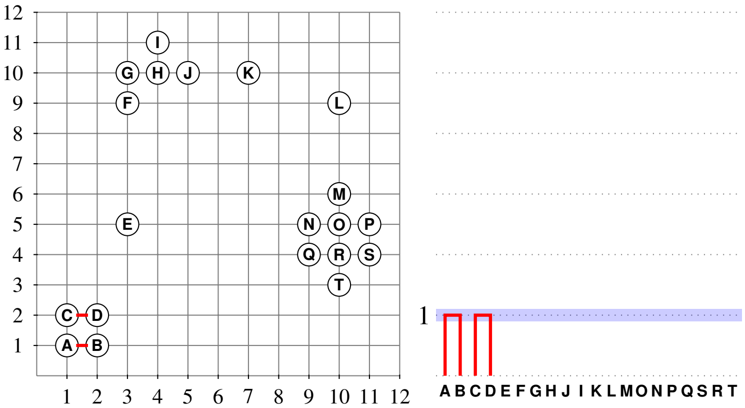
Start example · Skip example





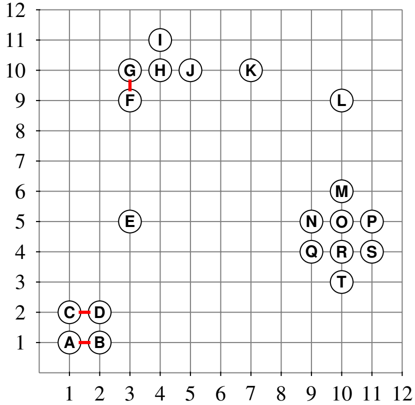
Example: Single Link Clustering

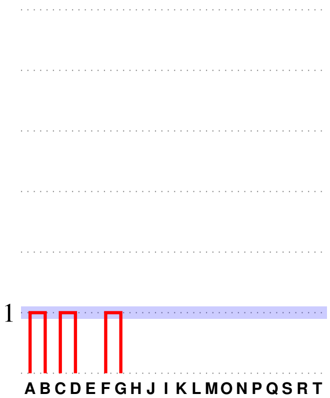
Start example   Skip example



Example: Single Link Clustering

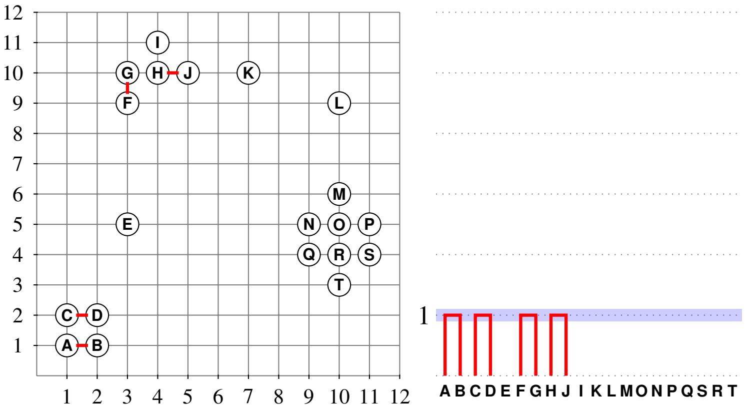
▸ Start example ⏩ Skip example





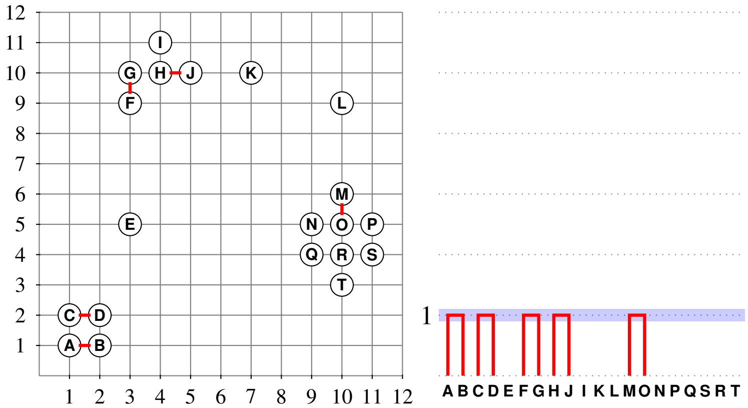
Example: Single Link Clustering

Start example   Skip example



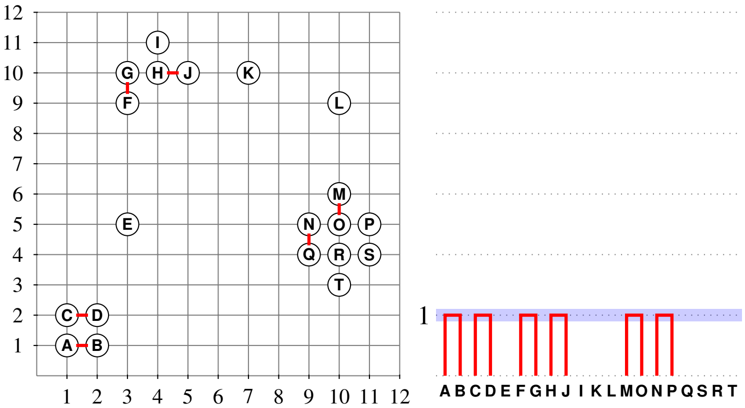
Example: Single Link Clustering

Start example Skip example



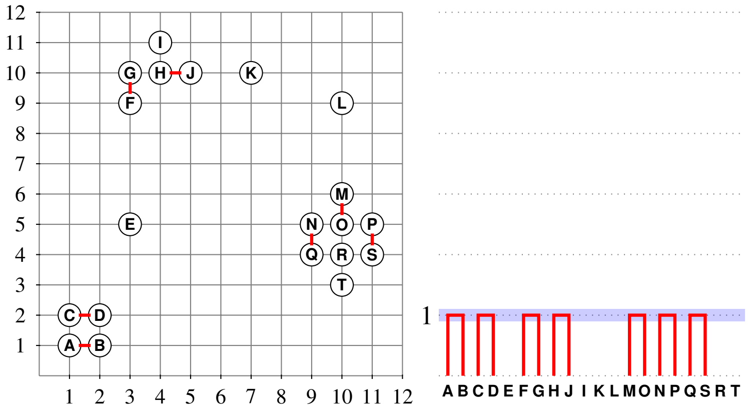
Example: Single Link Clustering

Start example Skip example



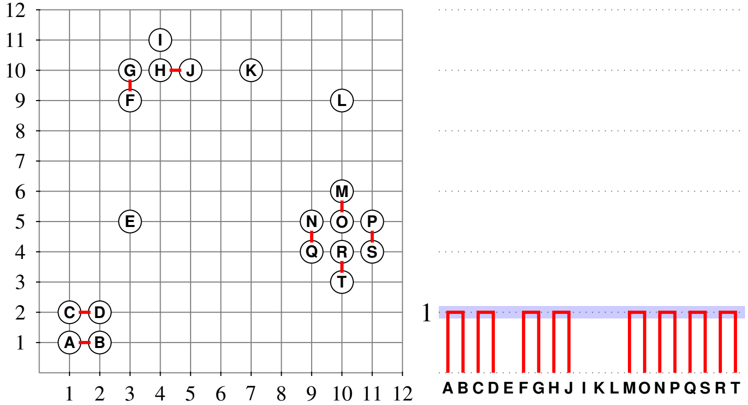
Example: Single Link Clustering

Start example
Skip example



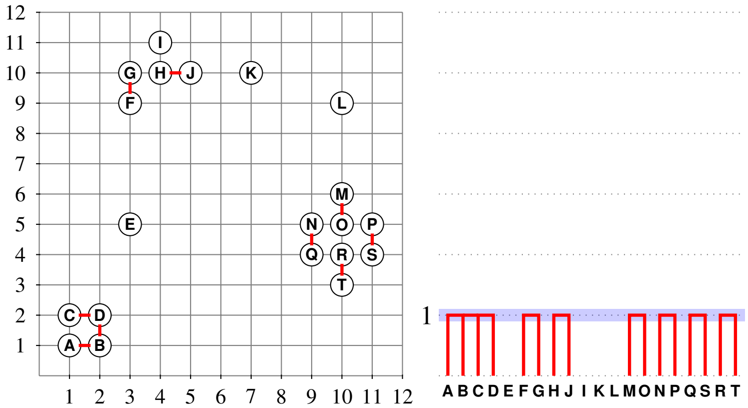
Example: Single Link Clustering

▸ Start example ▸▸ Skip example



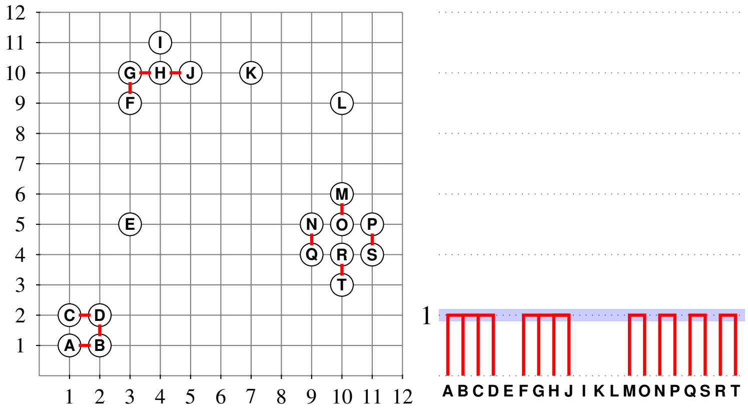
Example: Single Link Clustering

Start example Skip example



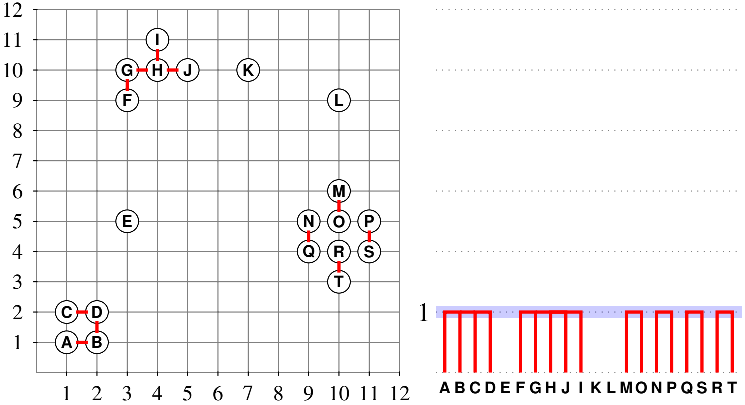
Example: Single Link Clustering

Start example   Skip example



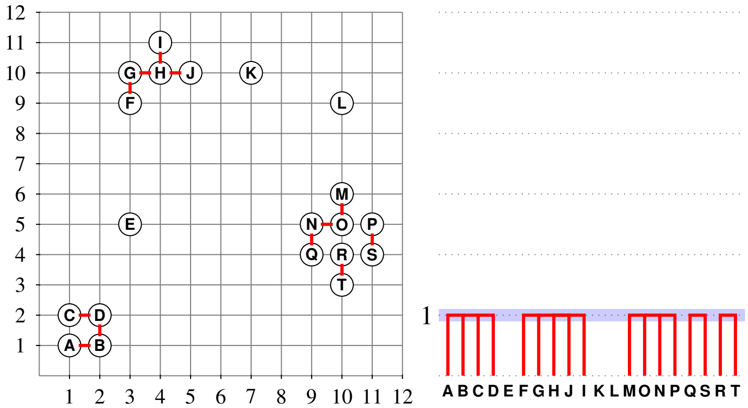
Example: Single Link Clustering

Start example | Skip example



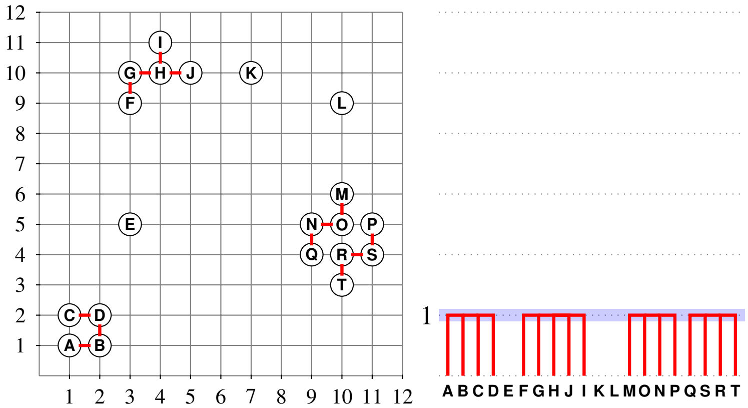
Example: Single Link Clustering

Start example
Skip example



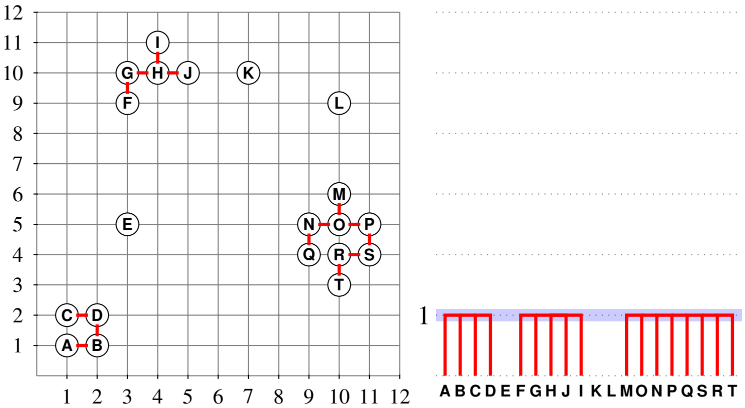
Example: Single Link Clustering

Start example   Skip example



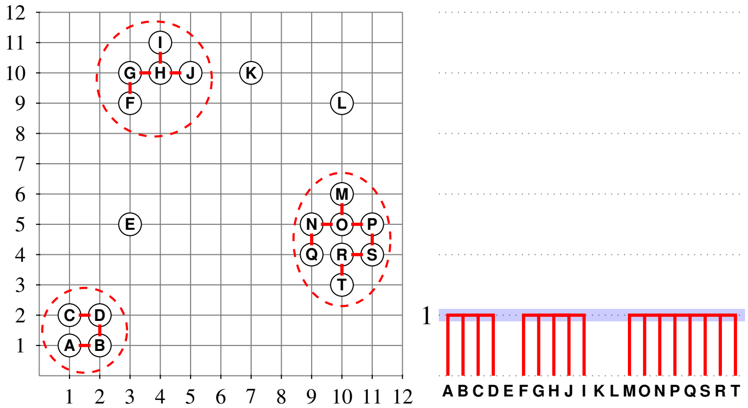
Example: Single Link Clustering

Start example | Skip example



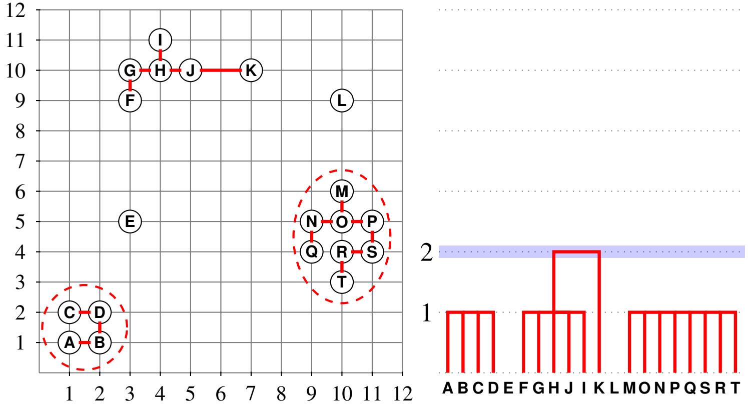
Example: Single Link Clustering

Start example   Skip example



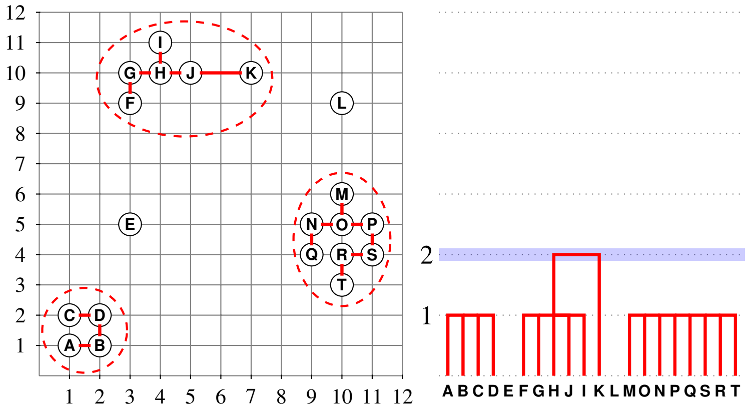
Example: Single Link Clustering

Start example   Skip example



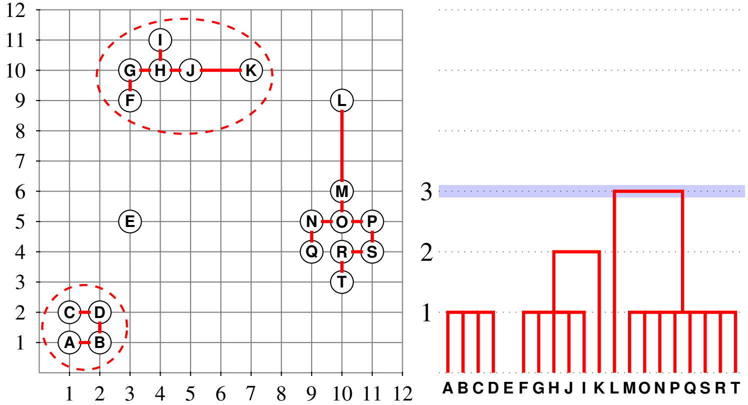
Example: Single Link Clustering

Start example Skip example



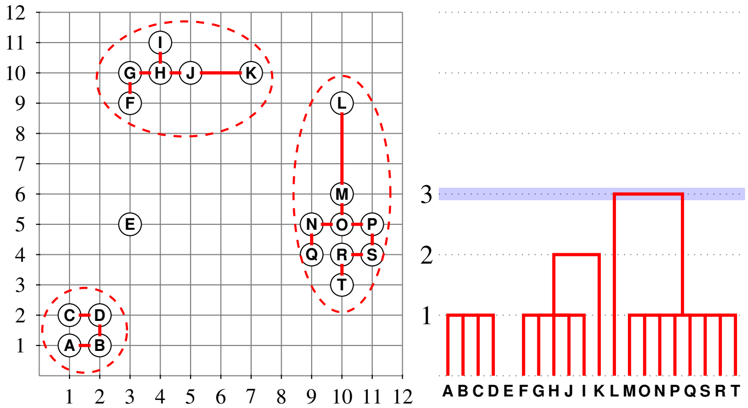
Example: Single Link Clustering

Start example   Skip example



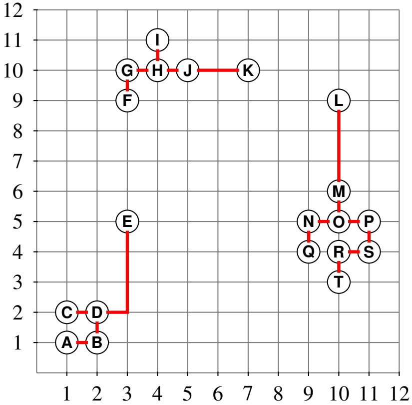
Example: Single Link Clustering

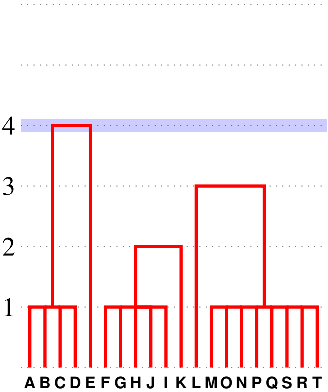
Start example   Skip example



Example: Single Link Clustering

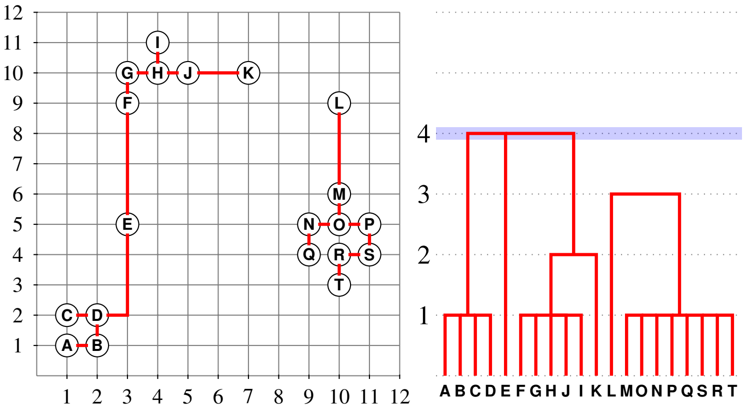
Start example
Skip example





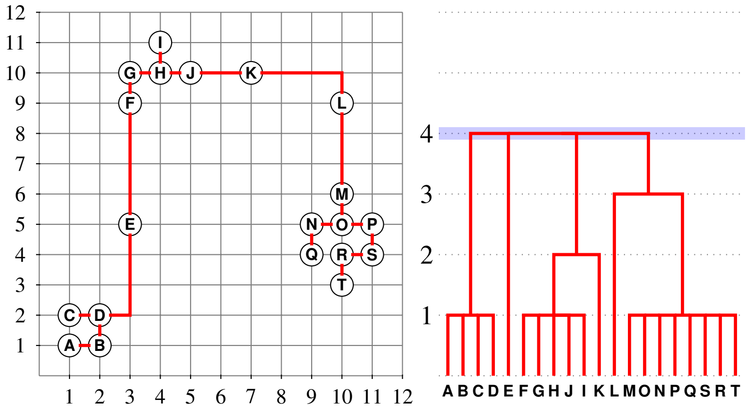
Example: Single Link Clustering

Start example Skip example



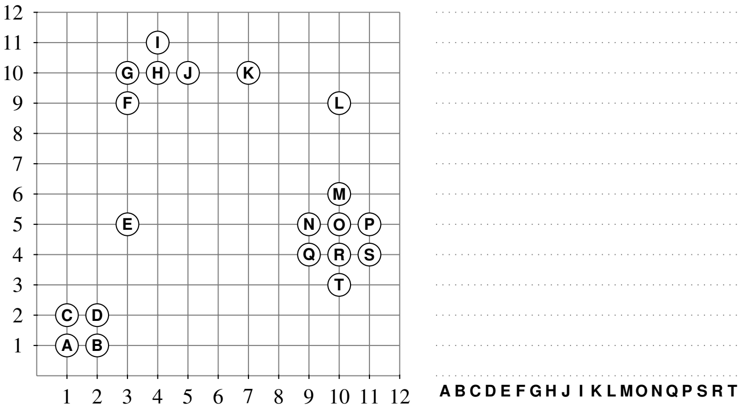
Example: Single Link Clustering

Start example
Skip example



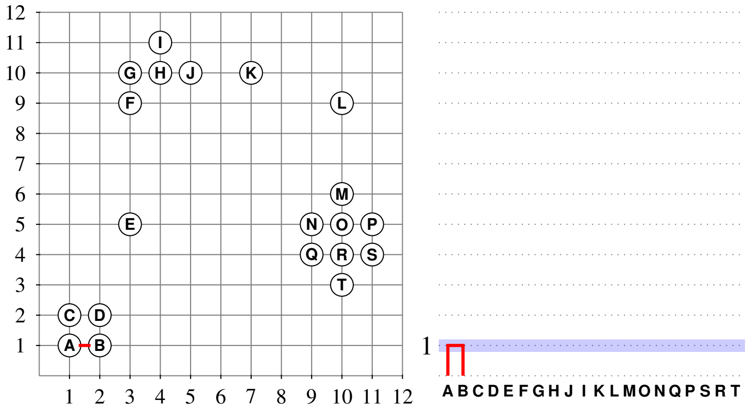
Example Complete Link Clustering

Start example   Skip example



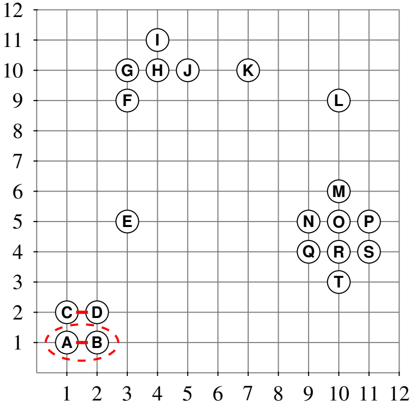
Example Complete Link Clustering

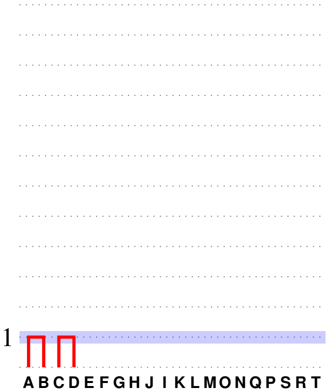
Start example   Skip example



Example Complete Link Clustering

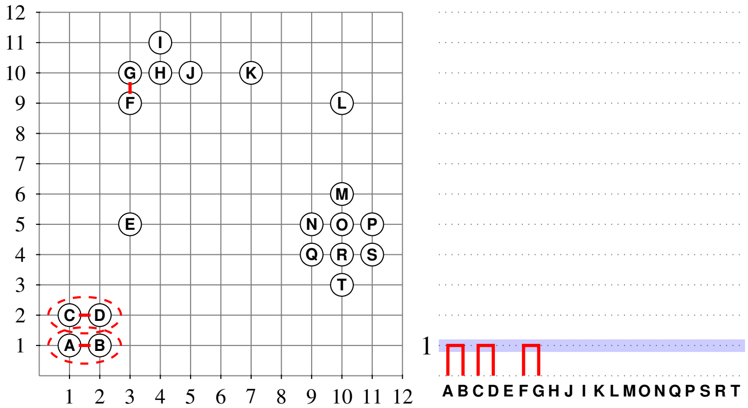
Start example
Skip example





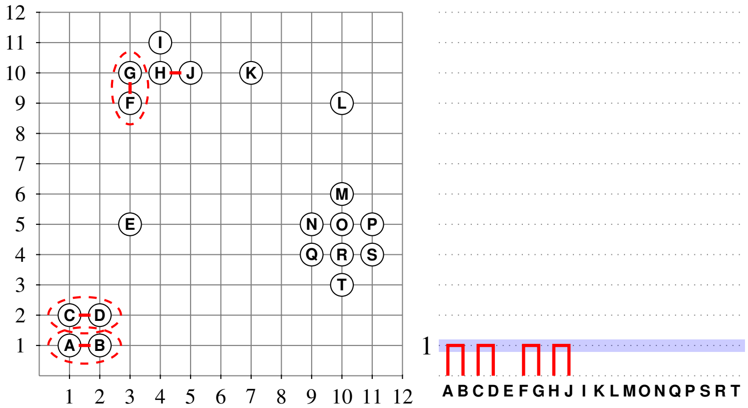
Example Complete Link Clustering

Start example
Skip example



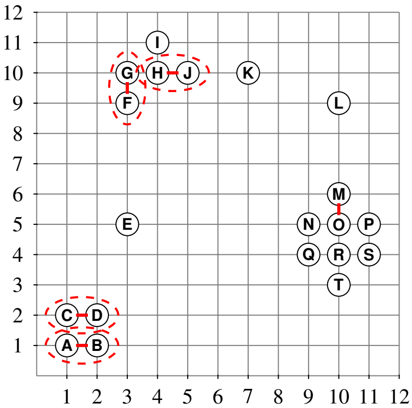
Example Complete Link Clustering

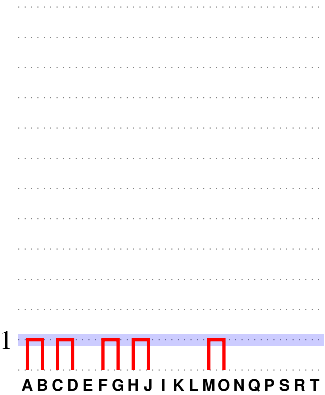
Start example Skip example



Example Complete Link Clustering

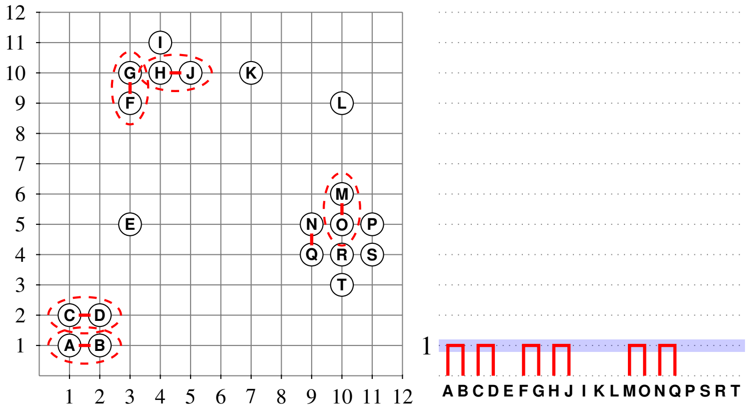
Start example | Skip example





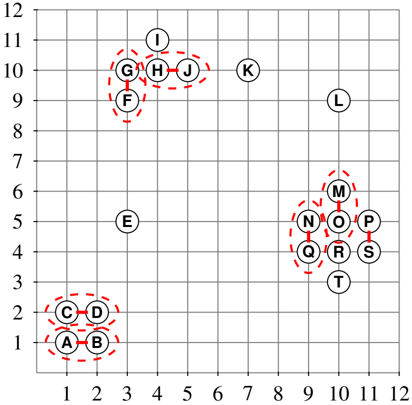
Example Complete Link Clustering

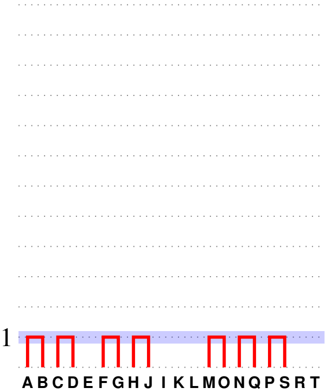
Start example
Skip example



Example Complete Link Clustering

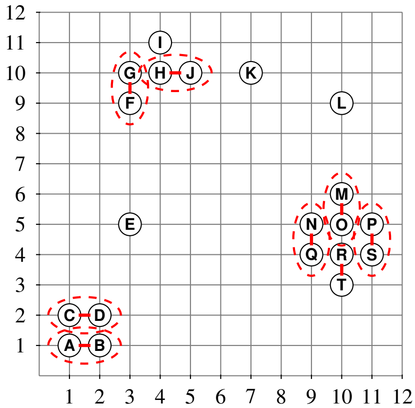
Start example Skip example

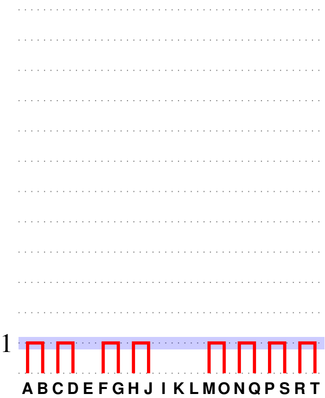




Example Complete Link Clustering

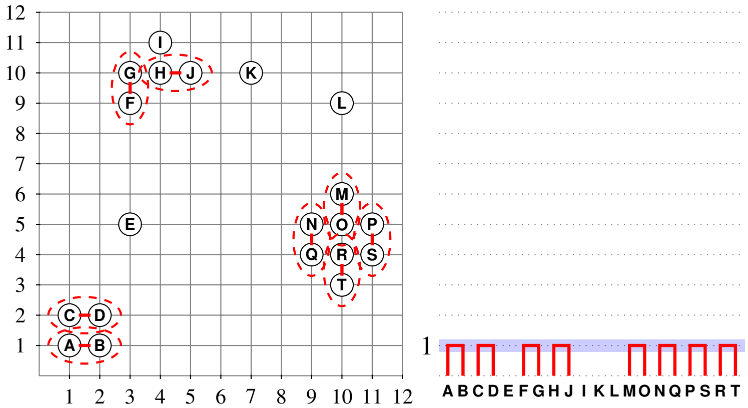
Start example
Skip example





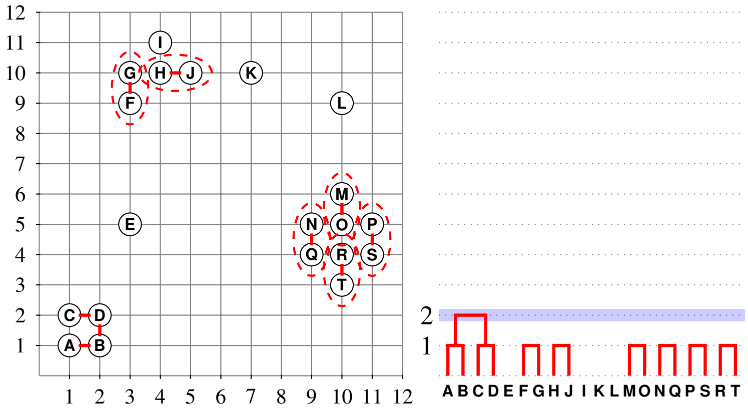
Example Complete Link Clustering

Start example Skip example



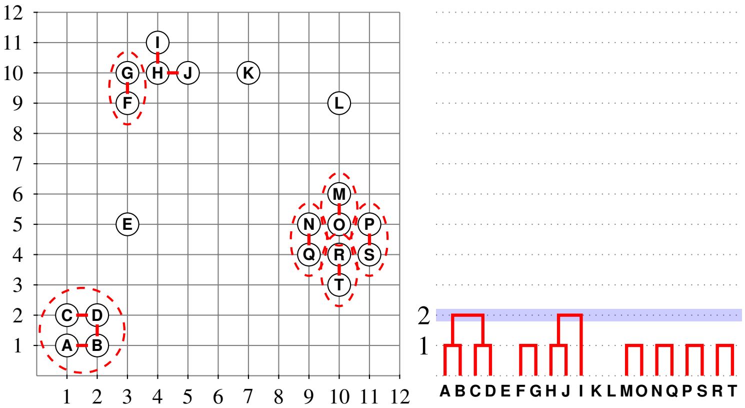
Example Complete Link Clustering

Start example Skip example



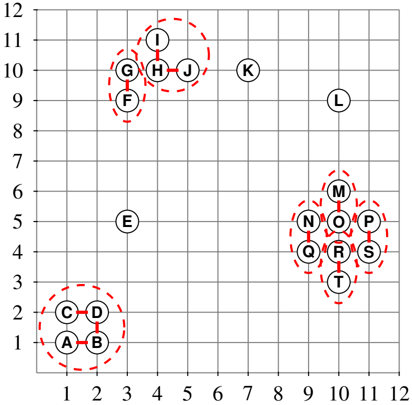
Example Complete Link Clustering

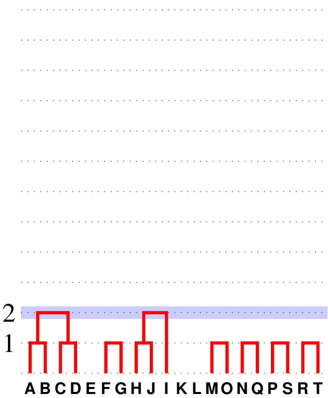
Start example   Skip example



Example Complete Link Clustering

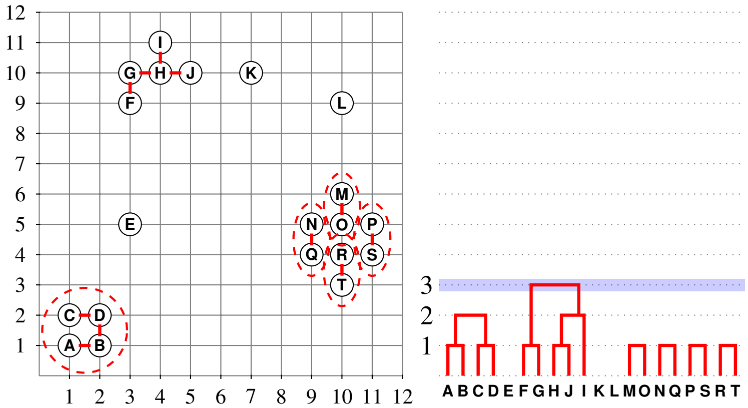
Start example · Skip example





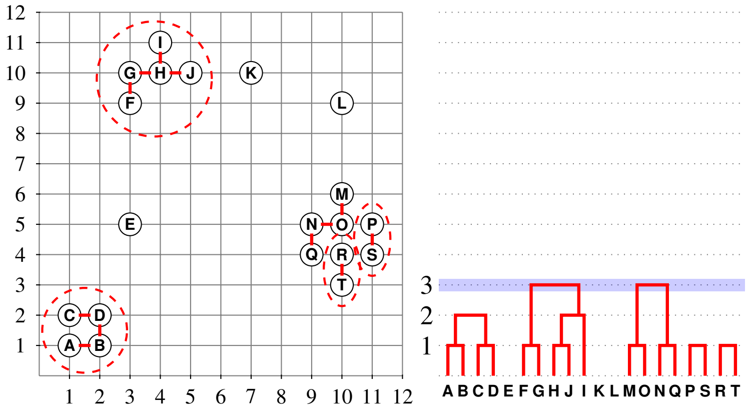
Example Complete Link Clustering

Start example Skip example



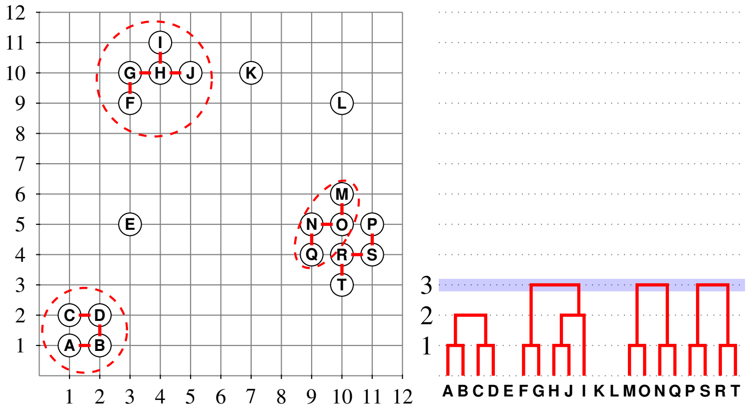
Example Complete Link Clustering

Start example
Skip example



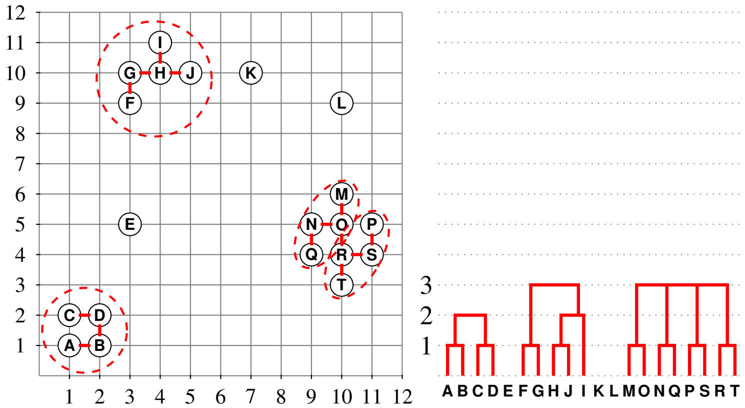
Example Complete Link Clustering

Start example | Skip example



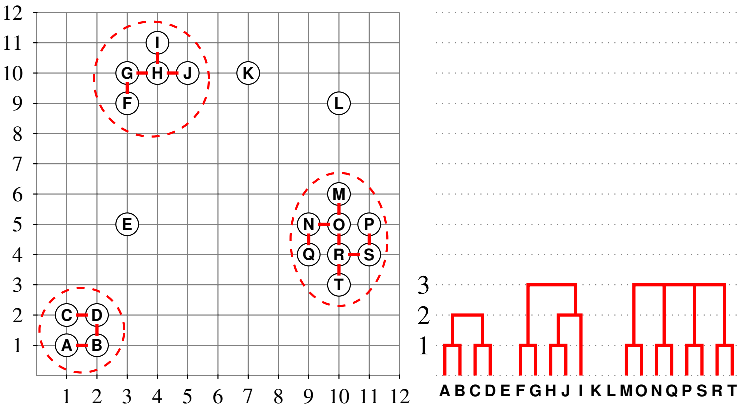
Example Complete Link Clustering

Start example   Skip example



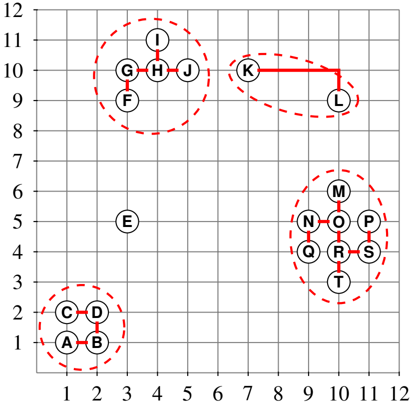
Example Complete Link Clustering

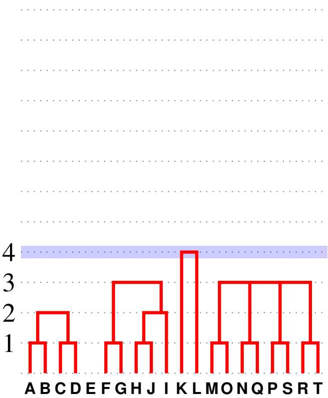
Start example Skip example



Example Complete Link Clustering

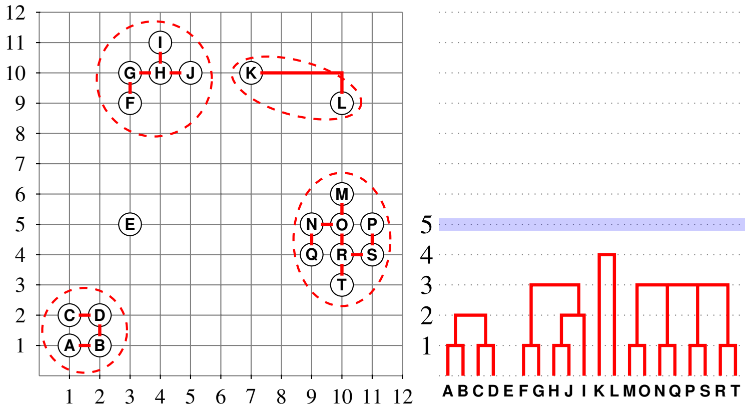
Start example Skip example





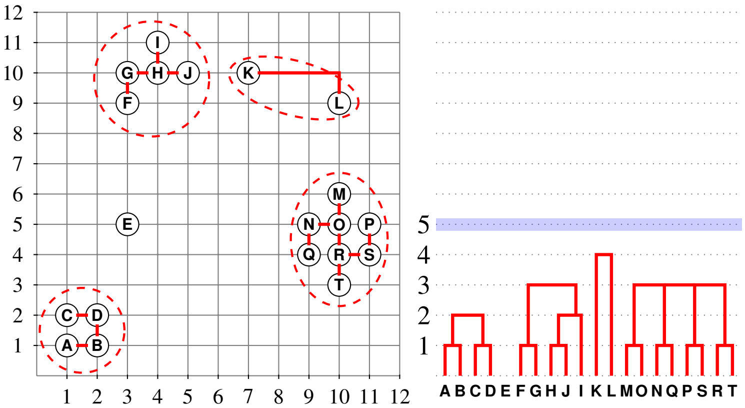
Example Complete Link Clustering

Start example Skip example



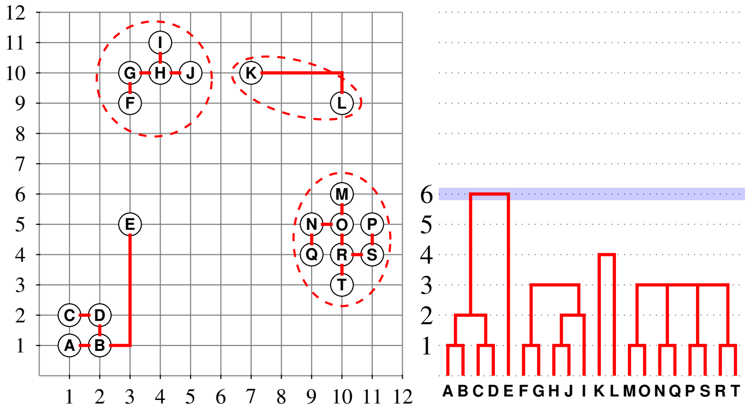
Example Complete Link Clustering

Start example   Skip example



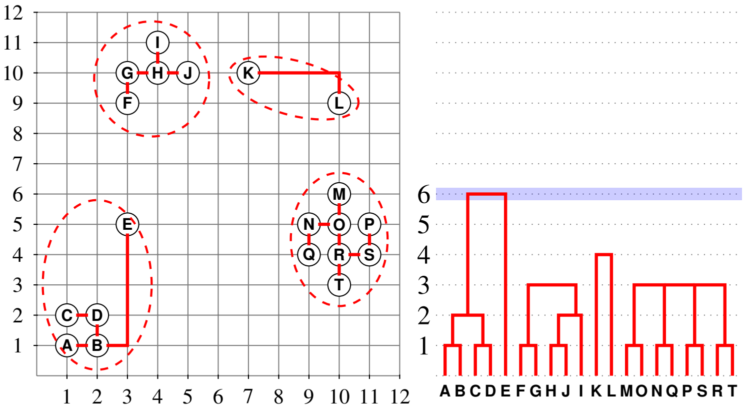
Example Complete Link Clustering

Start example Skip example



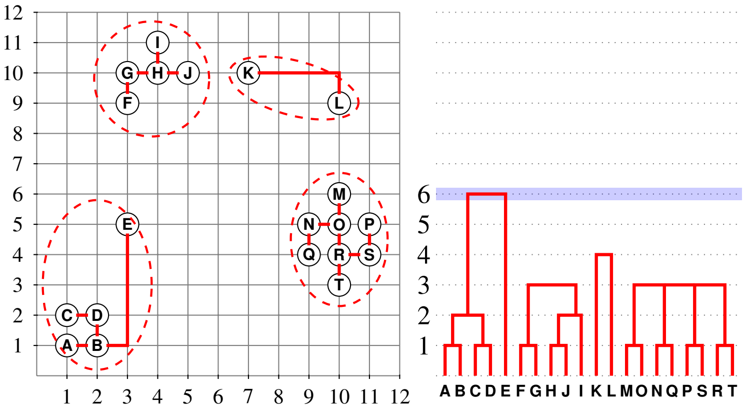
Example Complete Link Clustering

Start example
Skip example



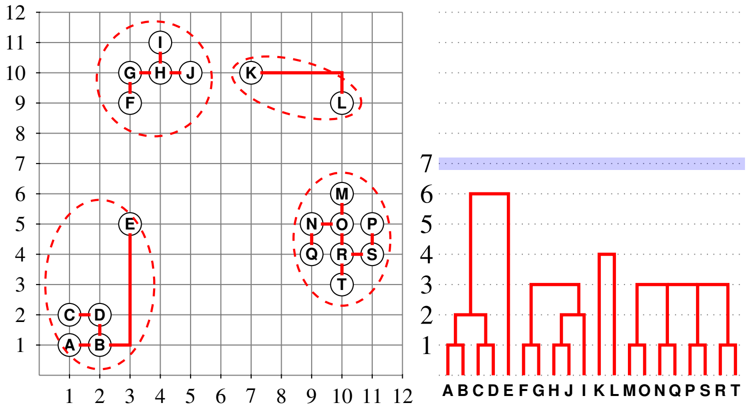
Example Complete Link Clustering

Start example Skip example



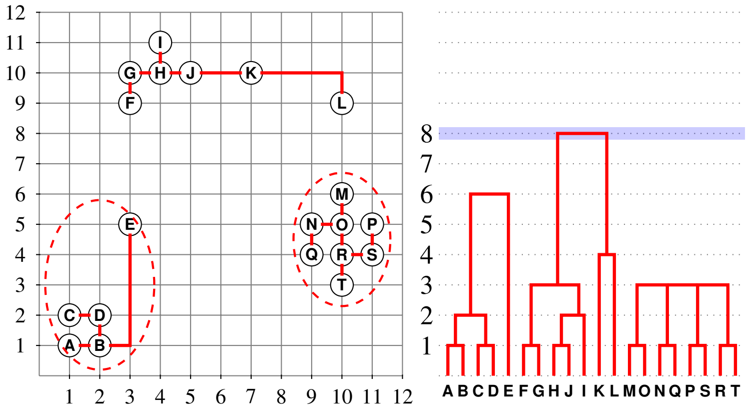
Example Complete Link Clustering

Start example   Skip example



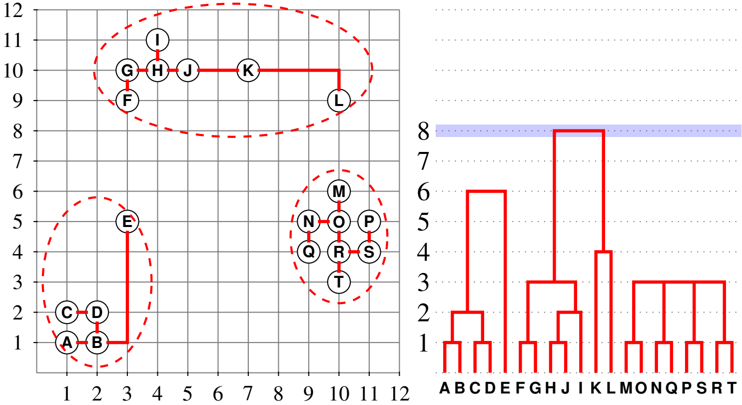
Example Complete Link Clustering

Start example | Skip example



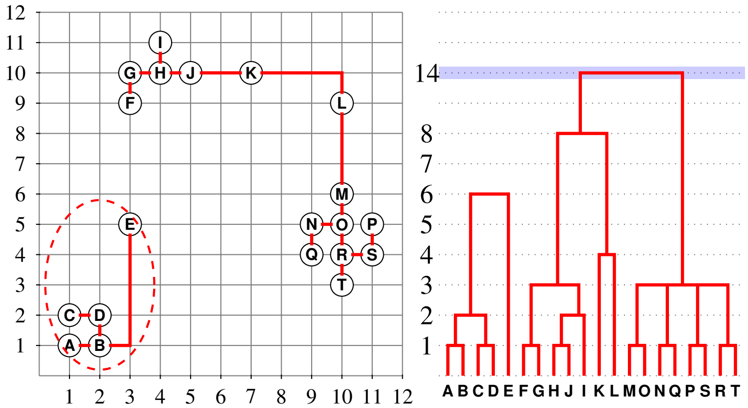
Example Complete Link Clustering

▸ Start example ▸▸ Skip example



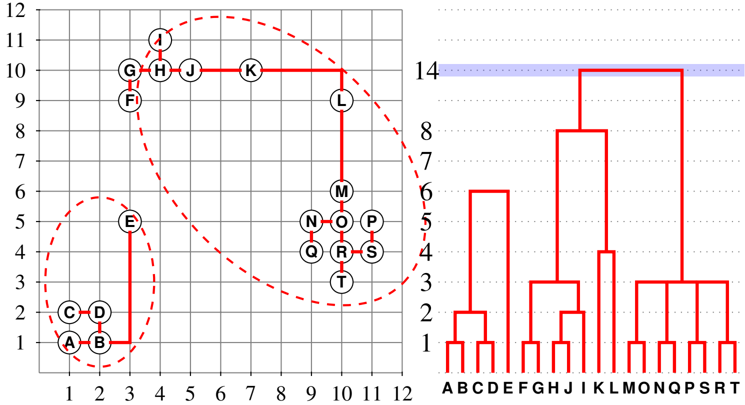
Example Complete Link Clustering

Start example
Skip example



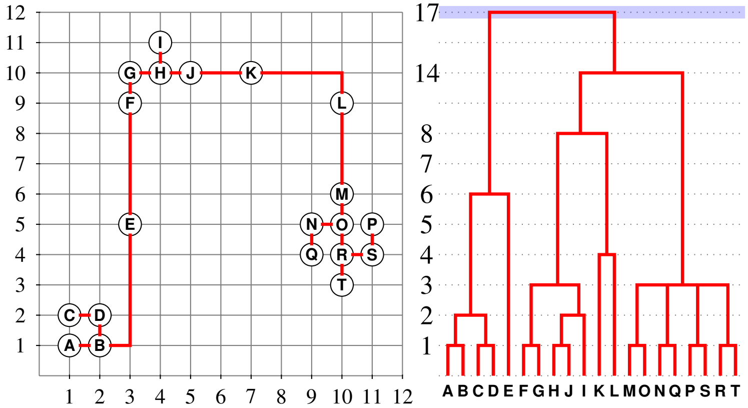
Example Complete Link Clustering

Start example Skip example



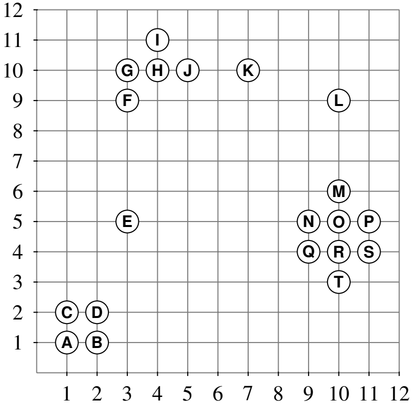
Example Complete Link Clustering

Start example Skip example



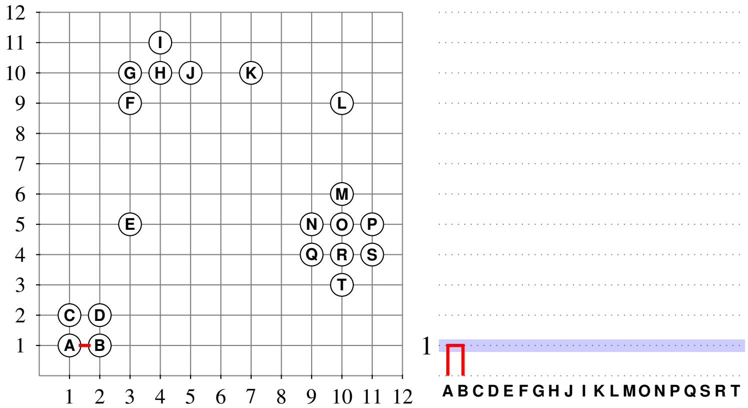
Example Average Link Clustering

Start example
Skip example



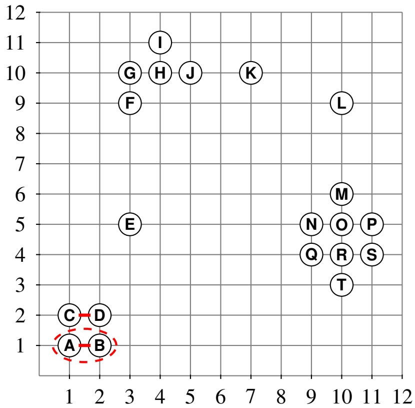
Example Average Link Clustering

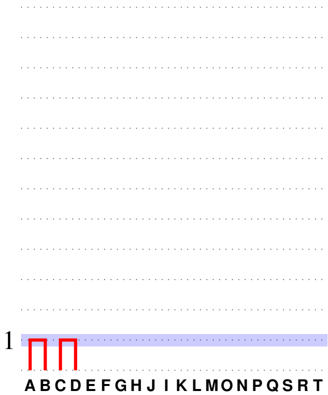
Start example | Skip example



Example Average Link Clustering

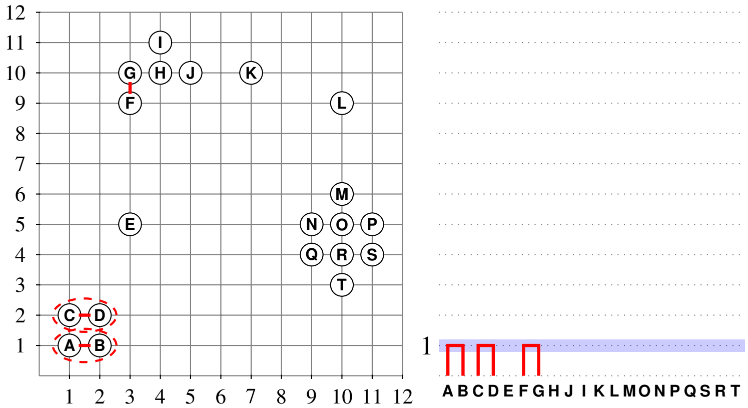
Start example Skip example





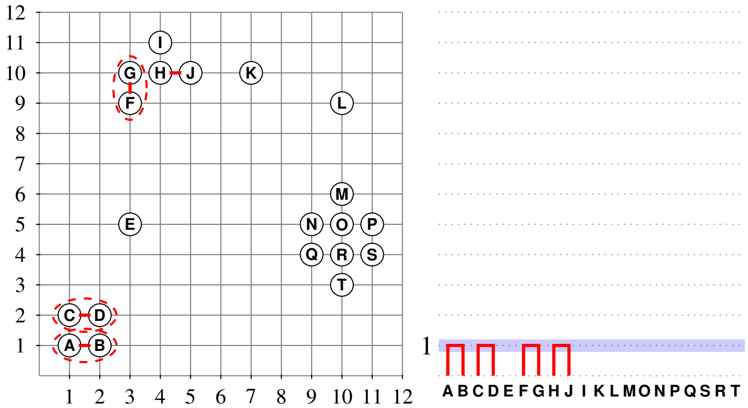
Example Average Link Clustering

Start example   Skip example



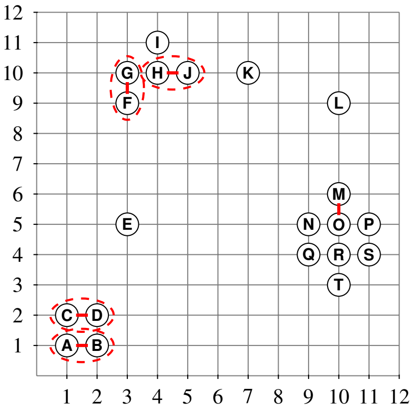
Example Average Link Clustering

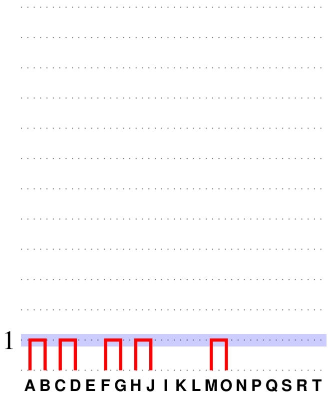
▸ Start example ⏩ Skip example



Example Average Link Clustering

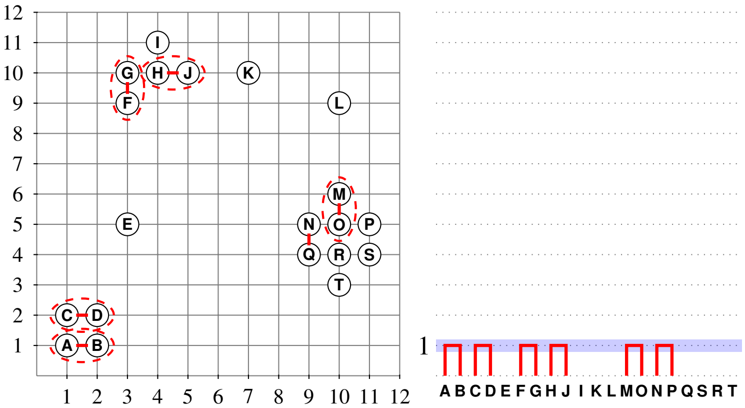
Start example
Skip example





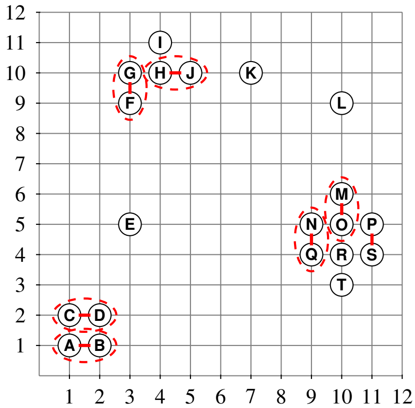
Example Average Link Clustering

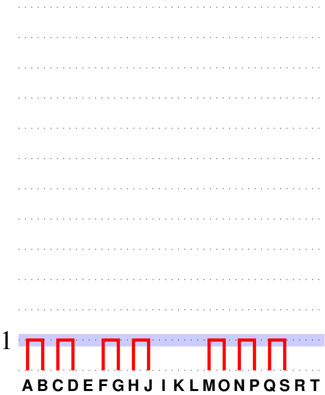
Start example Skip example



Example Average Link Clustering

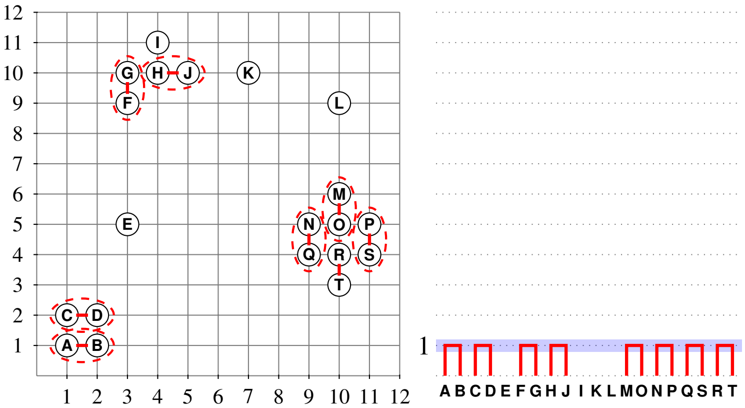
Start example Skip example





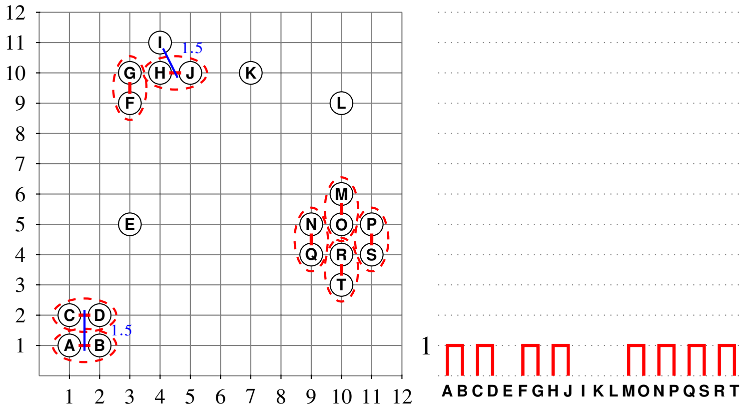
Example Average Link Clustering

Start example Skip example



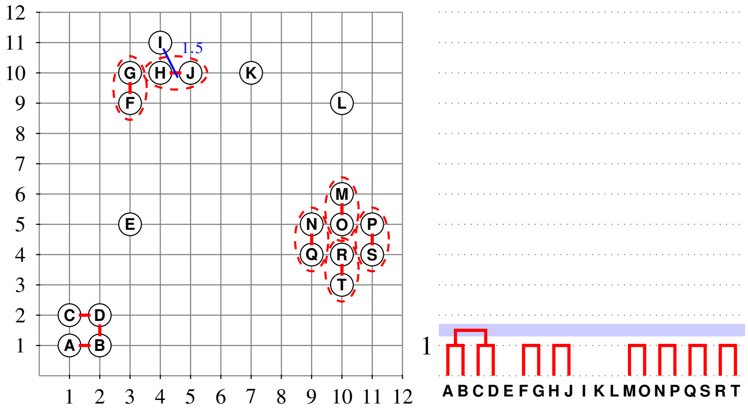
Example Average Link Clustering

Start example   Skip example



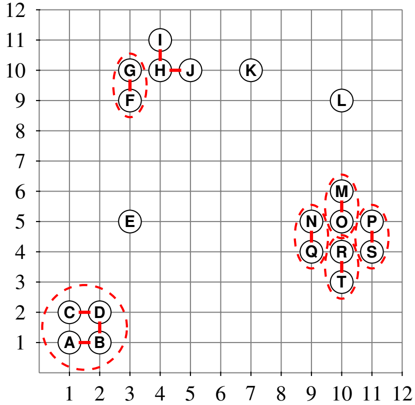
Example Average Link Clustering

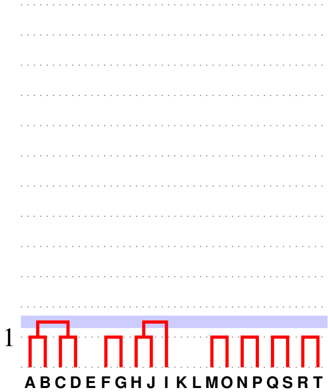
Start example Skip example



Example Average Link Clustering

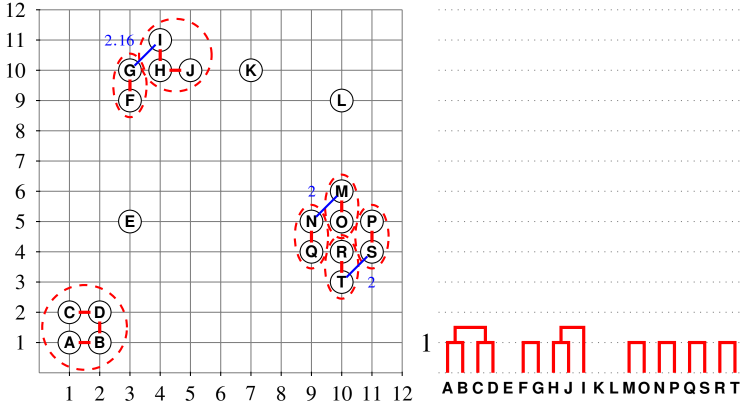
Start example
Skip example





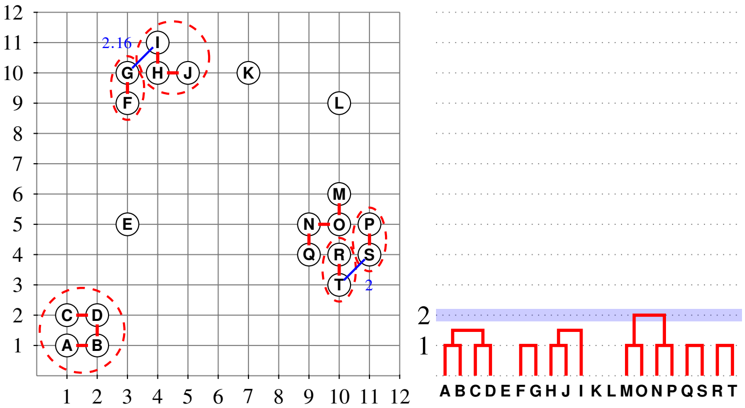
Example Average Link Clustering

Start example   Skip example



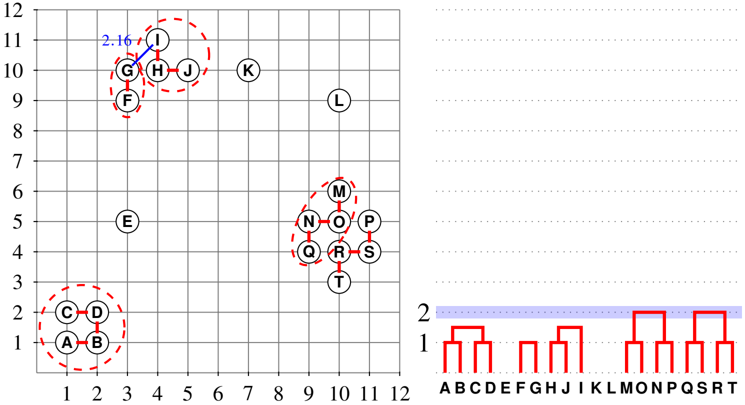
Example Average Link Clustering

Start example Skip example



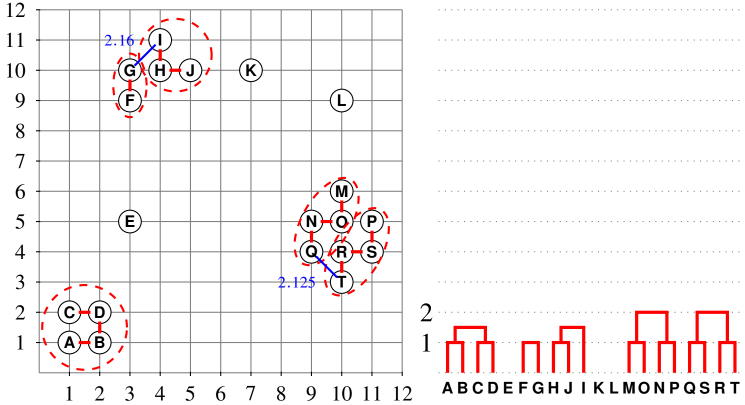
Example Average Link Clustering

Start example | Skip example



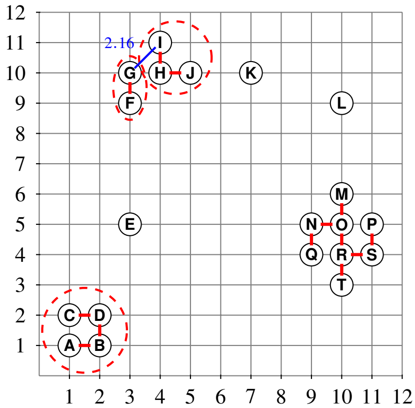
Example Average Link Clustering

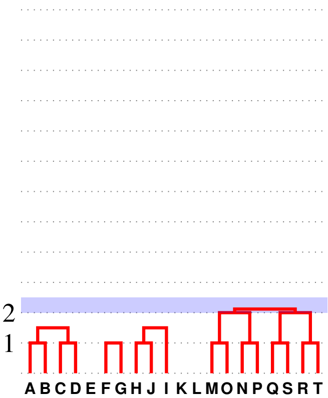
Start example Skip example



Example Average Link Clustering

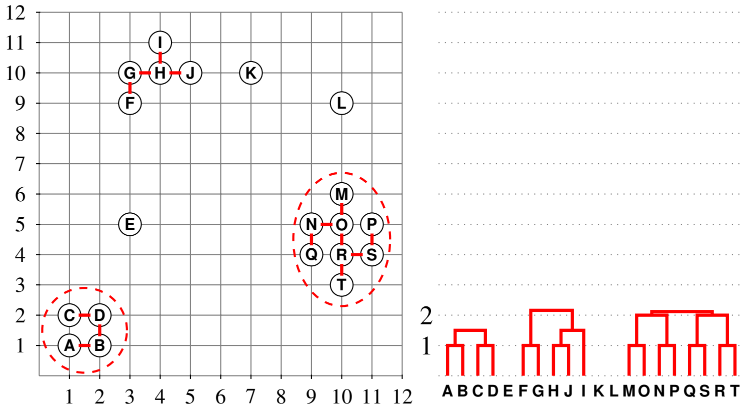
Start example | Skip example





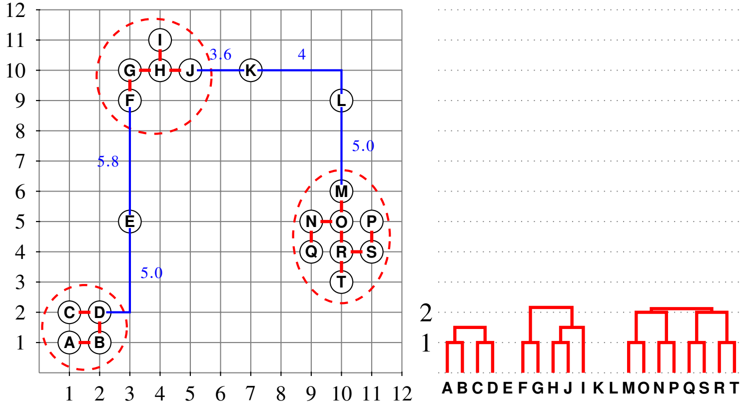
Example Average Link Clustering

▸ Start example ▸▸ Skip example



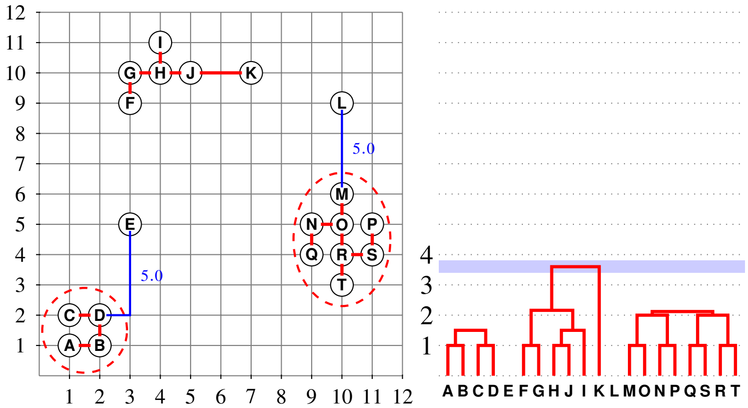
Example Average Link Clustering

Start example Skip example



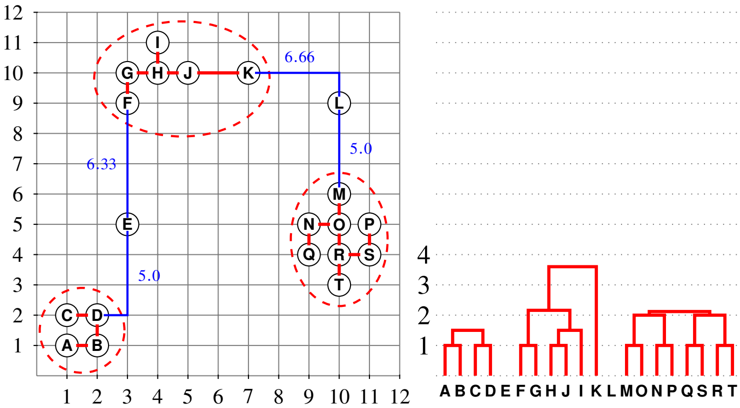
Example Average Link Clustering

▸ Start example ▸▸ Skip example



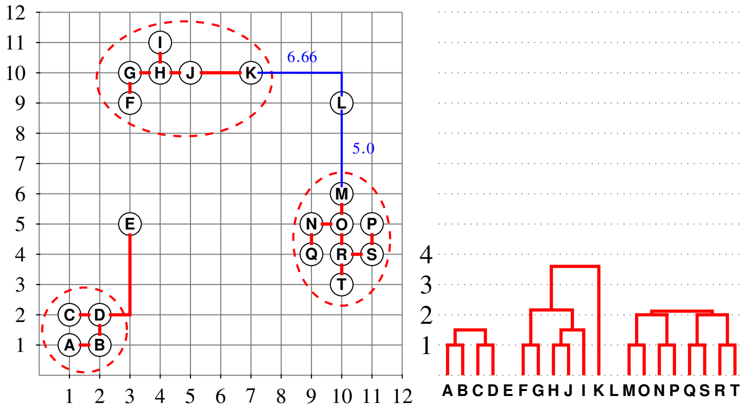
Example Average Link Clustering

Start example | Skip example



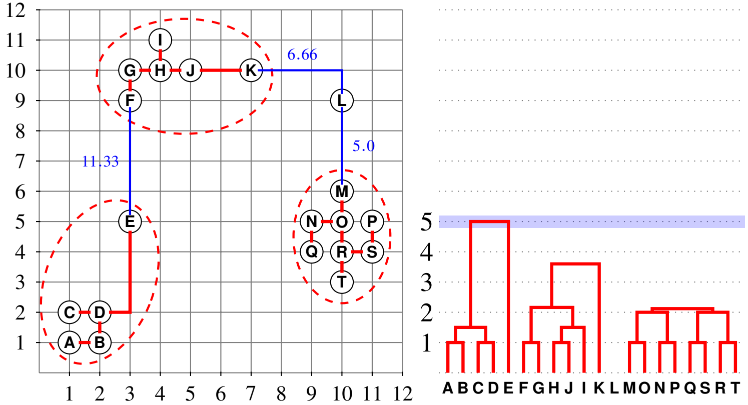
Example Average Link Clustering

Start example | Skip example



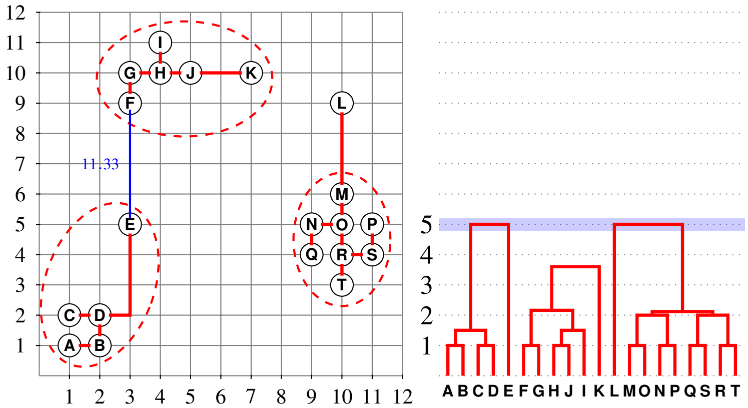
Example Average Link Clustering

Start example | Skip example



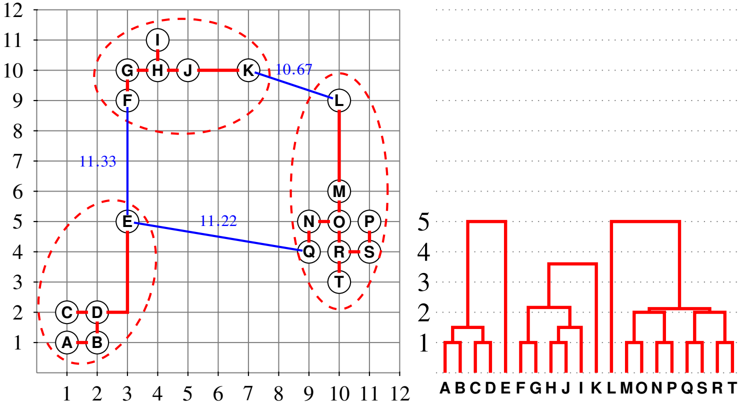
Example Average Link Clustering

Start example · Skip example



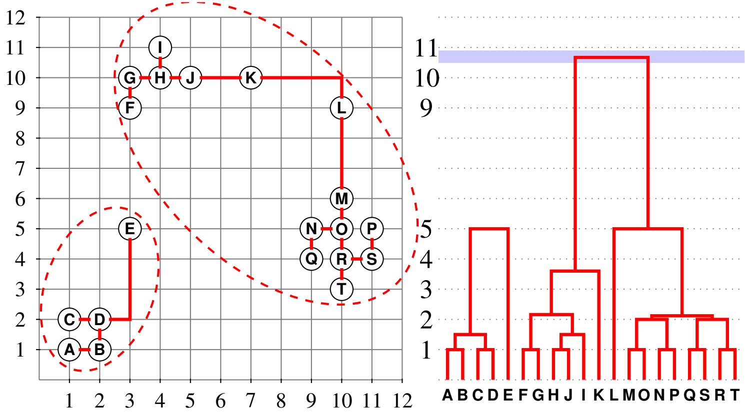
Example Average Link Clustering

Start example
Skip example



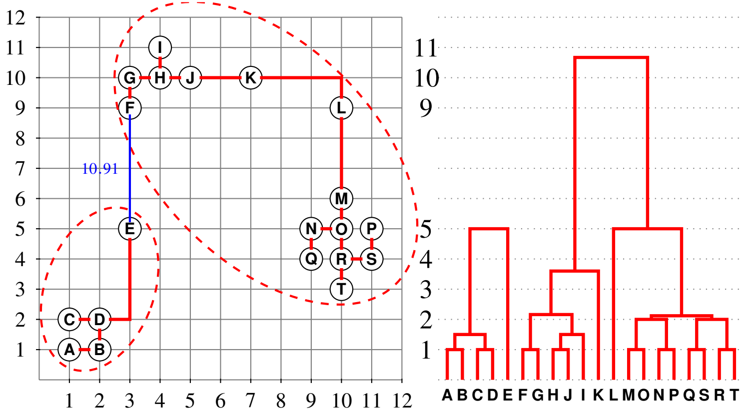
Example Average Link Clustering

Start example | Skip example



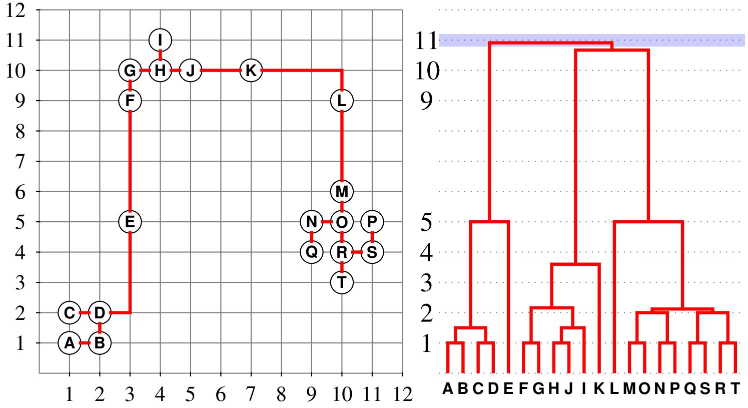
Example Average Link Clustering

Start example
Skip example



Example Average Link Clustering

Start example | Skip example



Discussion

pros

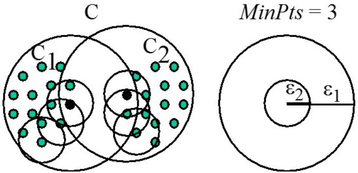
- Does not require knowledge of number of clusters.
- Not only flat partition but a hierarchy of clusters.
- A single partition can be retrieved by some horizontal cut.

cons

- If you want to have a flat partition, where is a good place to cut?
- Greedy heuristic, cannot correct bad decisions.
- single-link-effect, complete-link-effect
- in general: inefficient

Density-based Hierarchical Clustering

- observation: for a constant value of MinPts, density-based clusters with a smaller value of ε are completely contained within clusters with a larger value of ε



- idea: in a DBSCAN-like run compute simultaneously clustering for different density-thresholds:
  - first high density clusters
  - then clusters with lower density
- design a representation different from a dendrogram, to remain readable for large datasets
- cluster model: density-based cluster

OPTICS [Ankerst et al., 1999]

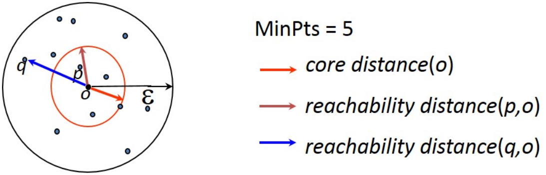
Basic concepts:

- core distance of an object o w.r.t. $\varepsilon, \mathrm{MinPts}$:

$$\mathrm{coredist}_{\varepsilon,\mathrm{MinPts}}(o) = \begin{cases} \text{undefined, if } |\mathrm{RQ}(o,\varepsilon)| < \mathrm{MinPts} \\ \mathrm{dist}_{\mathrm{MinPts}}(o), \text{ else} \end{cases}$$

- reachability distance of object p w.r.t. object o:

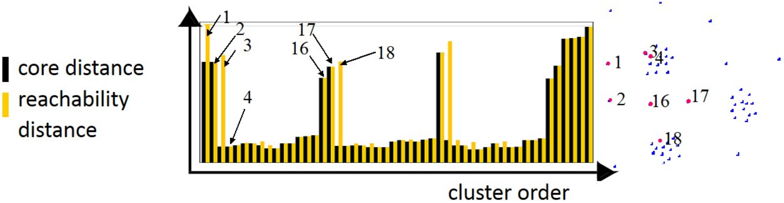
$$\mathrm{reachdist}_{\varepsilon,\mathrm{MinPts}}(p,o) = \begin{cases} \text{undefined, if } |\mathrm{RQ}(o,\varepsilon)| < \mathrm{MinPts} \\ \max\{\mathrm{coredist}(o), \mathrm{dist}(o,p)\}, \text{ else} \end{cases}$$



OPTICS [Ankerst et al., 1999]

Result: cluster order

- OPTICS does not directly deliver a (hierarchical) clustering but a "cluster order" w.r.t. ε and MinPts
- starts with an arbitrary object
- continue with the object exhibiting minimal $\mathrm{reachdist}$ to any of the previously seen objects



OPTICS [Ankerst et al., 1999]

data structures:

- SeedList (keeps visited points sorted w.r.t. current reachability distance)
- ClusterOrder (result, is built successively)

Algorithm 7.6 (OPTICS [Ankerst et al., 1999])

SeedList := $\emptyset$;
WHILE *there are unlabeled objects in* $\mathcal{D}$ **DO**
  IF *SeedList* = $\emptyset$
  THEN
    insert arbitrary object into SeedList with reachdist $= \infty$;
  ELSE
    insert first object ("o") from SeedList
      with current reachdist *into ClusterOrder;*
    label o *as processed;*
  FOR ALL *neighbor* $\in \mathrm{RQ}(o, \varepsilon)$ **DO**
    SeedList.update(neighbor, o*);*

OPTICS [Ankerst et al., 1999]

insert/update object in SeedList:

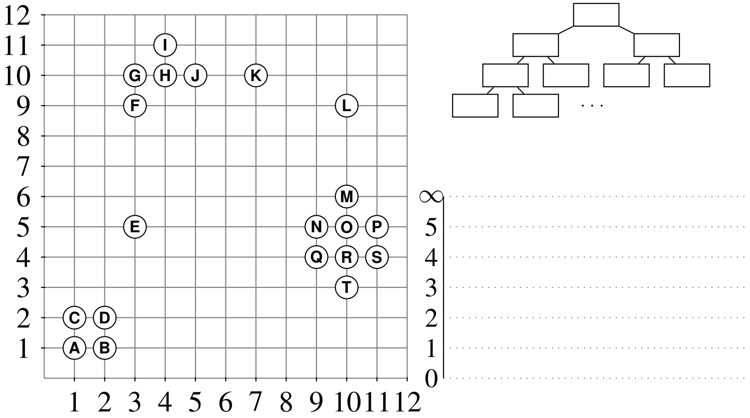
- for each object p in SeedList we keep the current $\mathrm{reachdist}$: p.rdist
- SeedList is organized as a heap (sorted with increasing rdist)

Algorithm 7.7 (OPTICS: update seed list)

```
SeedList :: update(o,obj)
  compute reachdist(o,obj) =: current_rdist_o;
  IF o is already in SeedList THEN
    IF current_rdist_o < o.rdist THEN
      o.rdist := current_rdist_o;
      reorganize heap;
  ELSE
    insert o with o.rdist := current_rdist_o into SeedList;
```

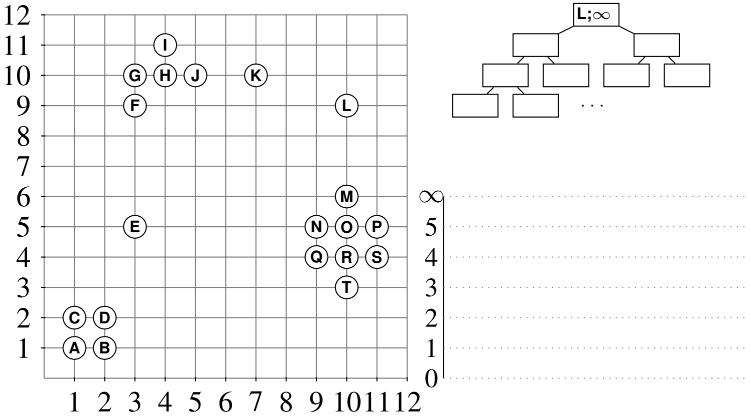
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example



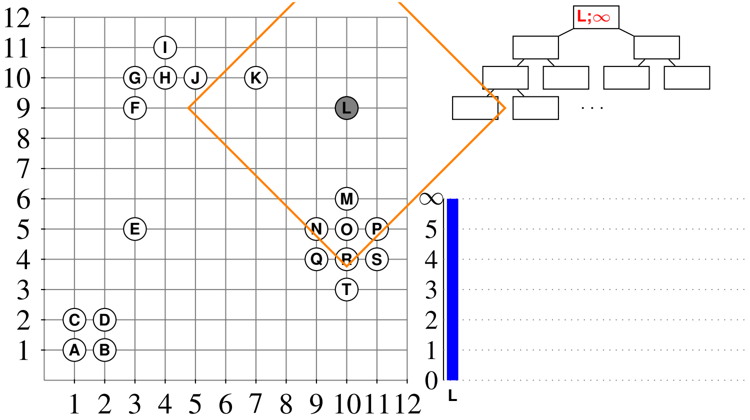
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example



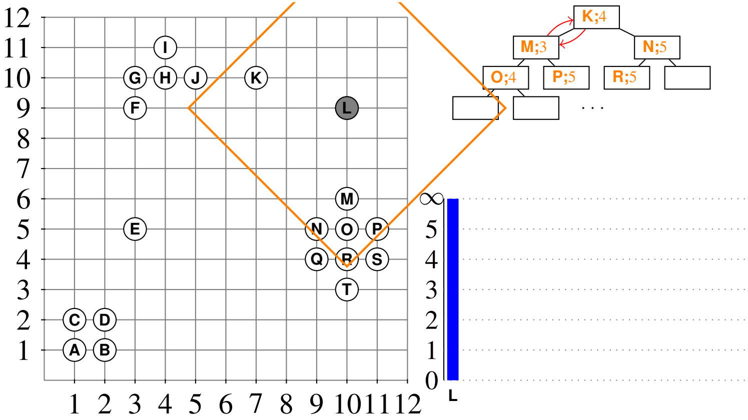
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example



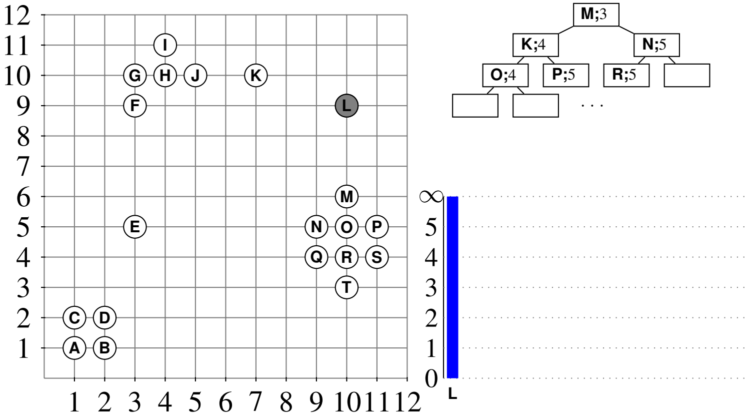
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example



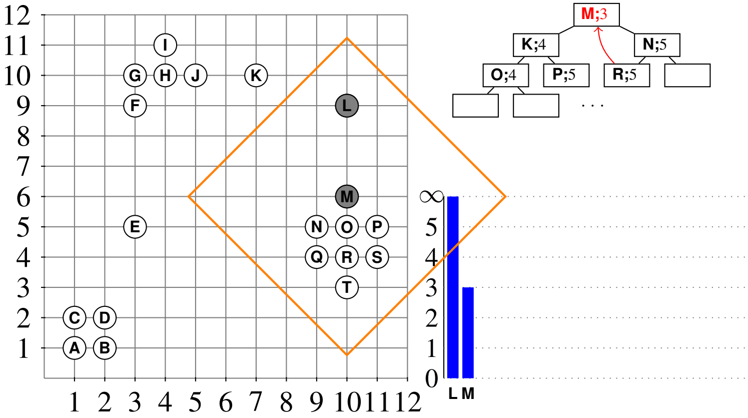
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example · Skip example



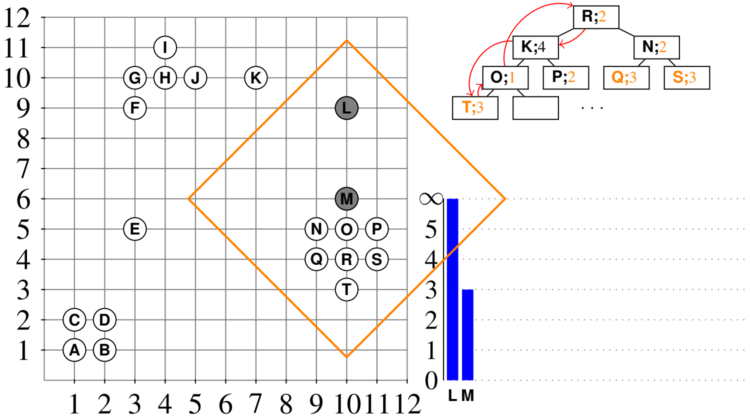
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example



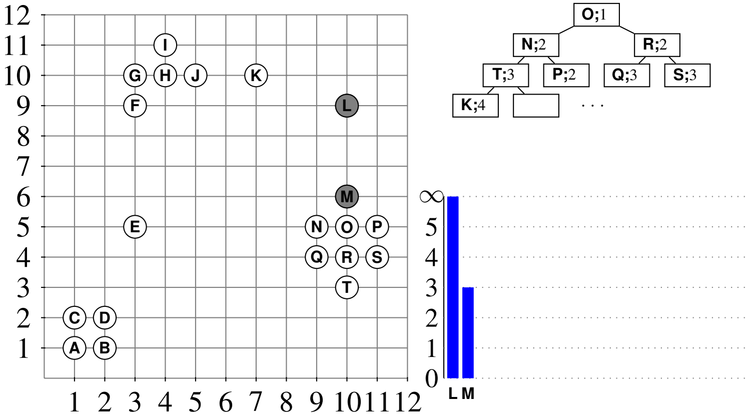
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example | Skip example



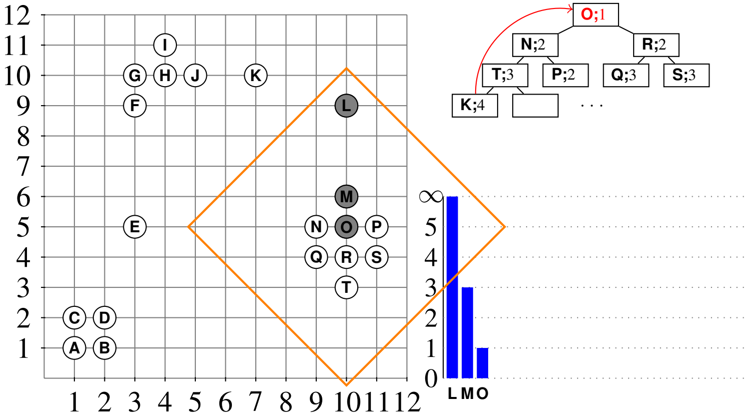
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example | Skip example



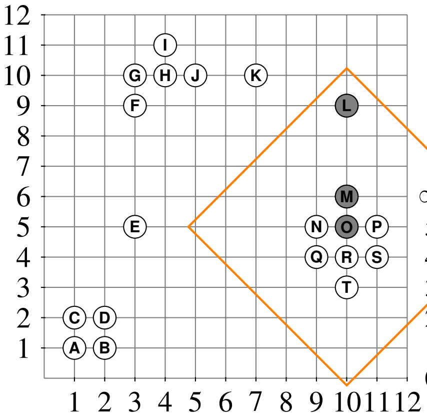
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

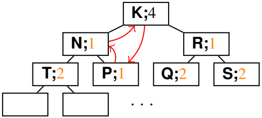
Start example
Skip example

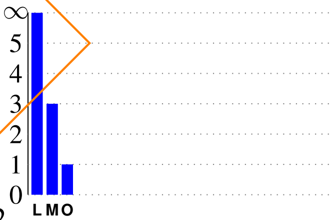


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example · Skip example

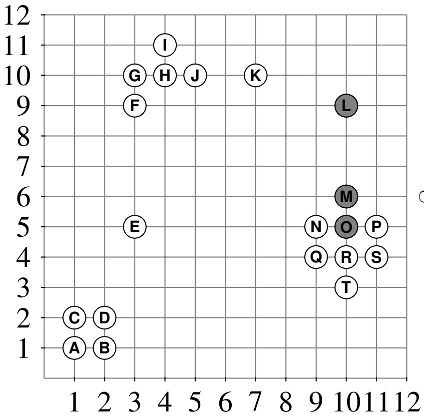


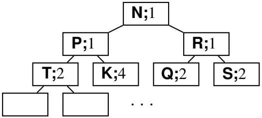


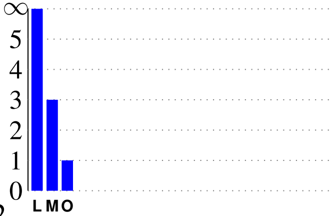


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example

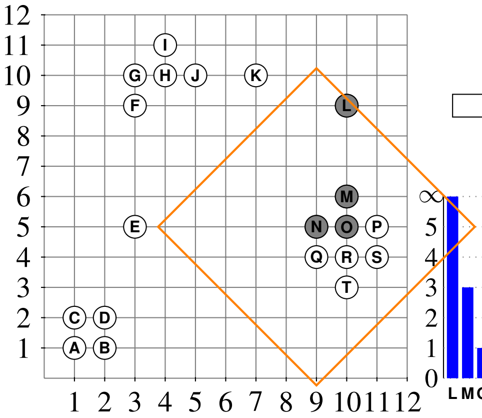


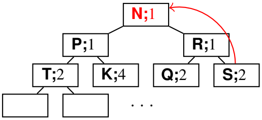


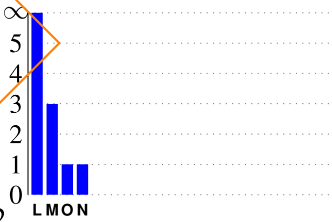


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example

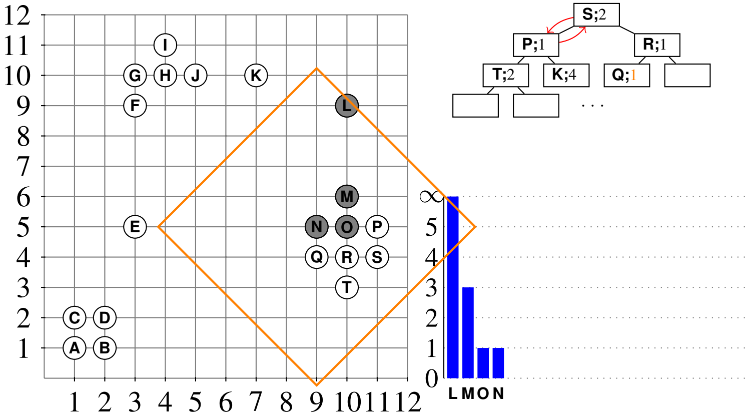






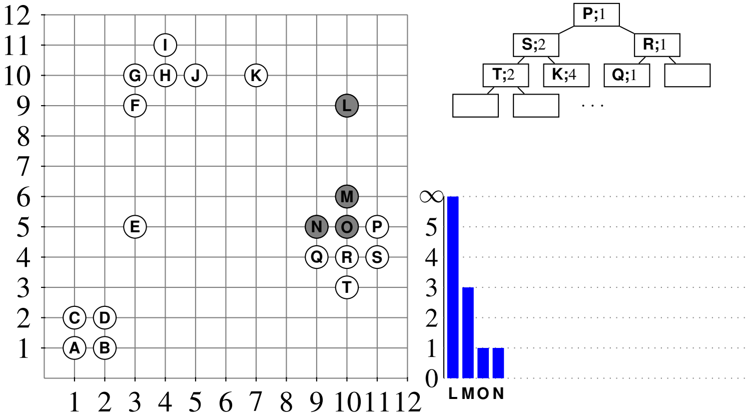
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example



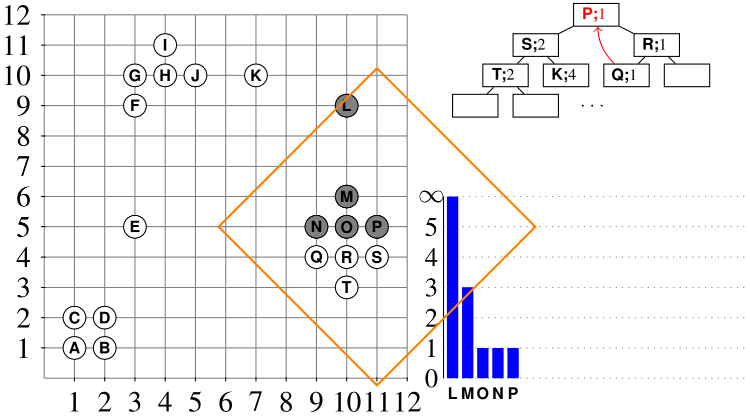
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example



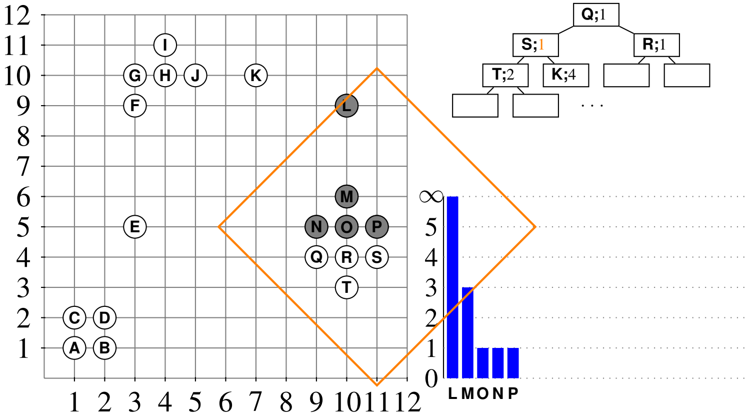
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example



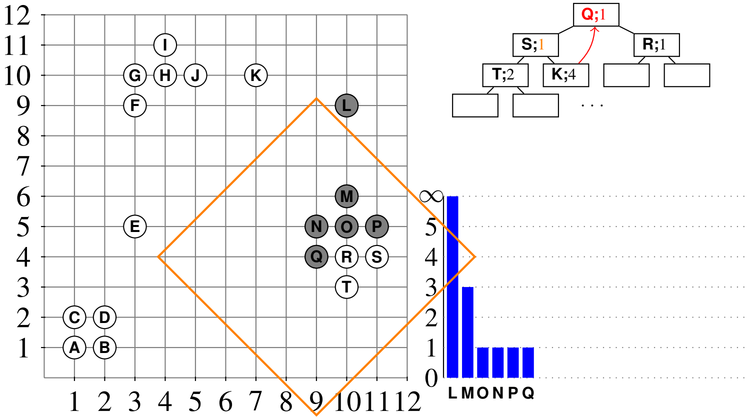
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example | Skip example



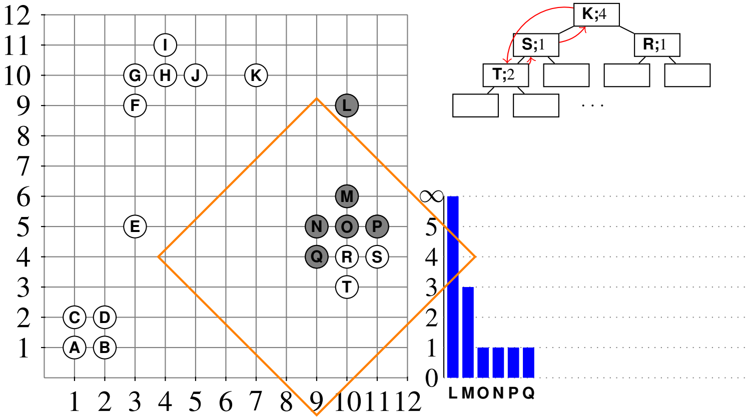
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example



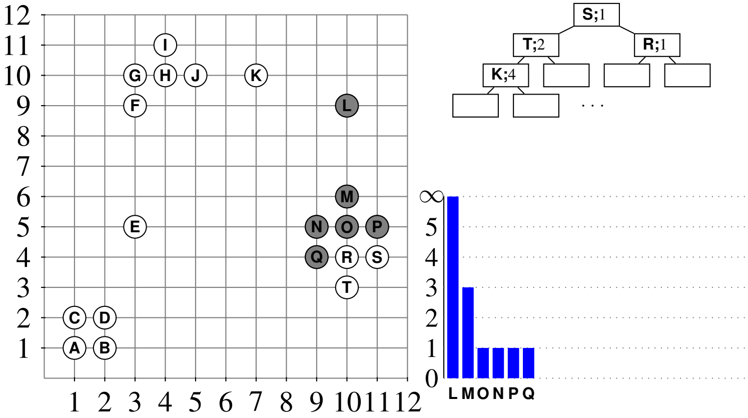
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example



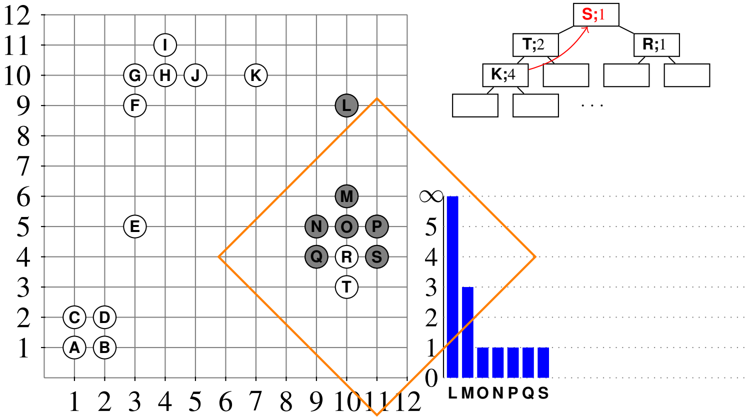
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example



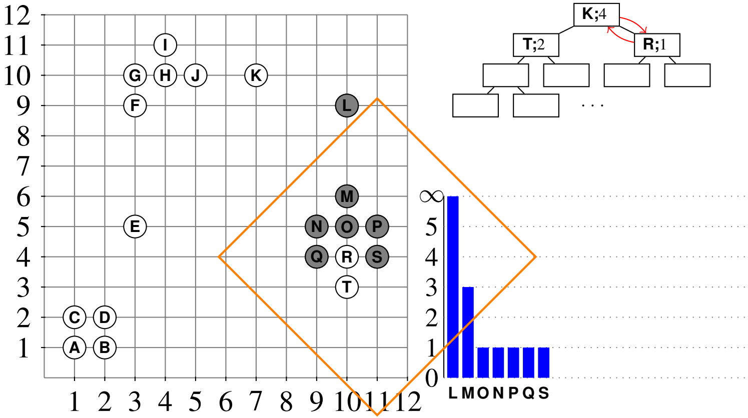
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example



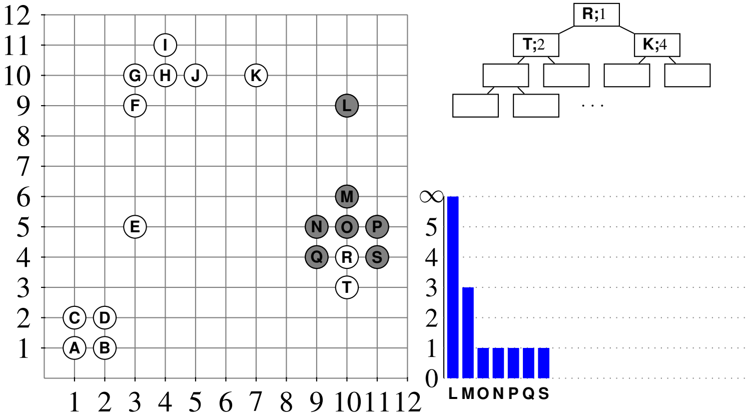
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example



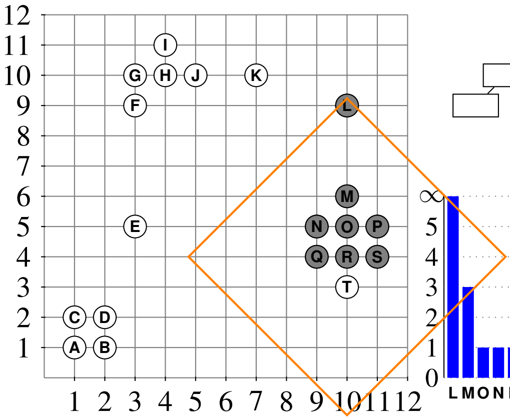
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

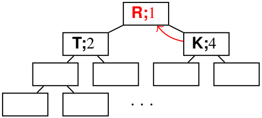
Start example
Skip example

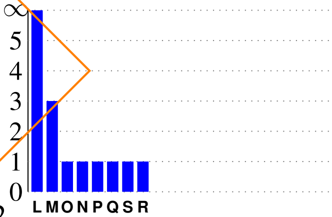


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example

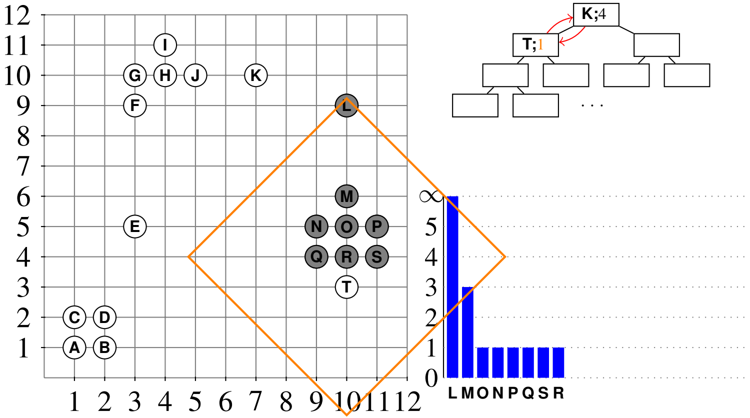






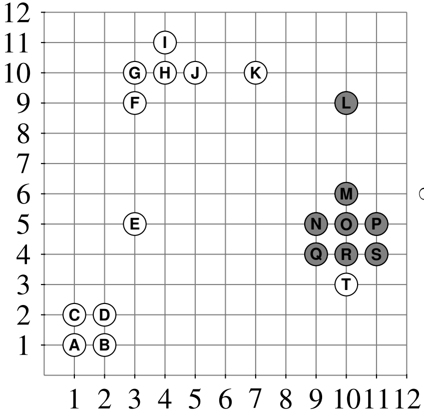
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

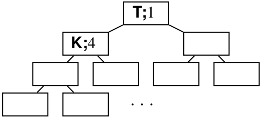
Start example
Skip example

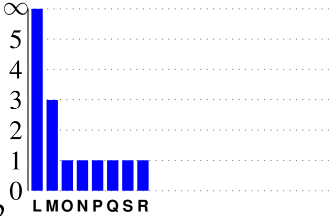


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example

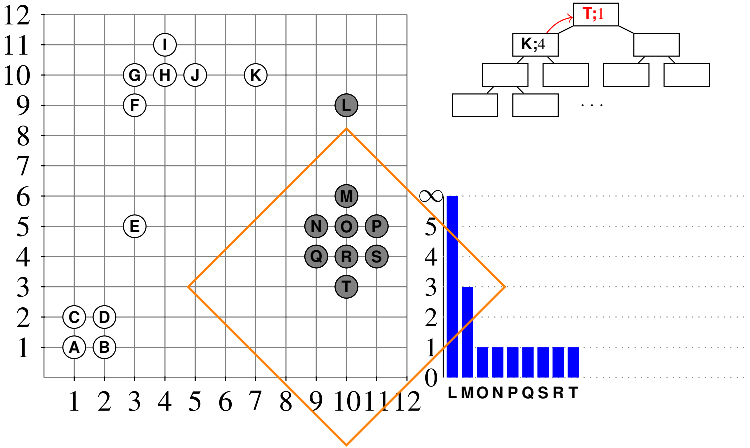






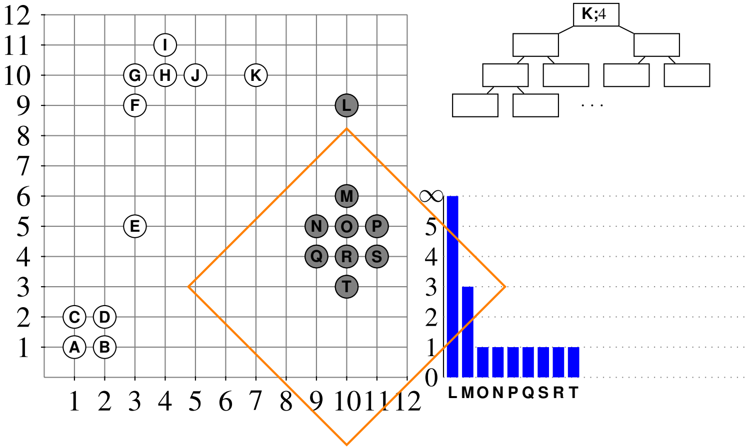
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example



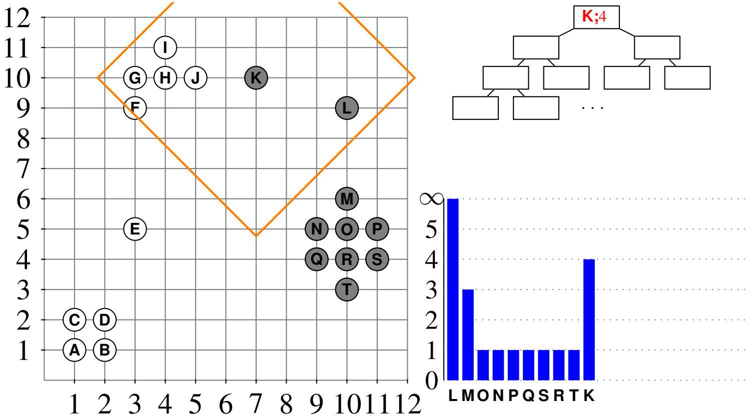
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example



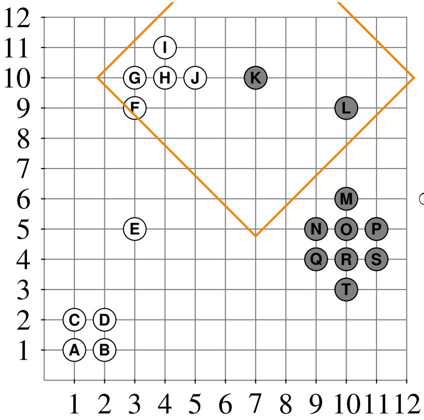
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

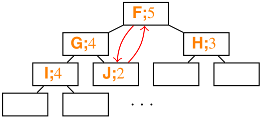
Start example   Skip example

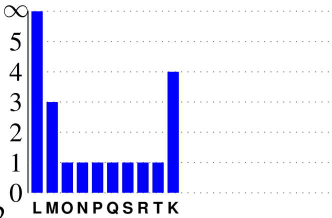


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example

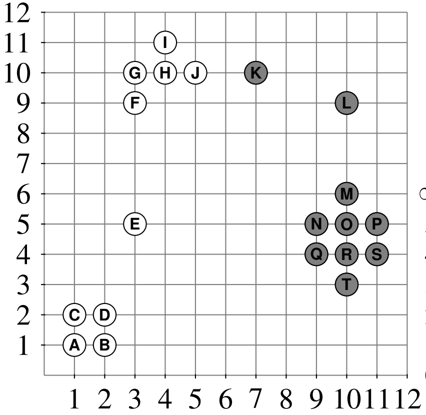


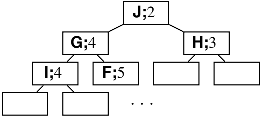


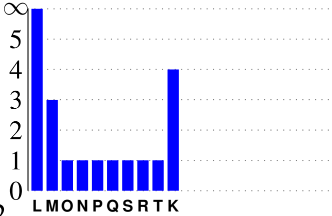


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example

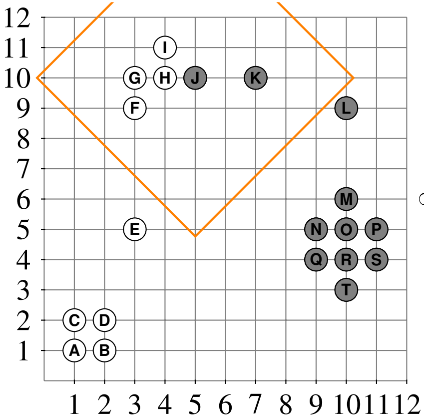


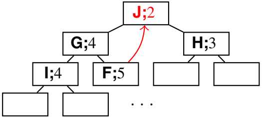


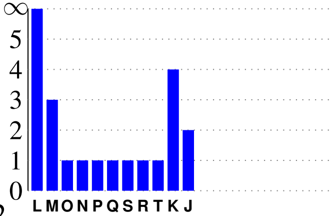


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example

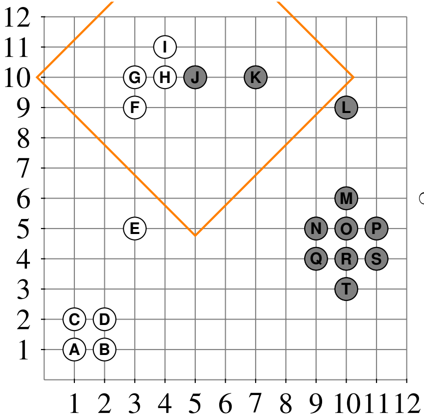


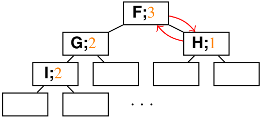


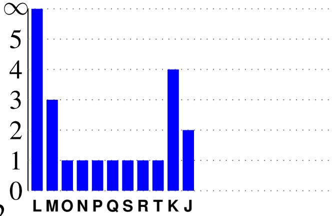


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example

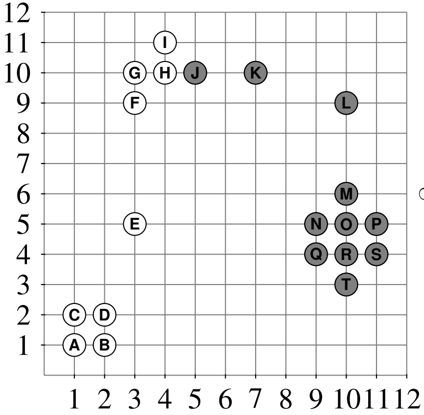


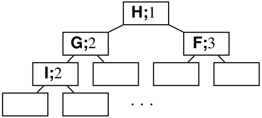


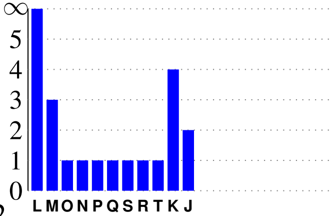


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

▸ Start example ▸▸ Skip example

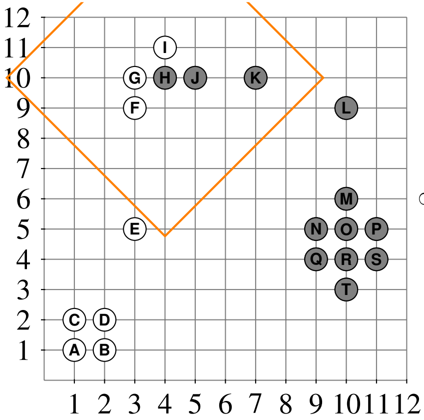


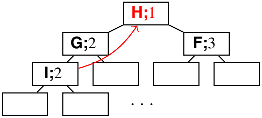


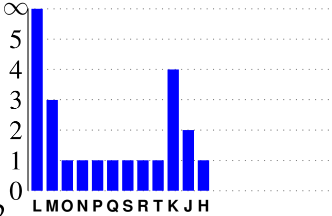


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example

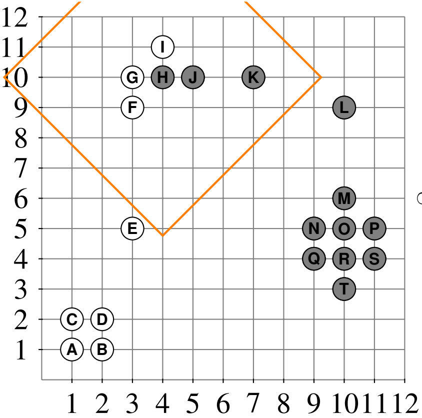


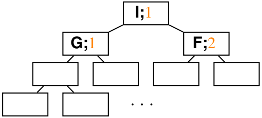


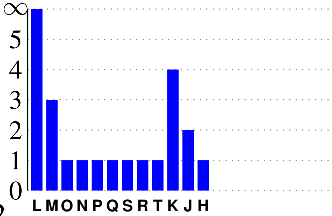


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example

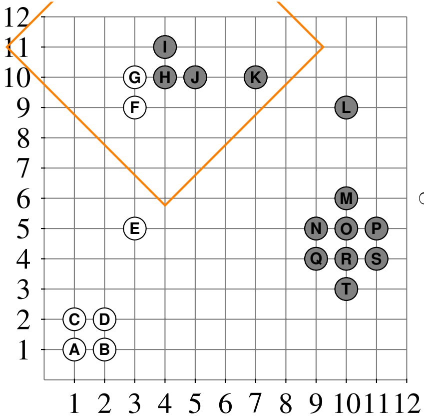


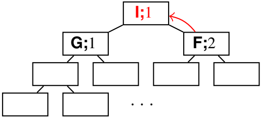


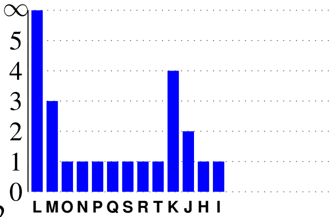


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example

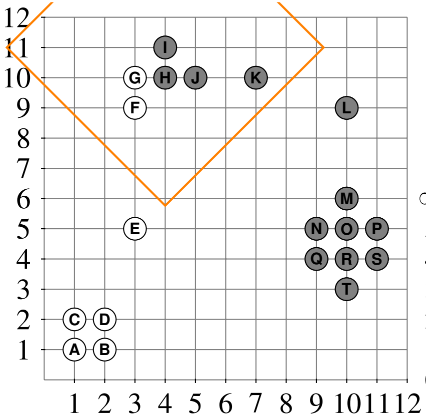


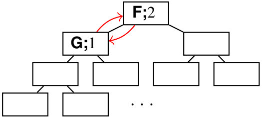


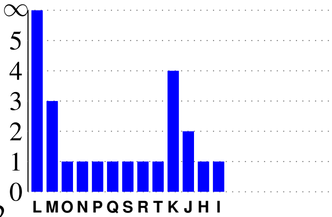


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example | Skip example

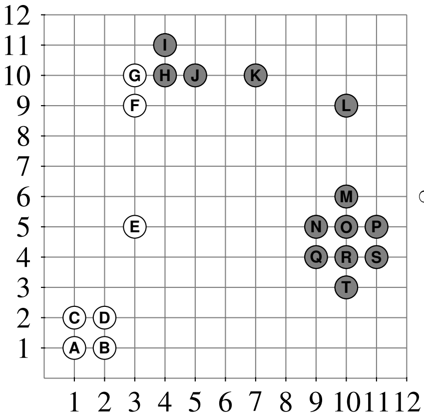


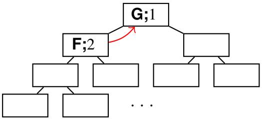


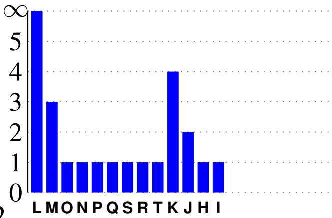


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example

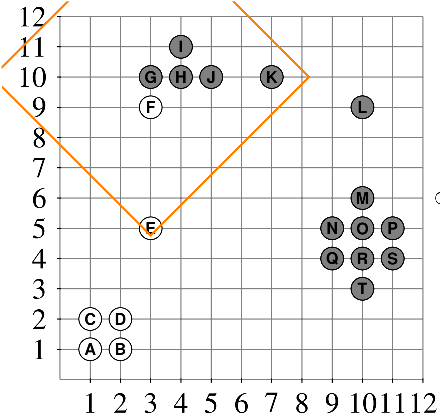


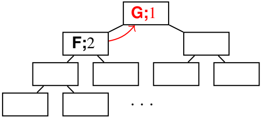


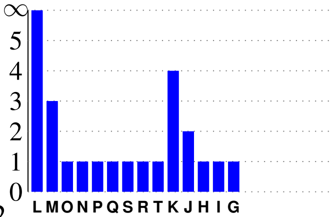


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example

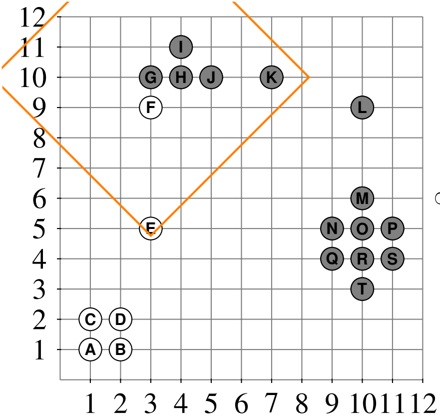


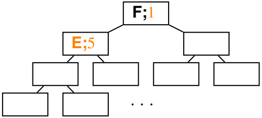


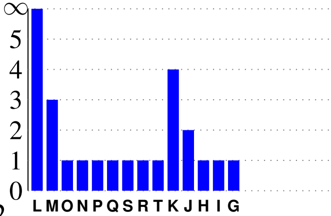


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example

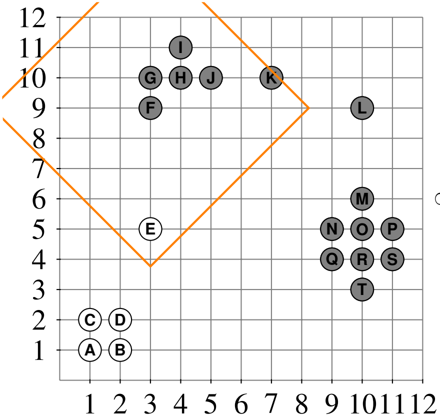


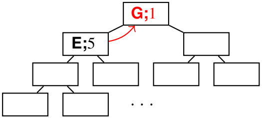


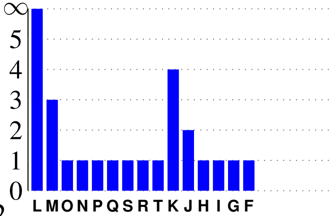


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example

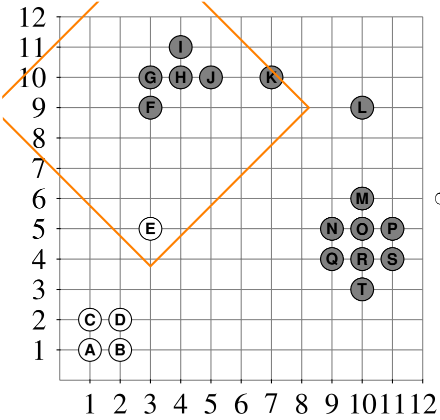


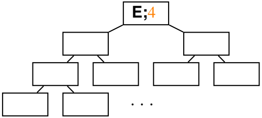


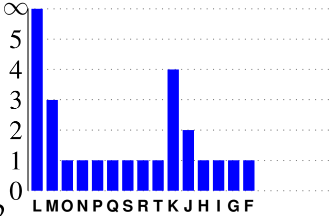


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example

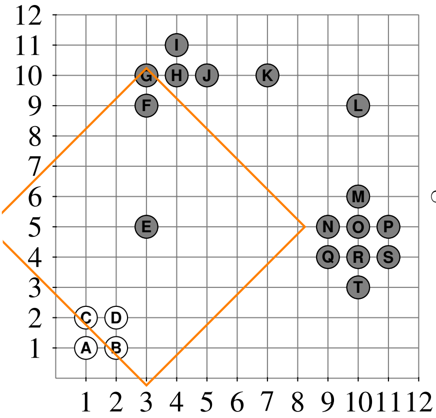


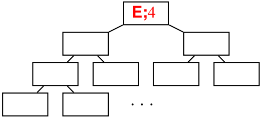


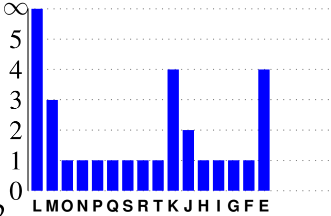


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example

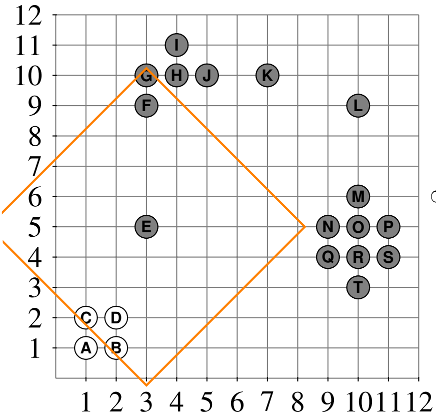


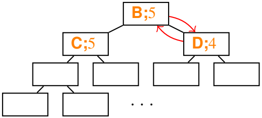


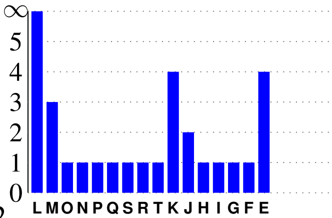


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example

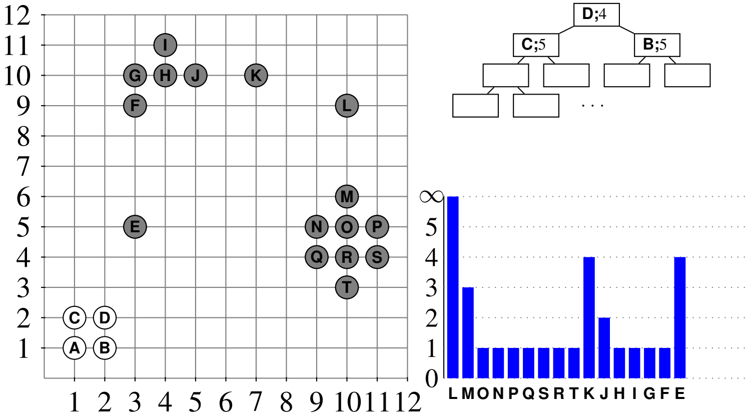






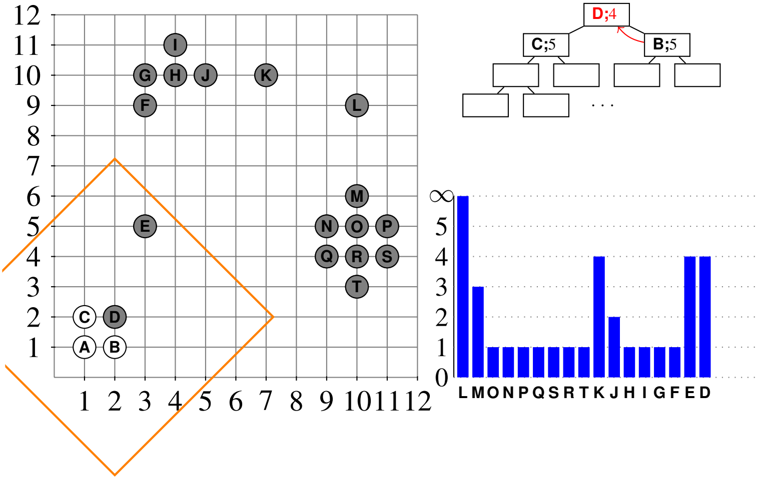
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example



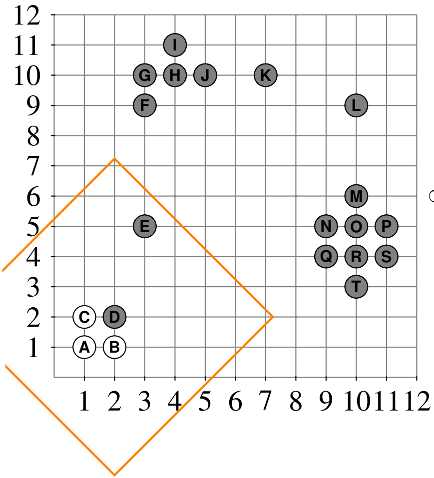
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

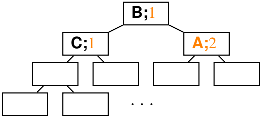
▸ Start example ⏩ Skip example

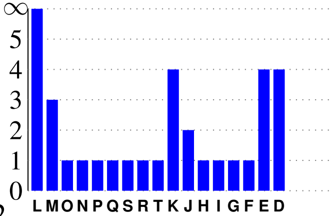


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example

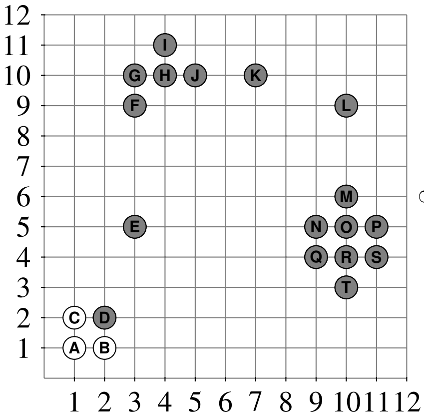


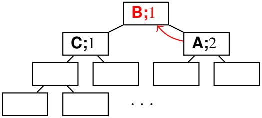


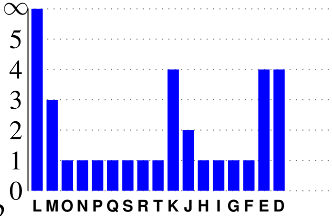


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example Skip example

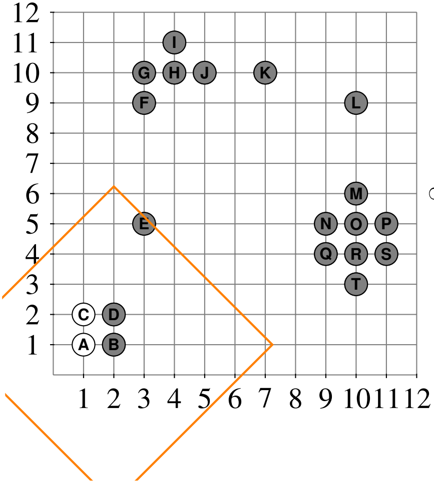


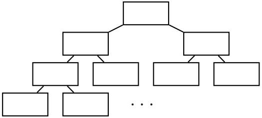


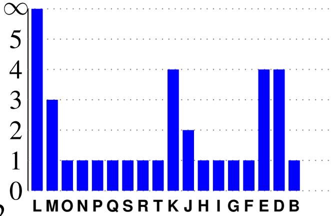


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example

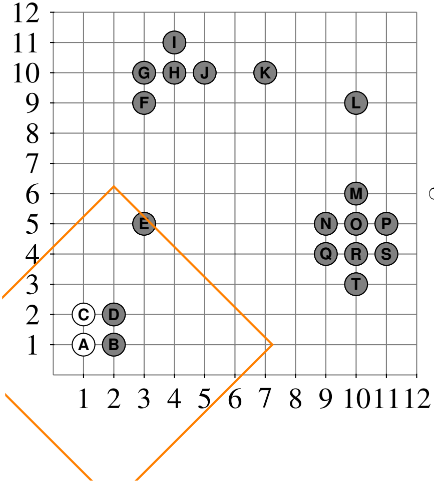


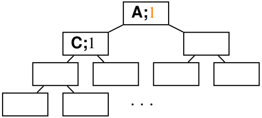


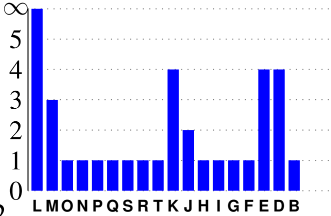


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example

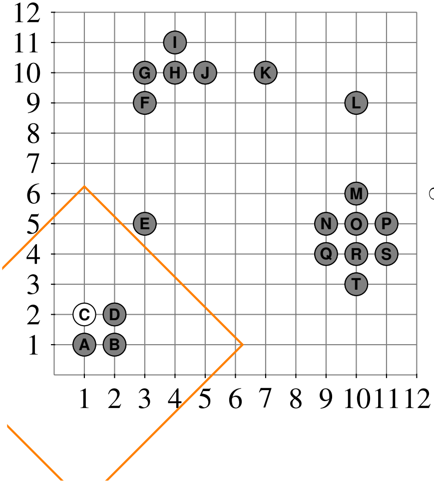


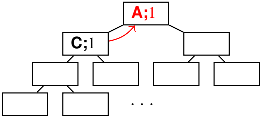


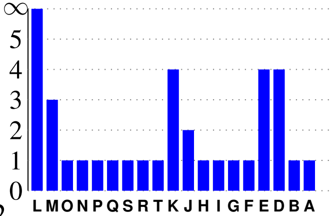


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example

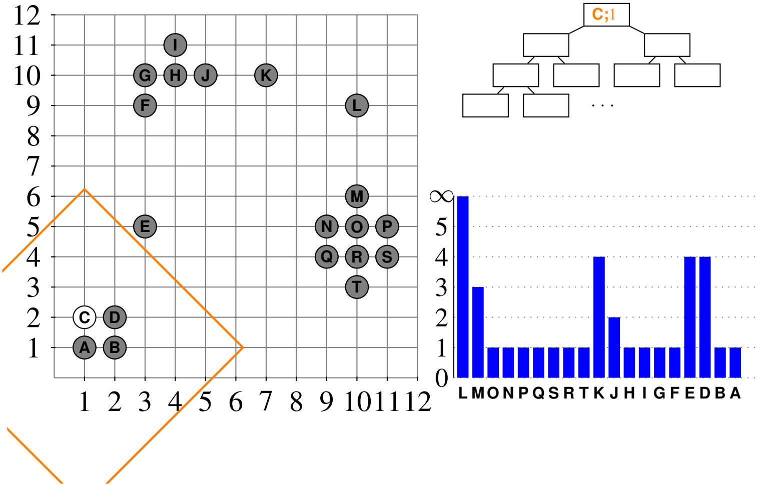






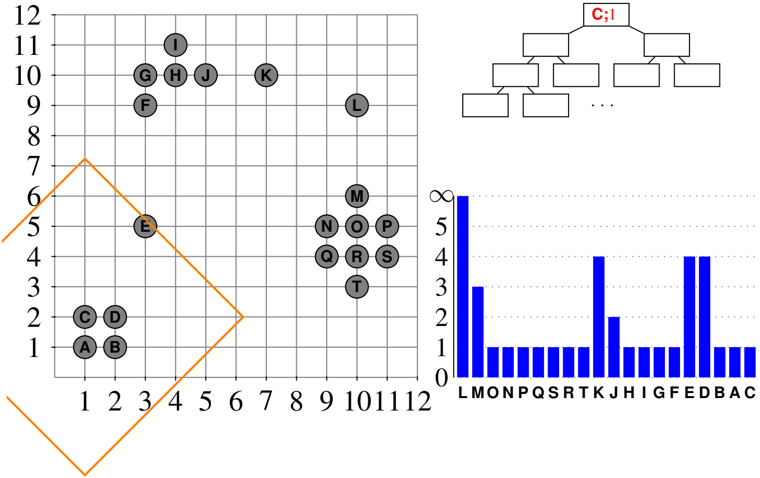
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example



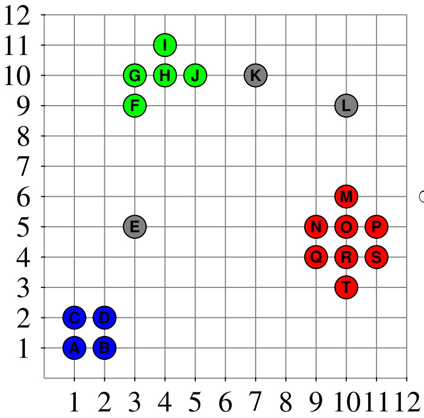
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

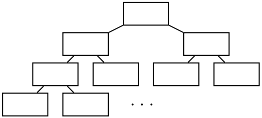
Start example   Skip example

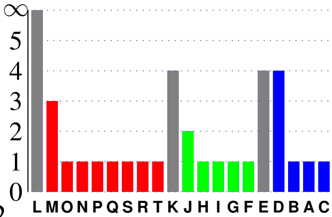


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example
Skip example



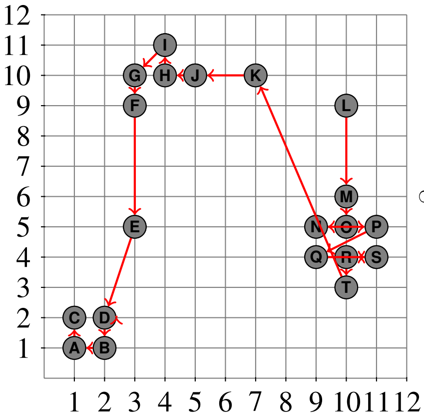


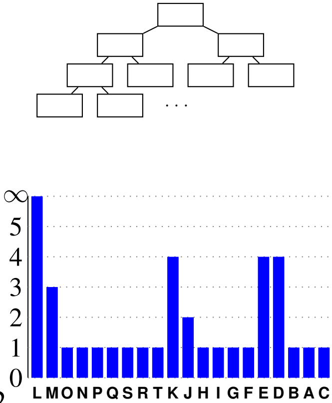


(Intuitively: cluster ≈ valley)

OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

Start example   Skip example

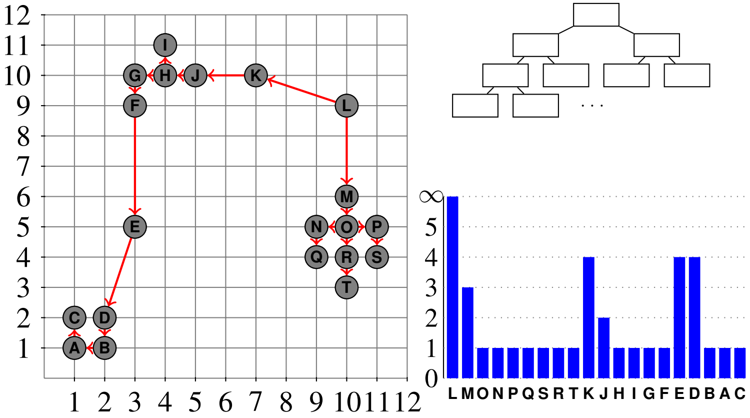




(Cluster order as graph: jumps

OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 2$

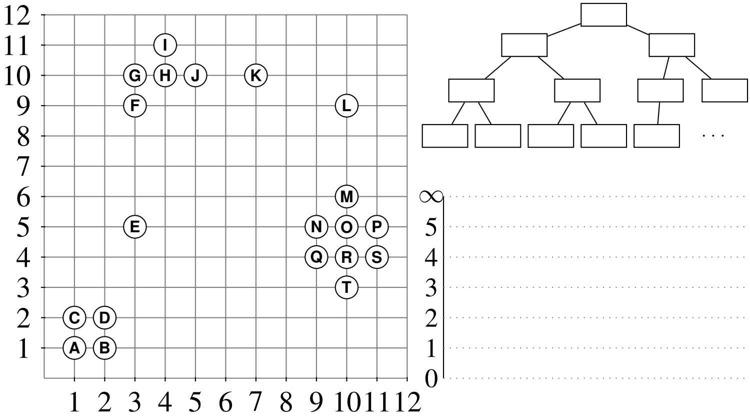
Start example Skip example



(“reached from” graph: ≈ spanning tree)

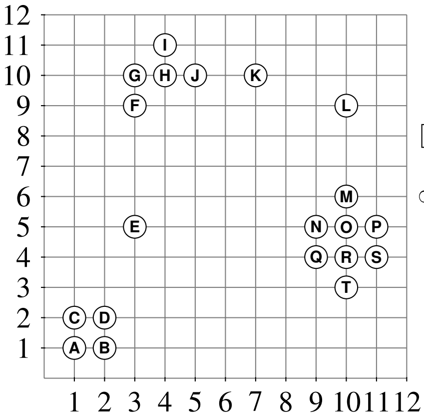
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

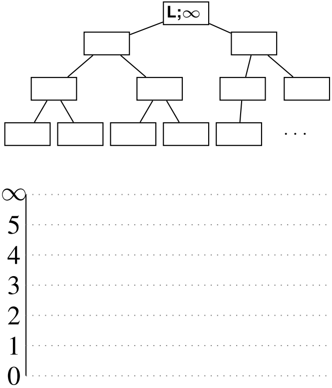
Start example
Skip example



OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

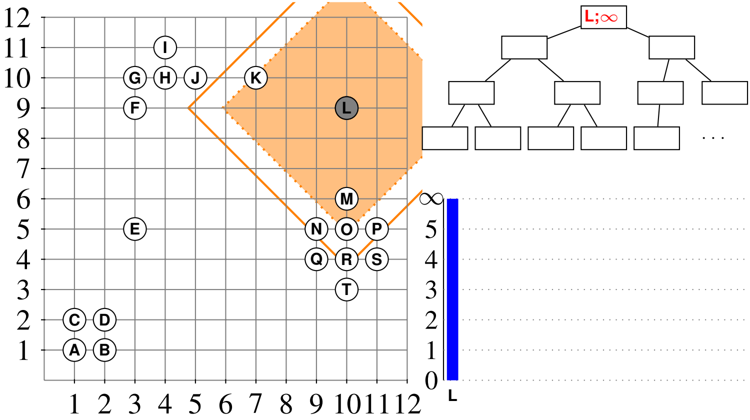
Start example | Skip example





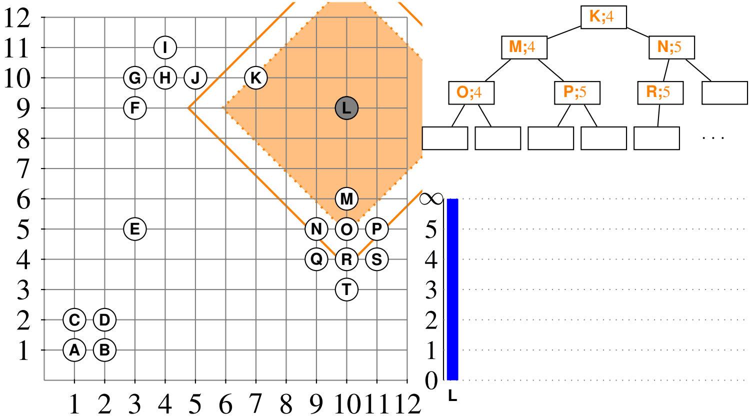
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example   Skip example



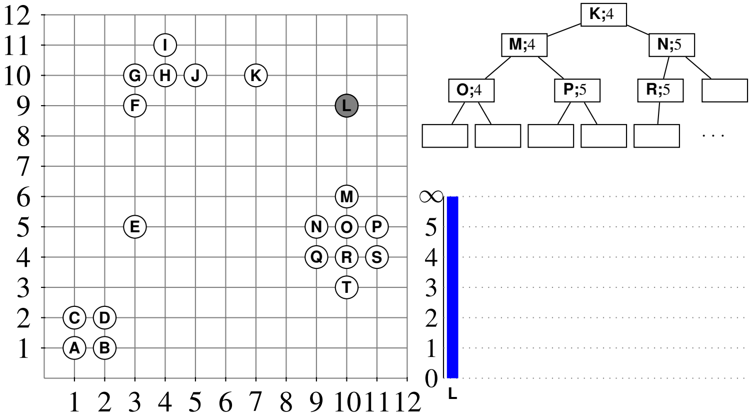
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example Skip example



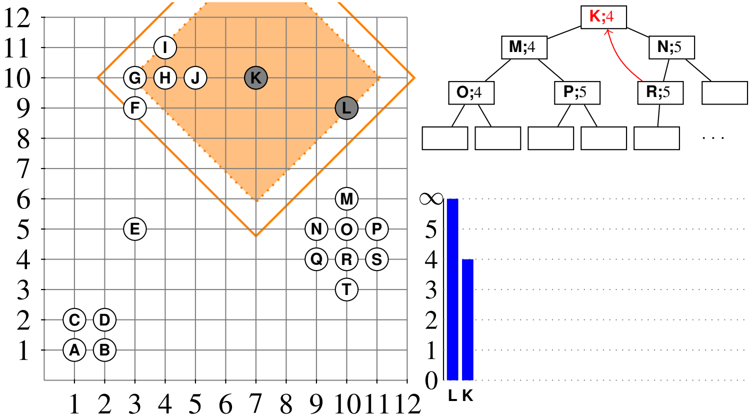
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example Skip example



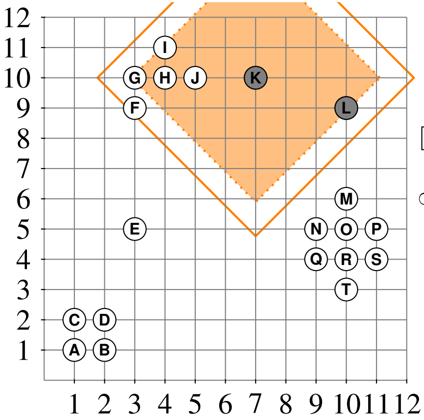
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

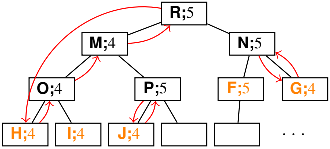
Start example Skip example



OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

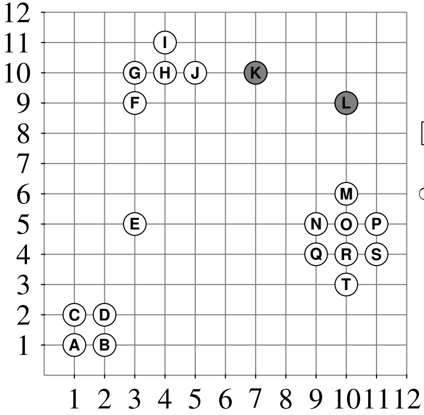
Start example
Skip example

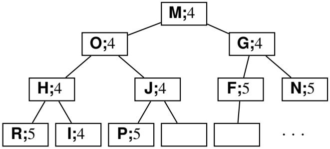




OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

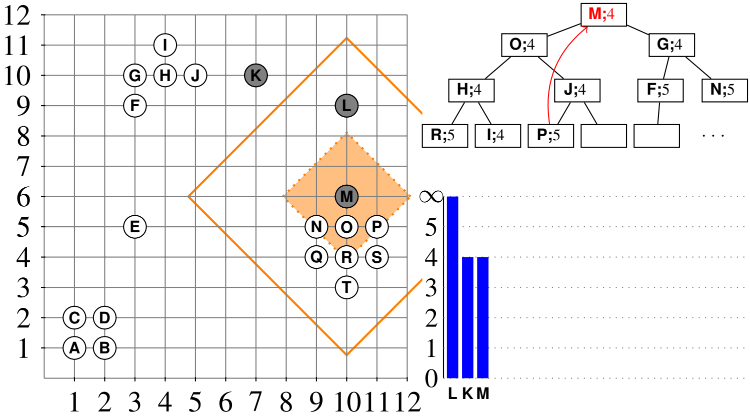
Start example
Skip example





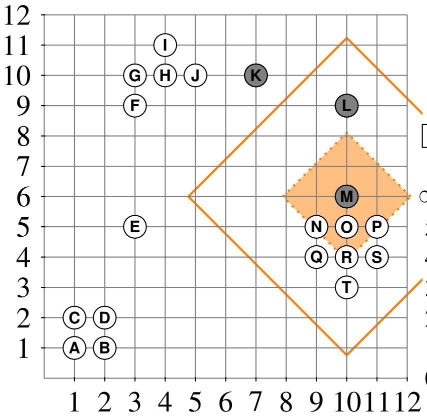
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

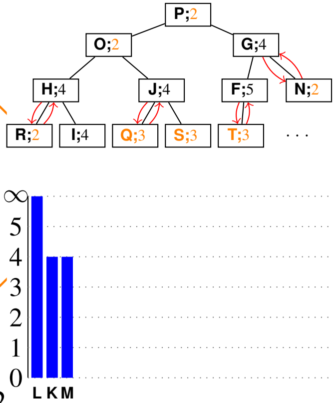
Start example
Skip example



OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

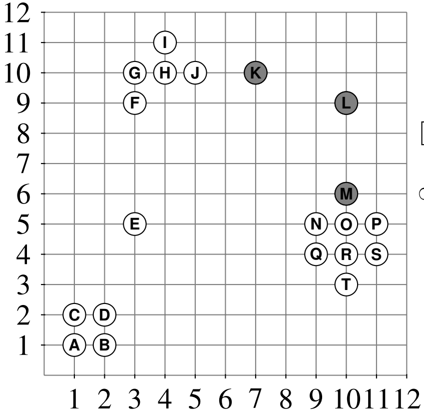
Start example
Skip example

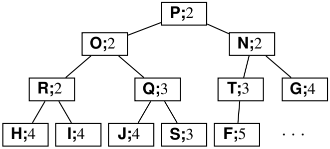


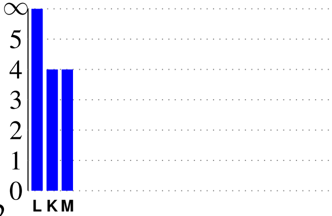


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example   Skip example

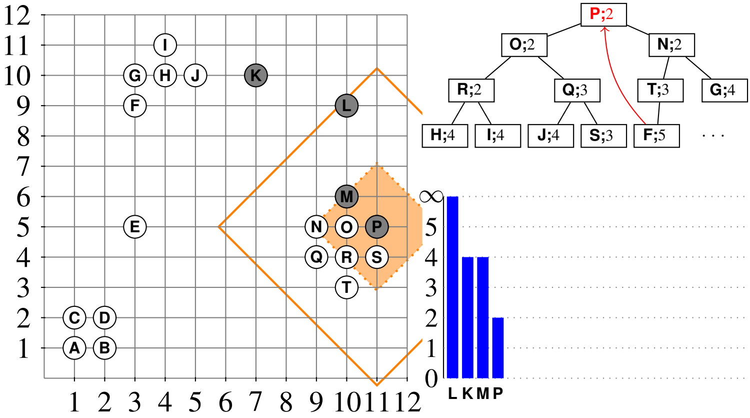






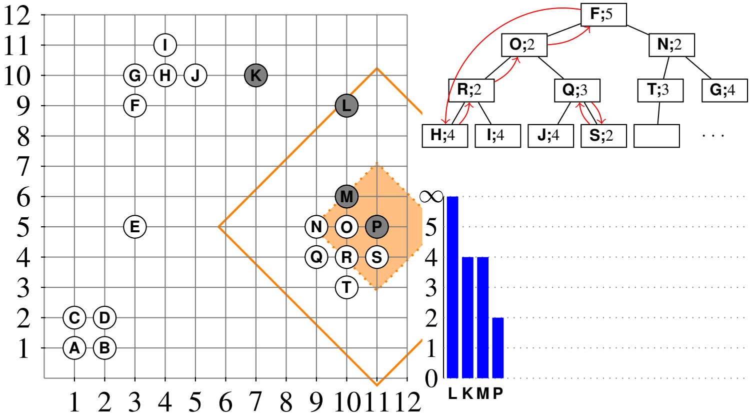
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example
Skip example



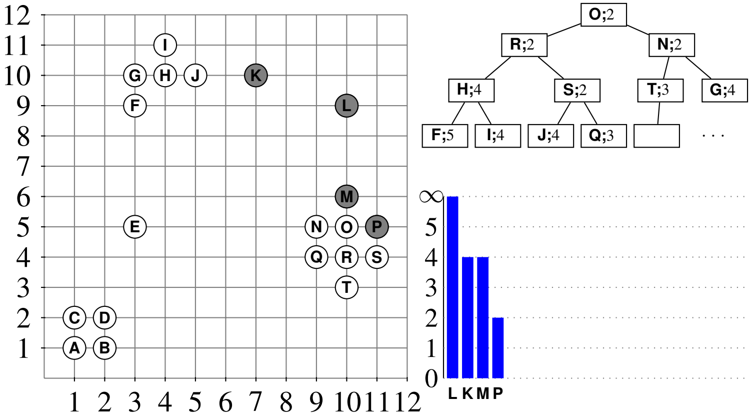
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example　Skip example



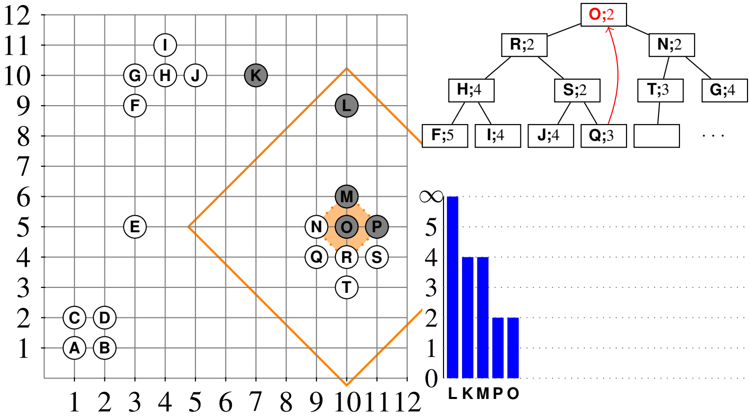
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

▸ Start example ▸▸ Skip example



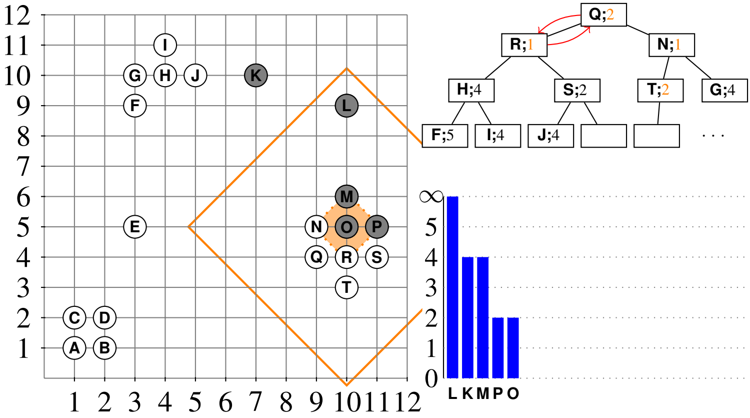
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

▸ Start example ⏩ Skip example



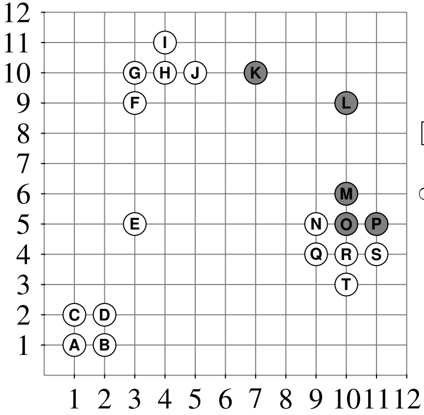
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

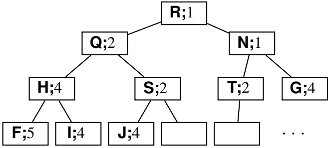
Start example
Skip example

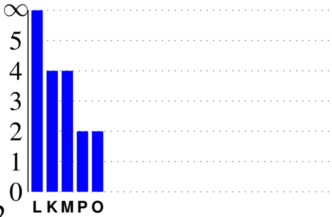


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example | Skip example

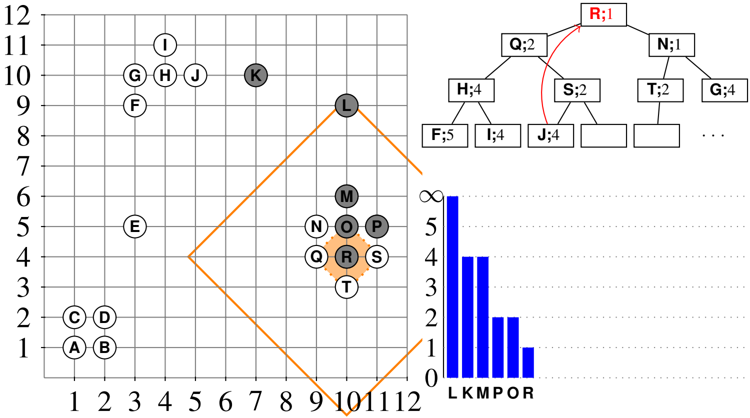






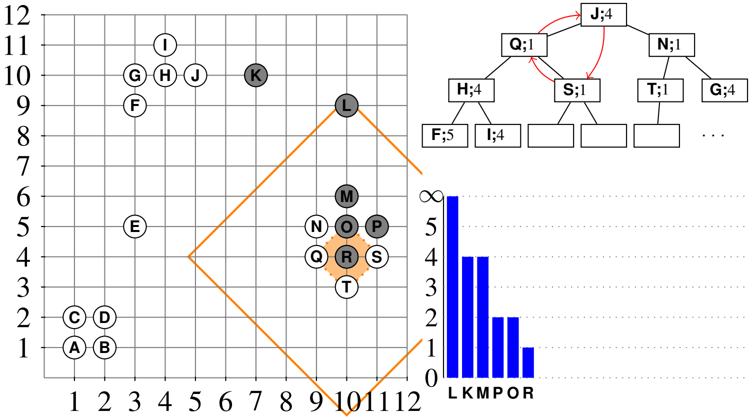
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example
Skip example



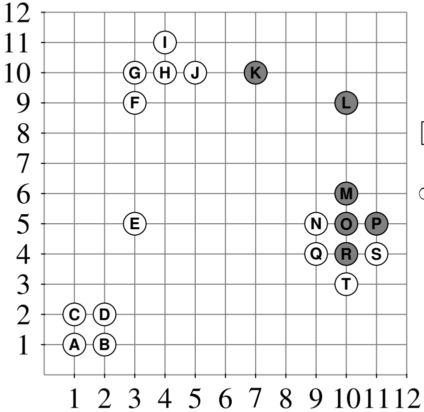
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

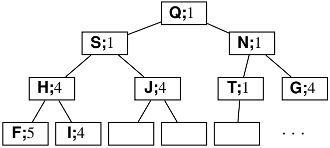
Start example
Skip example

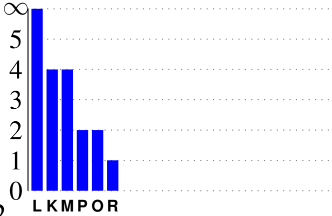


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example   Skip example



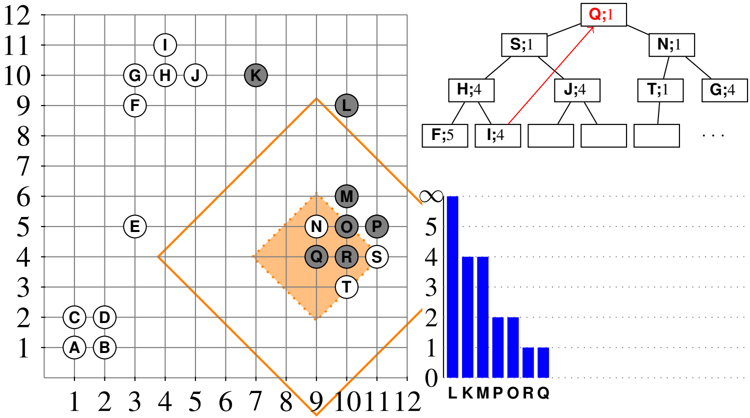




OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

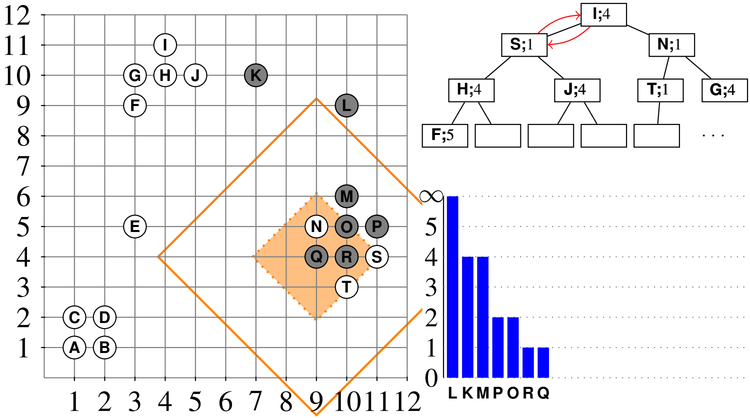
Start example

Skip example



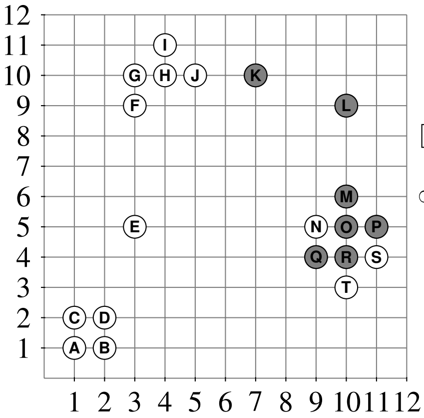
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

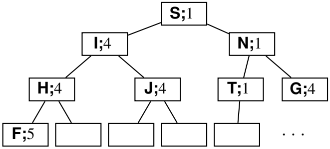
Start example Skip example

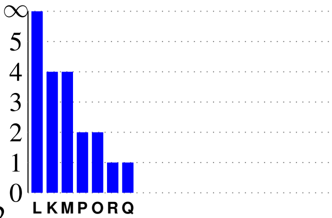


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example   Skip example

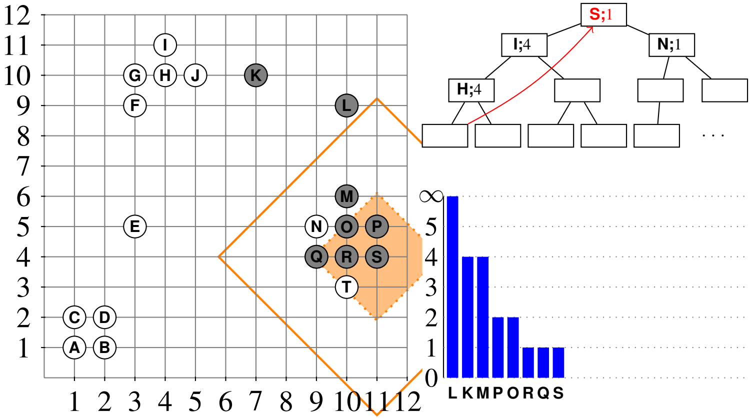






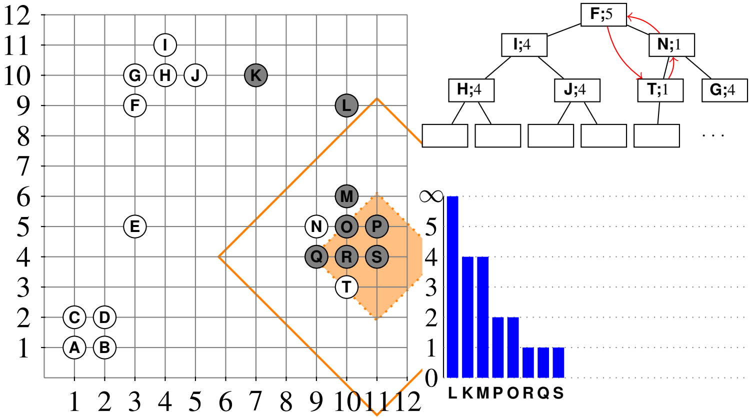
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example Skip example



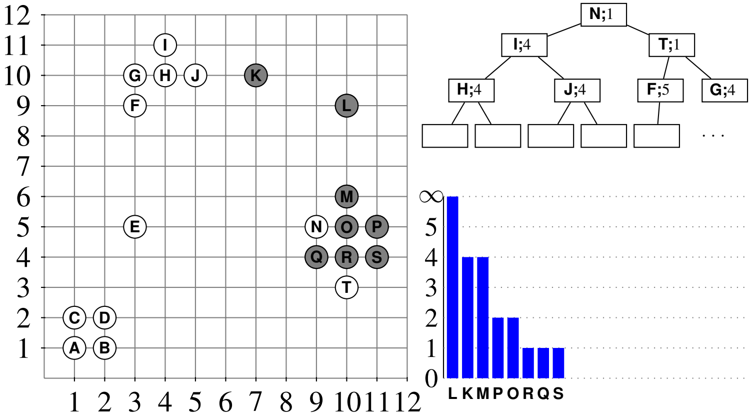
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

▸ Start example ▸▸ Skip example



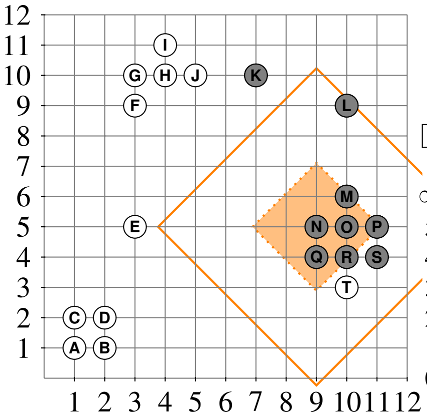
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

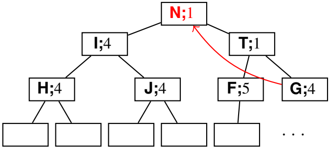
Start example   Skip example

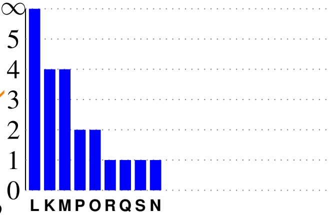


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example · Skip example

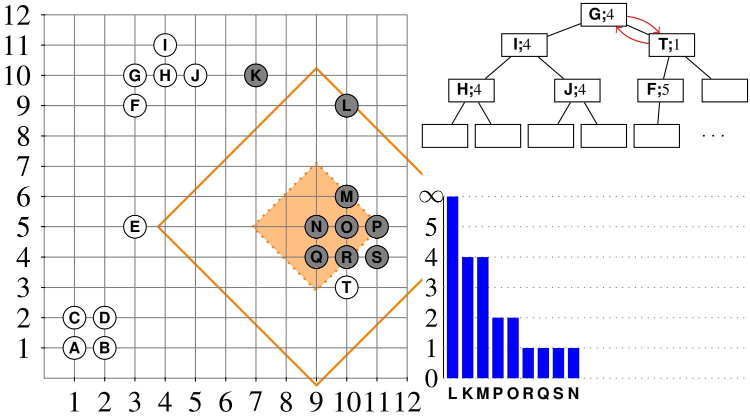






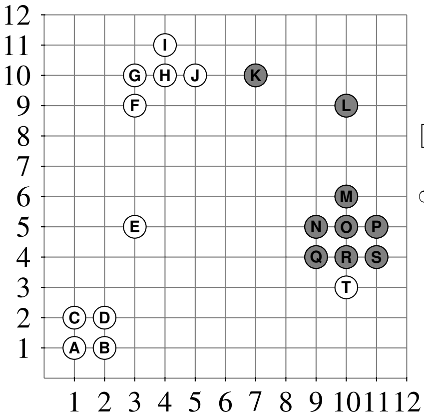
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

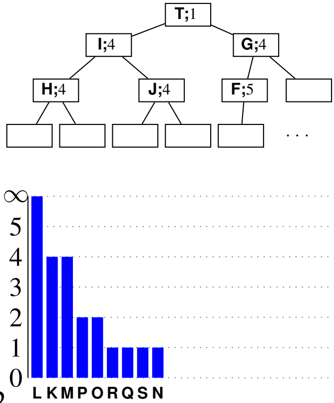
Start example
Skip example



OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example
Skip example

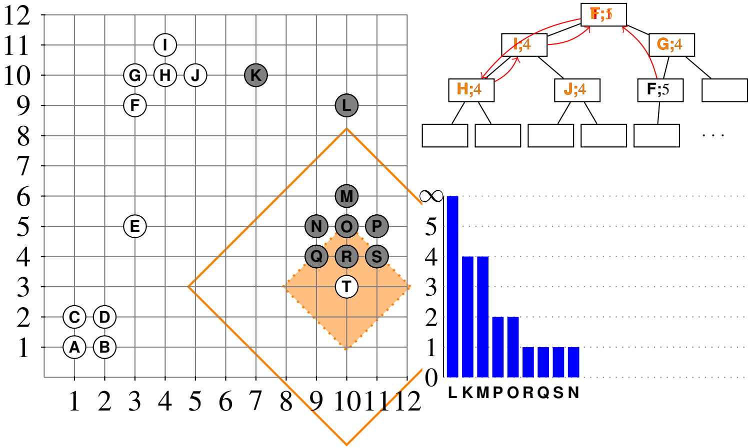




OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

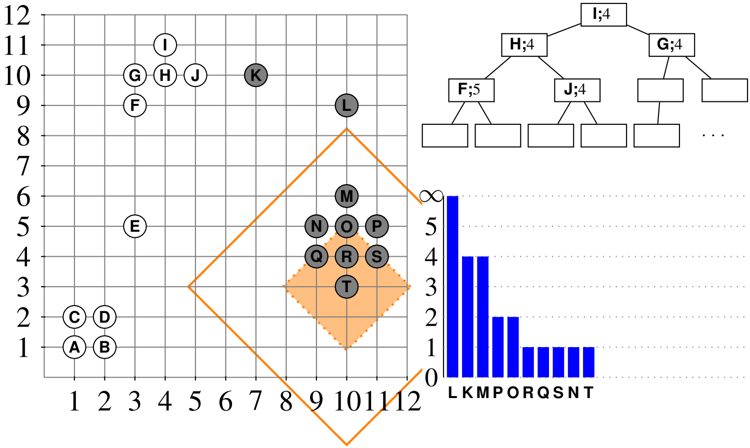
Start example
Skip example

DM566
Melih Kandemir
Introduction
Freq. Pattern Mining
Feature Spaces
Clustering – Basics
Classification – Basics
Bayesian Learning
Learning with Distributions
Discrete Random Variables and Exp.
Variance and Deviations
Continuous Distr.
Bayesian Learning II
Non-Parametric Learning
Non-Parametric Density Estimation
Density-based Clustering
Hierarchical Clustering
Outlier Detection
Summary
References



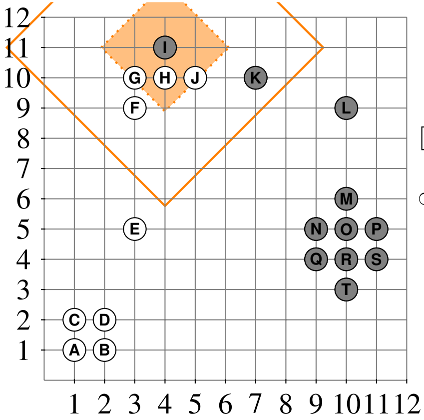
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

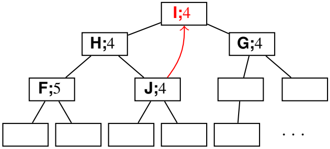
▸ Start example ▸▸ Skip example

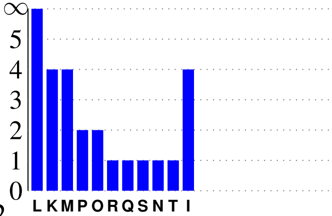


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example Skip example

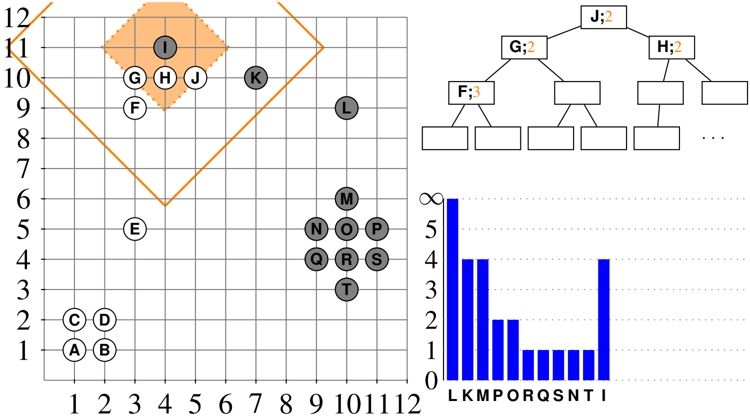






OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

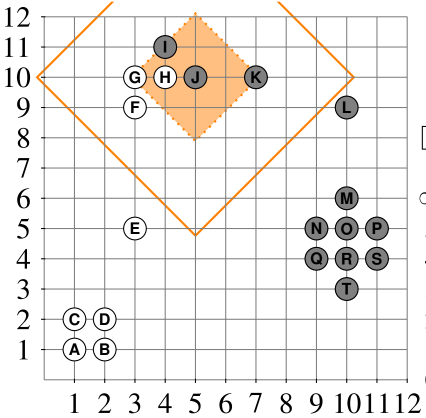
Start example | Skip example

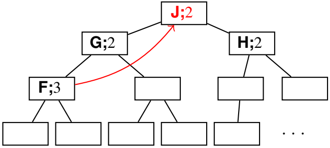


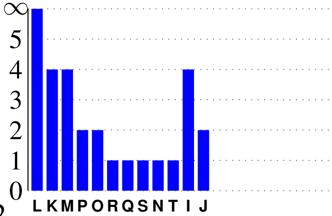
SDU
SYDDANSK UNIVERSITET

OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

▸ Start example ▸▸ Skip example

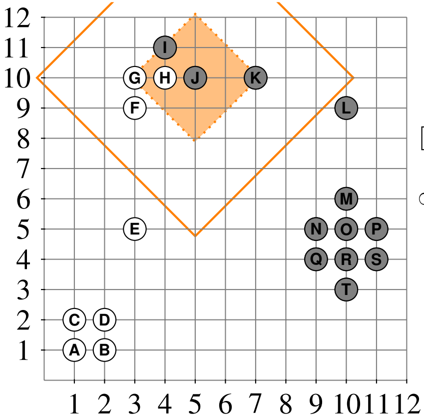


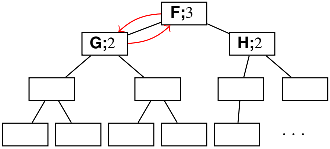


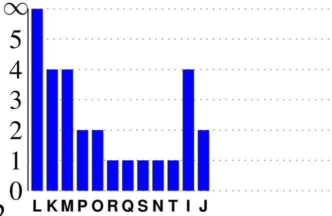


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example Skip example

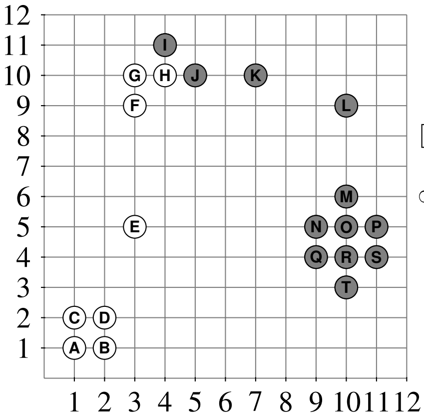


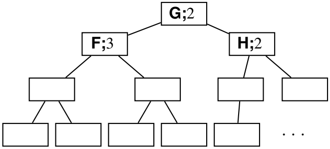


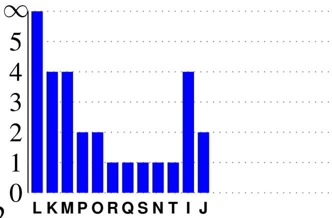


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example   Skip example

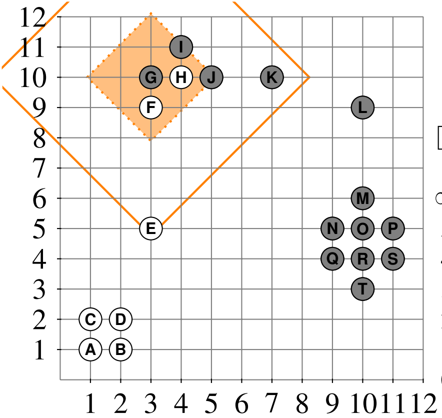


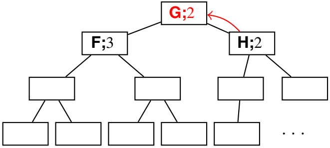


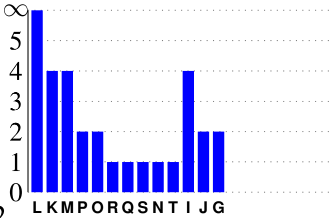


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example Skip example

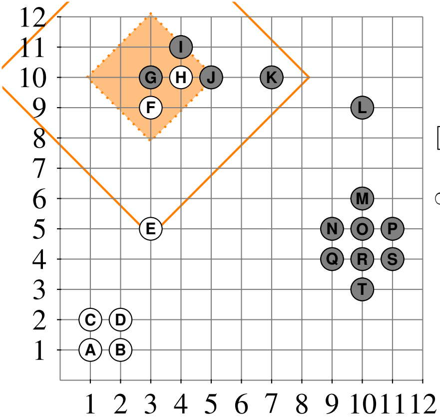


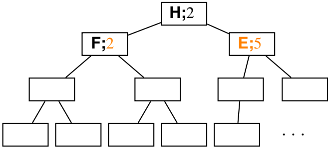


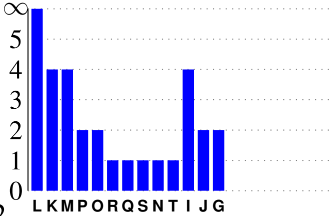


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example Skip example

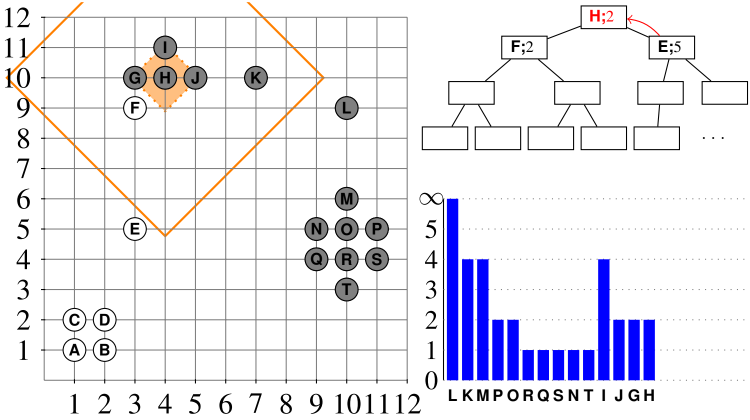






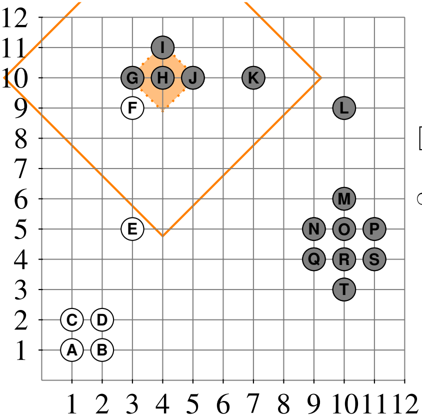
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

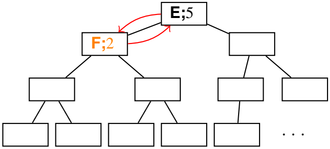
Start example
Skip example

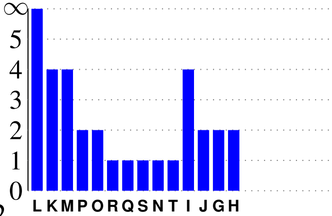


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example
Skip example

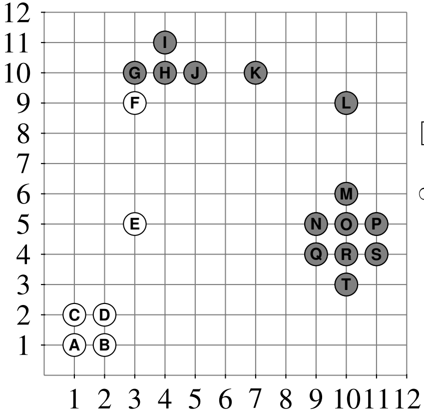


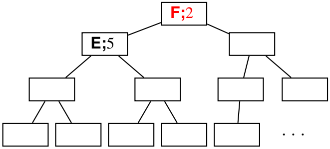


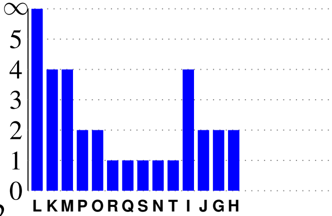


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example Skip example

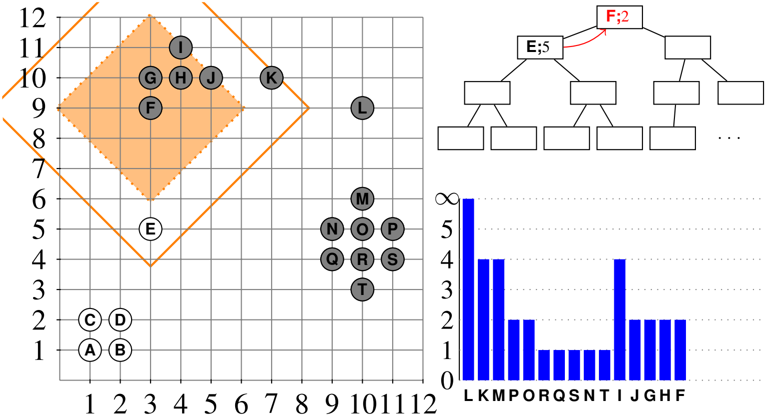






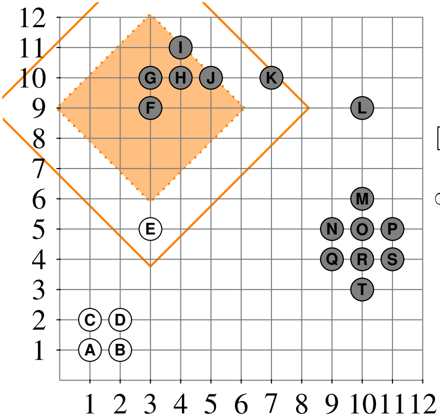
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

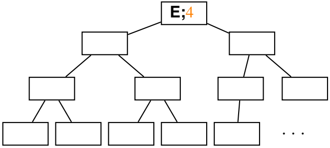
Start example
Skip example

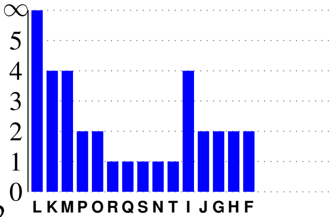


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example   Skip example

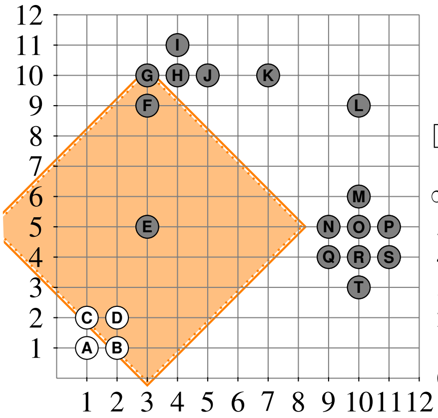


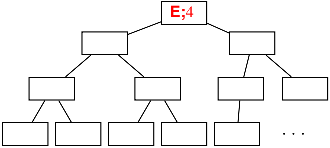


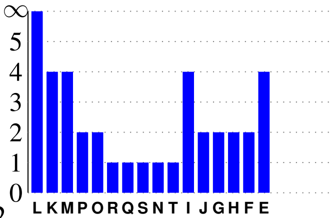


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example   Skip example

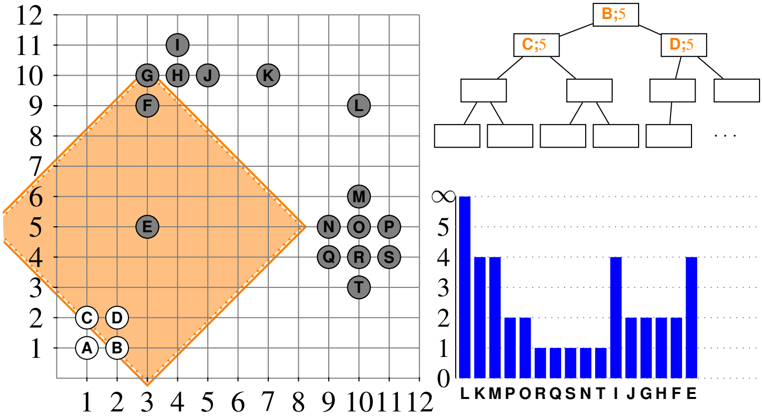






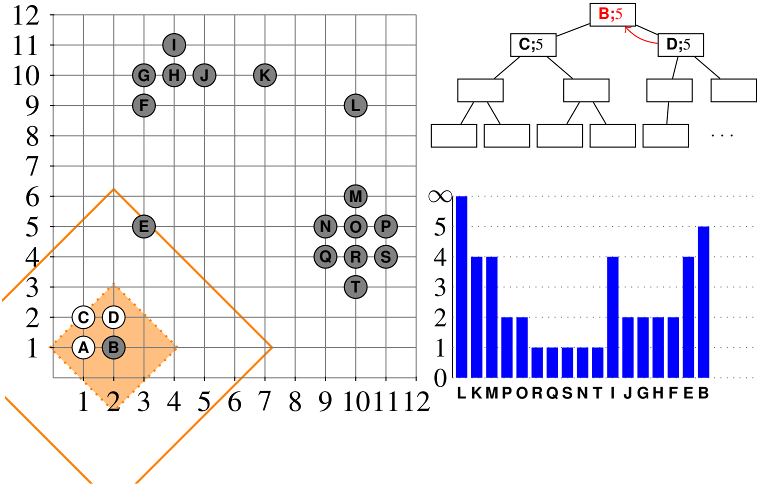
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example | Skip example



OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

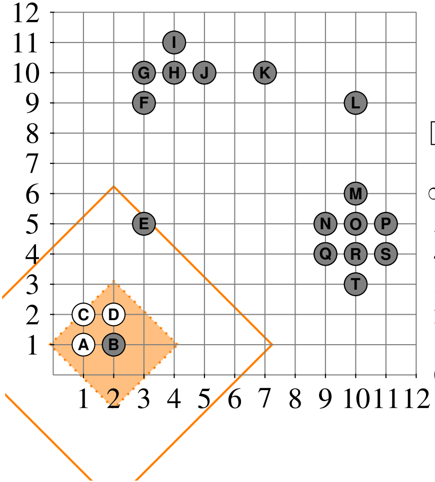
Start example Skip example

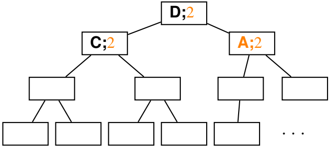


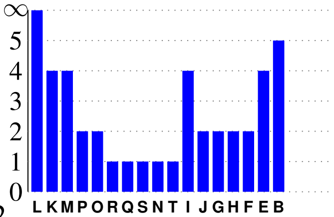
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example

Skip example

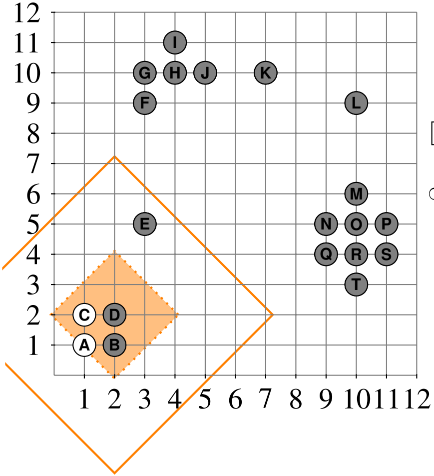


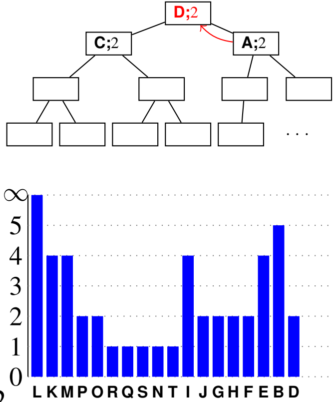




OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

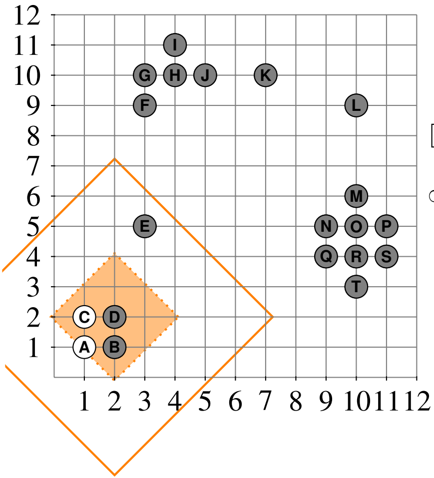
Start example
Skip example

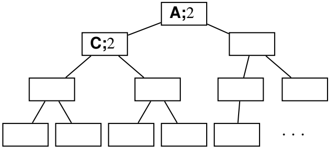


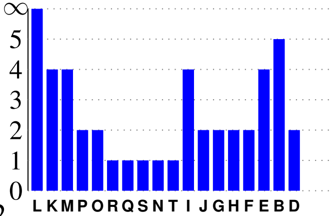


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example | Skip example

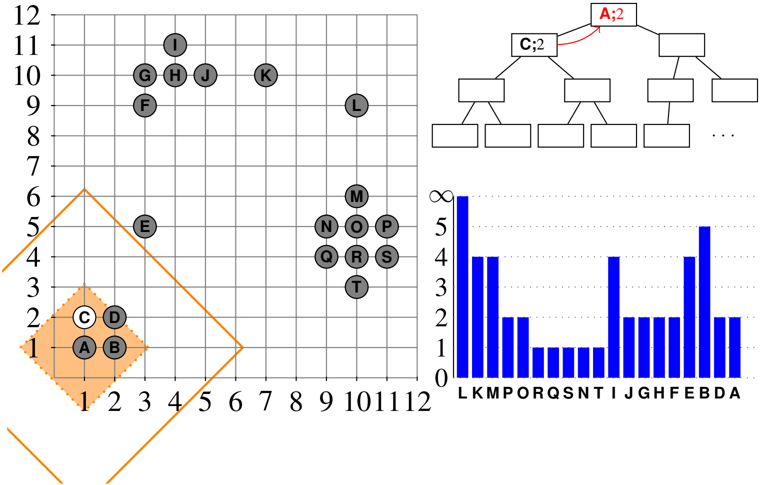






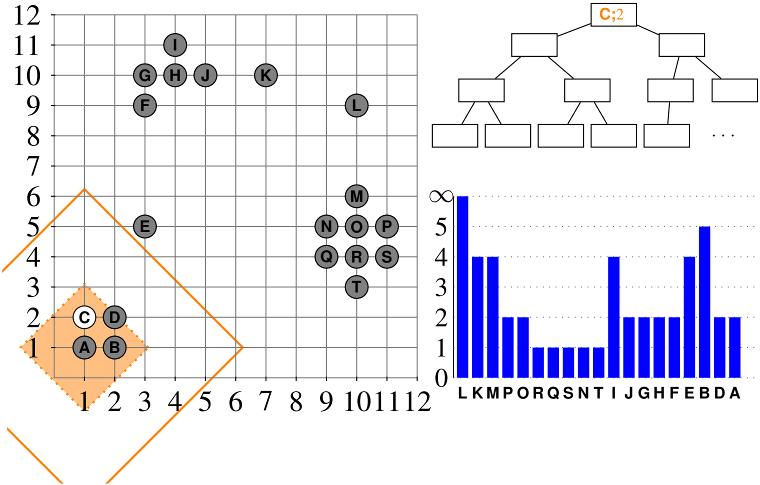
OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

▸ Start example ▸▸ Skip example



OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

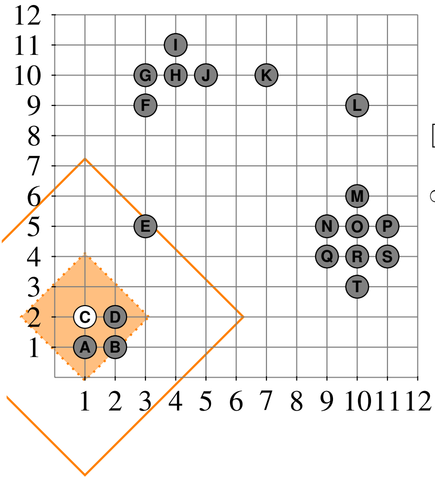
Start example
Skip example

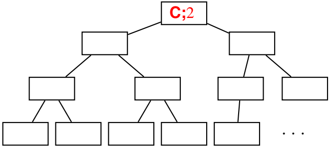


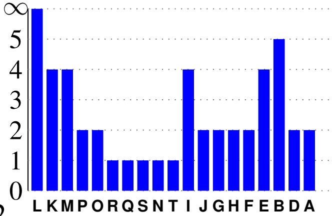
SDU SYDDANSK UNIVERSITET

OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example Skip example

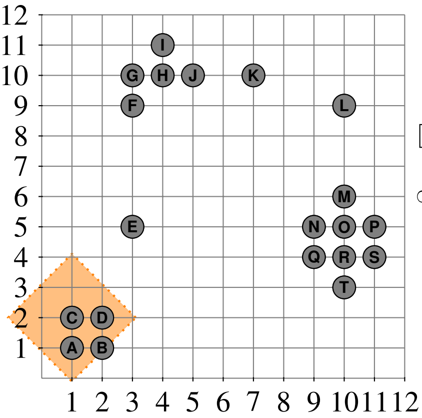


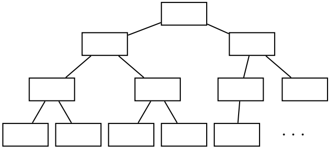


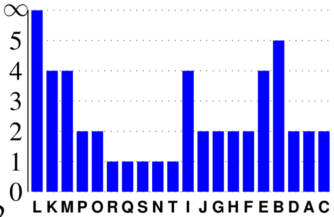


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example   Skip example

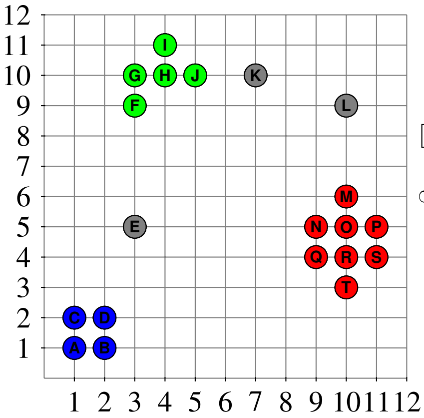


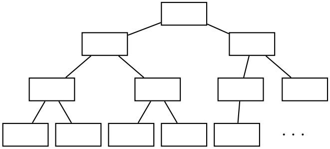


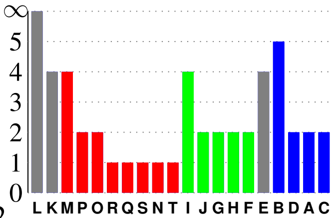


OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example   Skip example





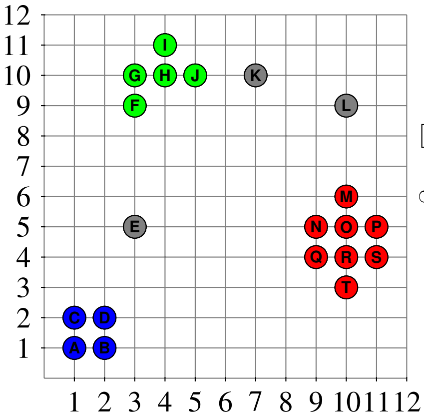


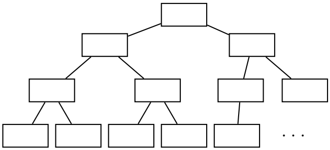
(Intuitively: cluster $\approx$ valley)

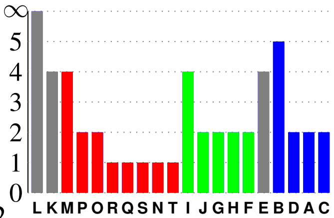
Previous “hill” would belong to the cluster, subsequent “hill” does not!

OPTICS – Example $\varepsilon = 5, \mathrm{MinPts} = 4$

Start example | Skip example





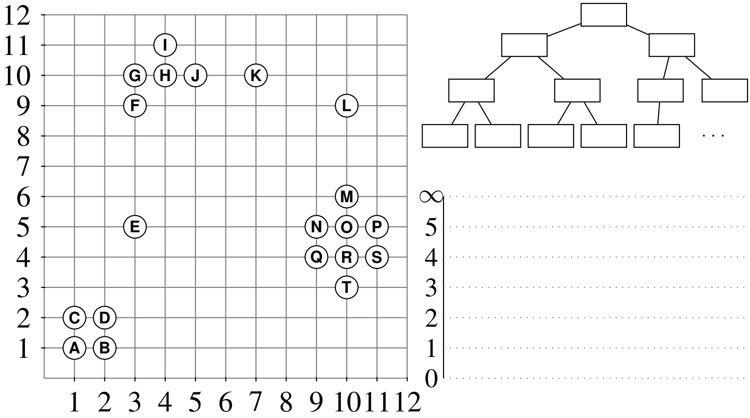


Plot is smoothed!

Heap happens to be more expensive here (not necessarily!)

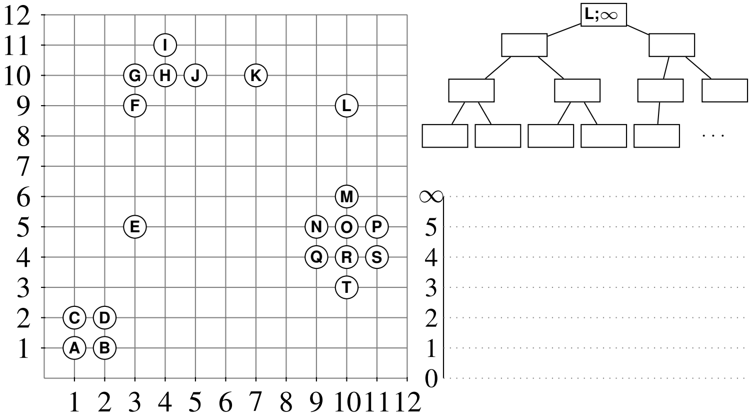
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example



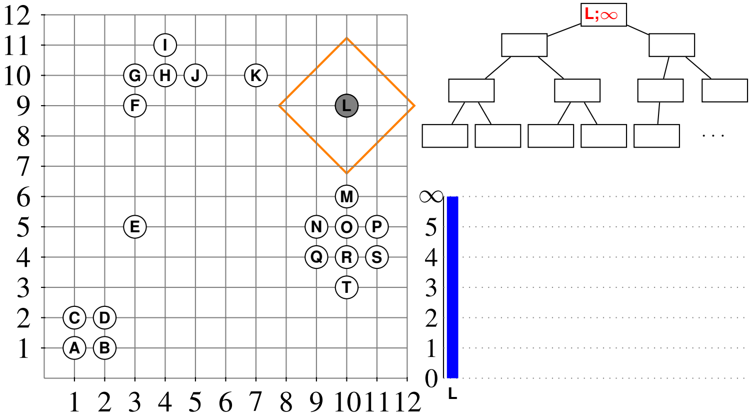
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example



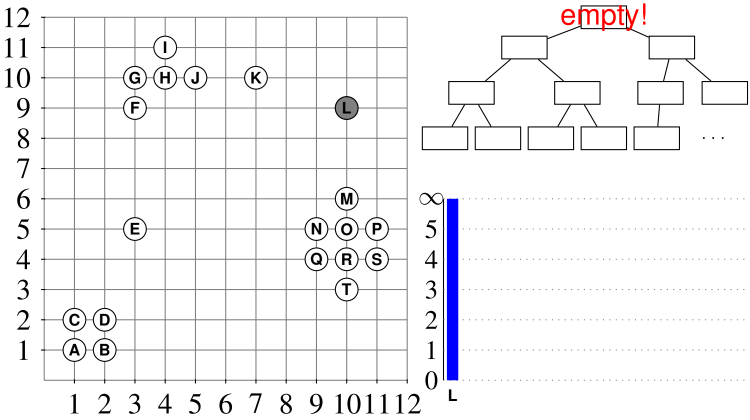
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example



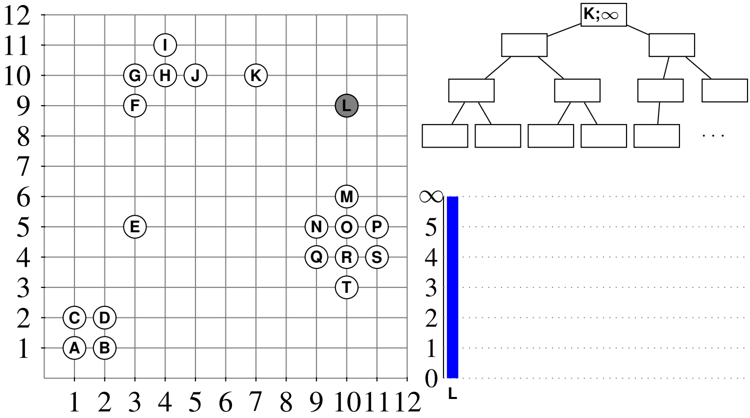
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example ⏩ Skip example



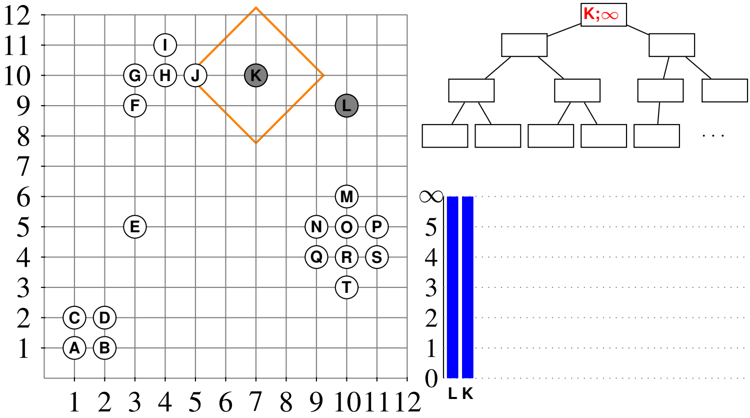
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example   Skip example



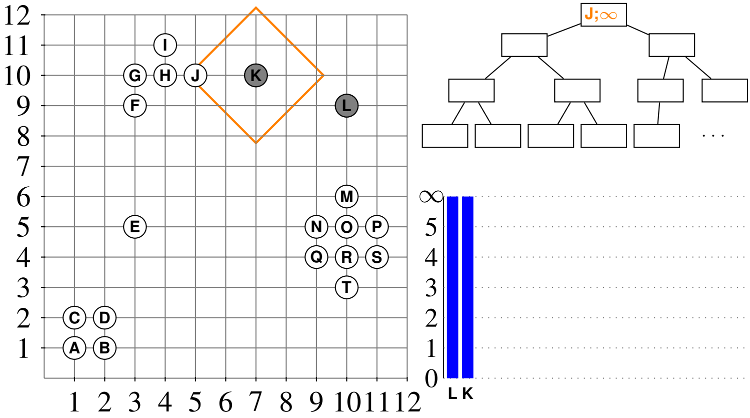
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example



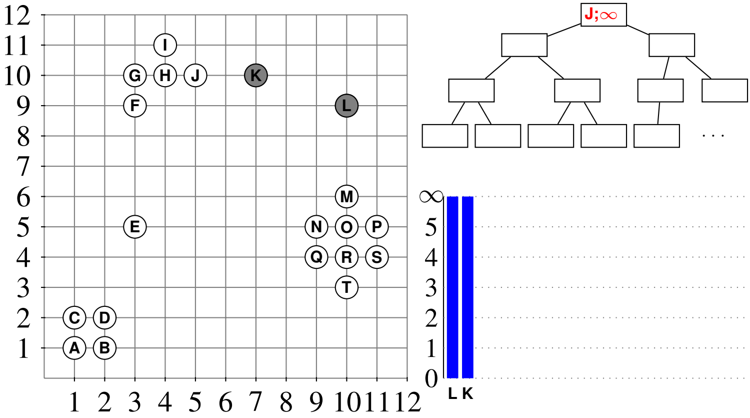
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example



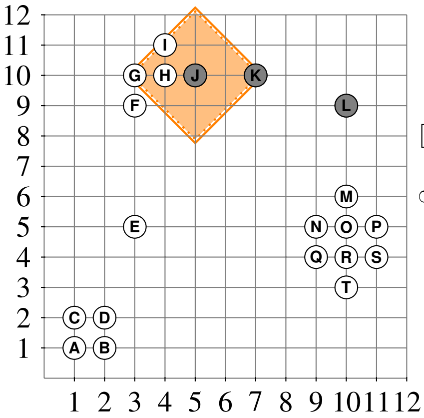
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

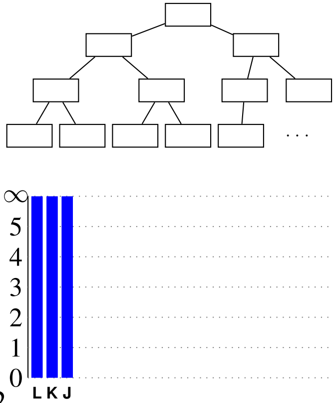
Start example | Skip example



OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

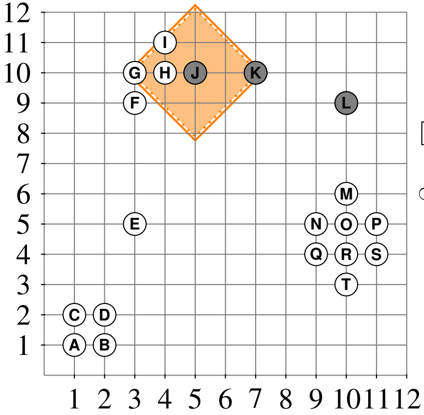
▸ Start example ▸▸ Skip example

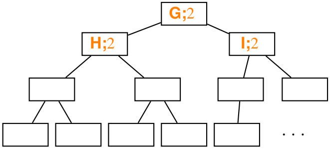


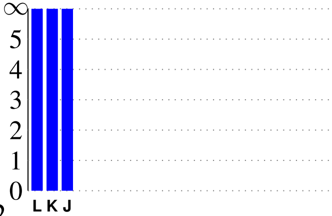


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example Skip example

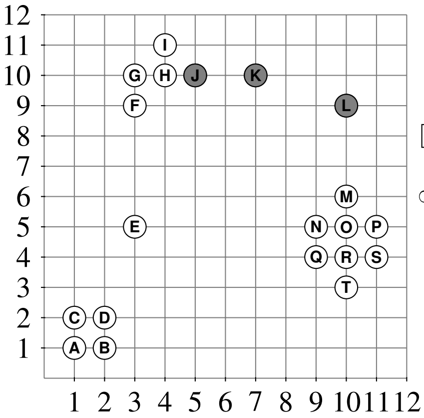


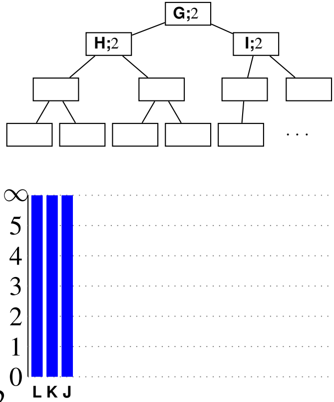




OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

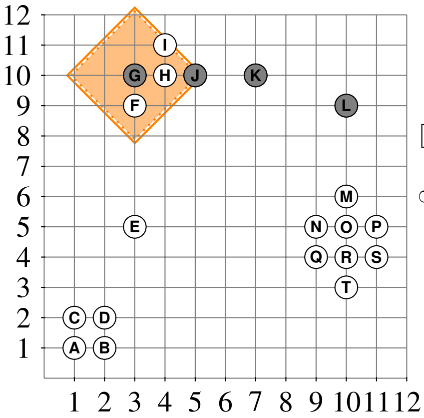
Start example
Skip example

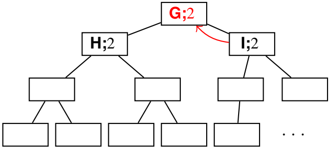


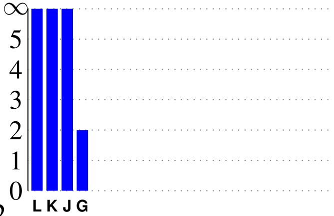


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example Skip example

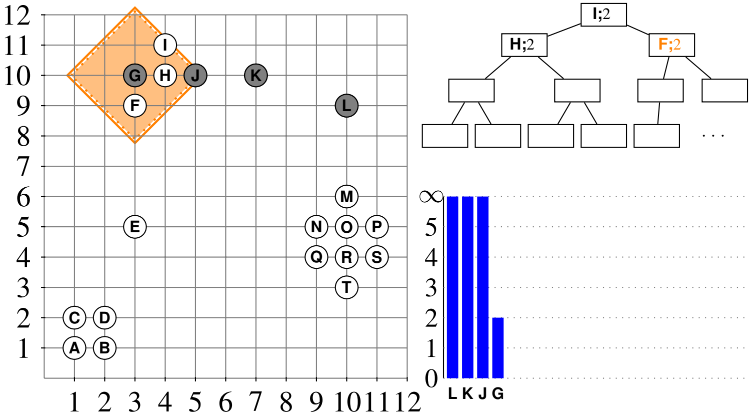






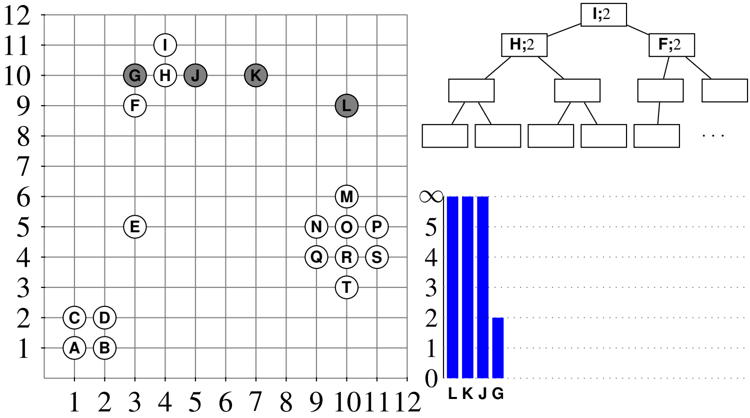
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example   Skip example



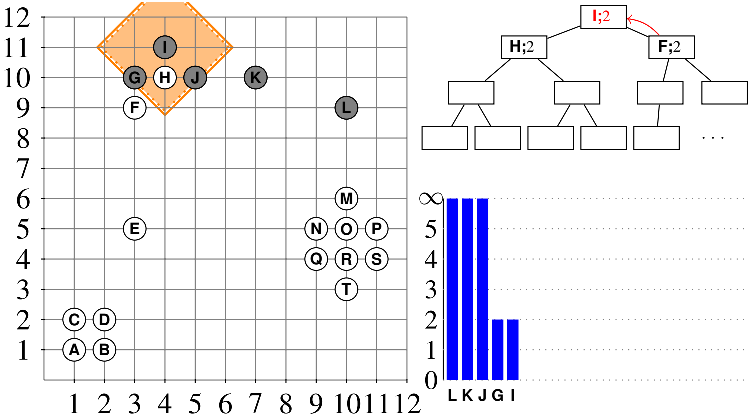
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example



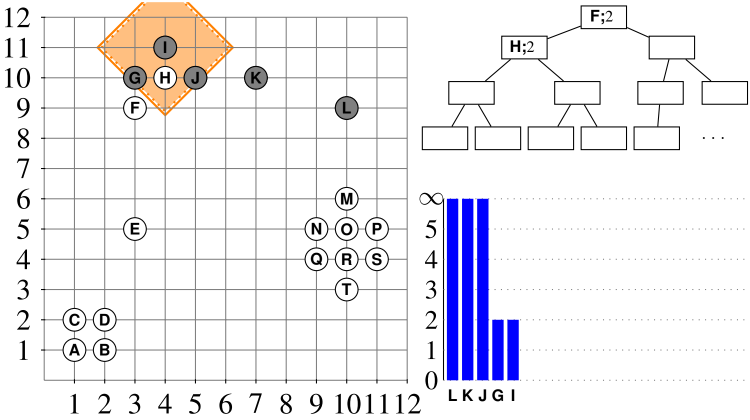
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example Skip example



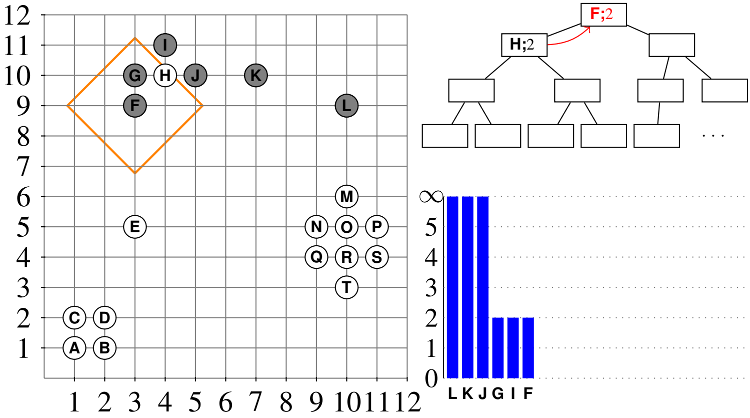
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example | Skip example



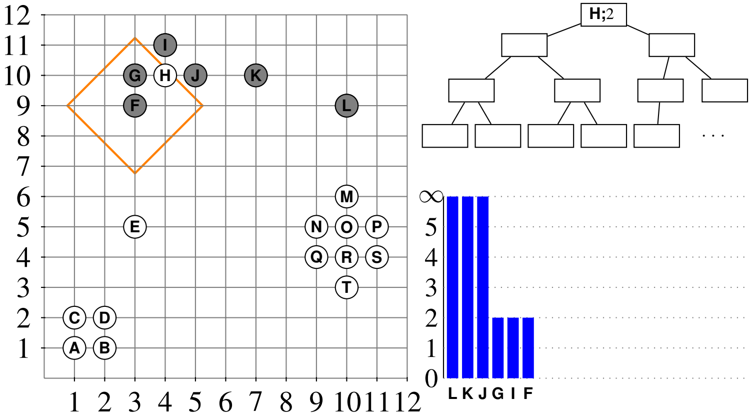
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example



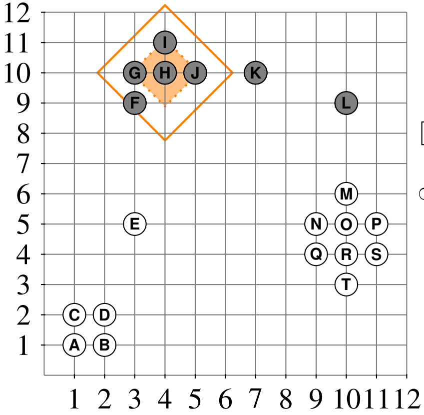
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

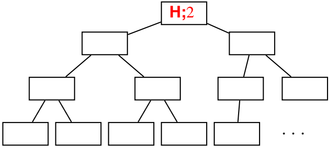
Start example Skip example

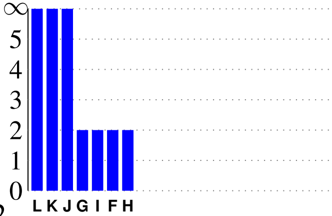


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example

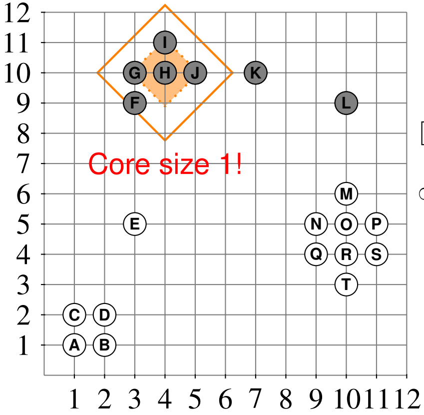


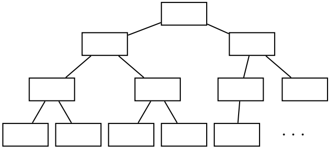


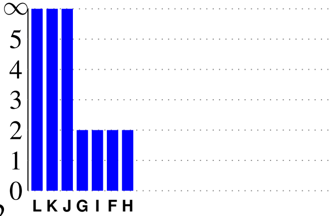


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example

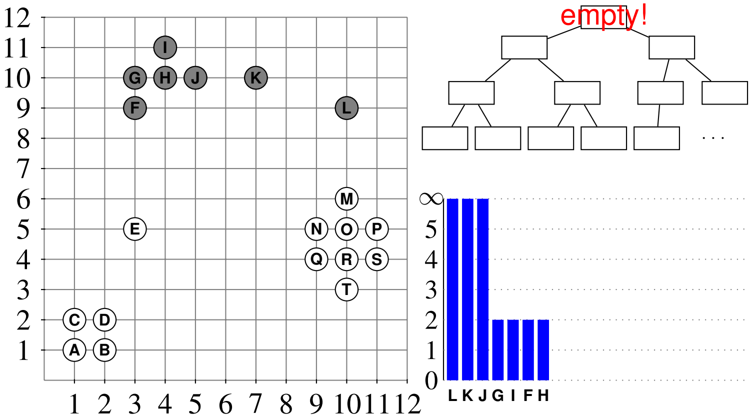






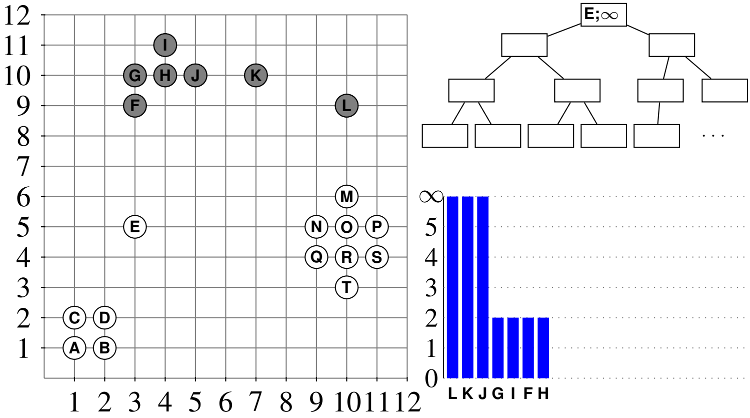
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example Skip example



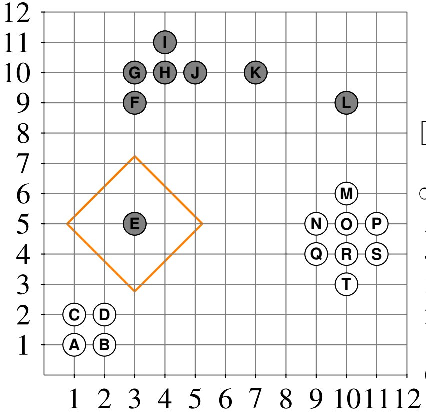
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

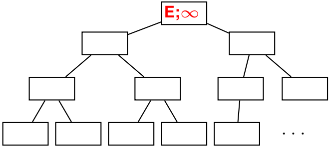
Start example Skip example

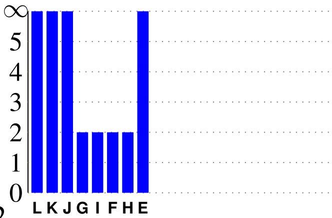


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example   Skip example



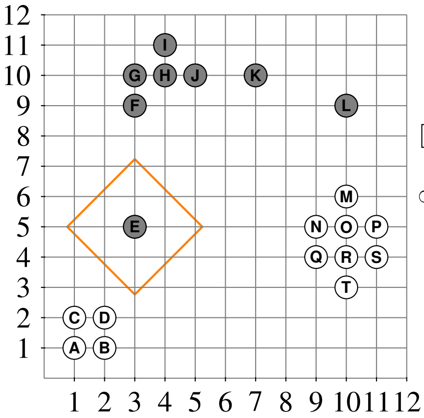


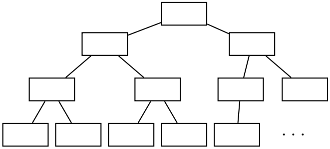


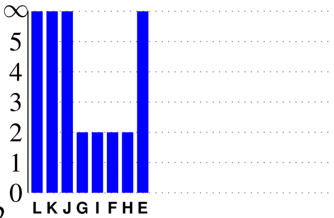
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example

Skip example

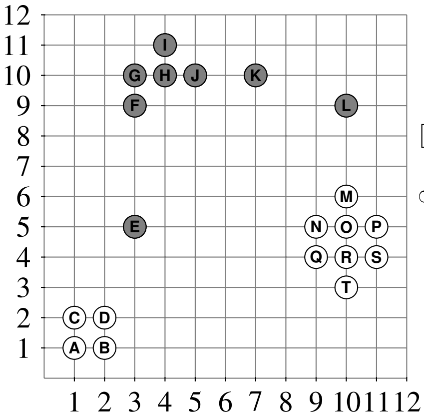


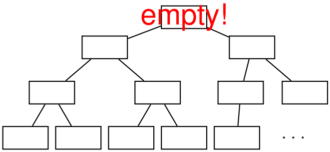


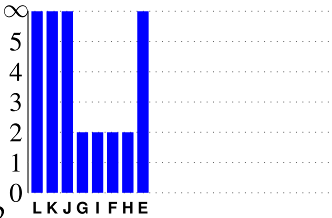


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

▸ Start example ▸▸ Skip example

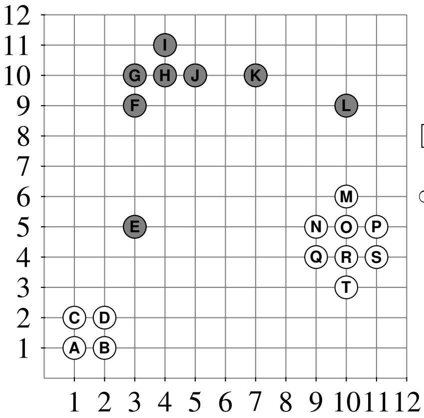


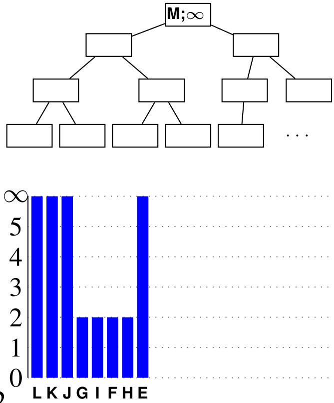




OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

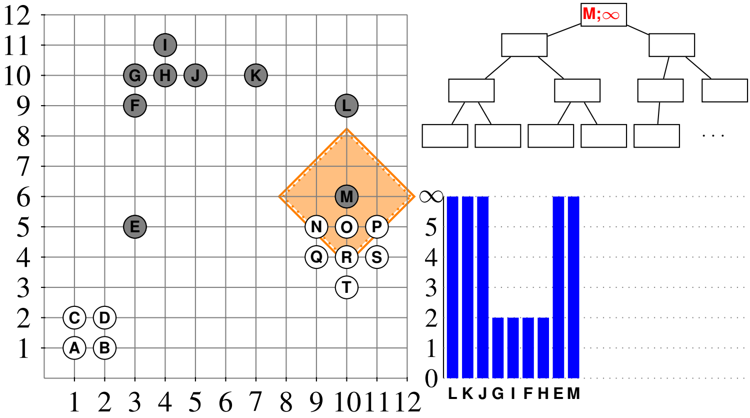
Start example
Skip example





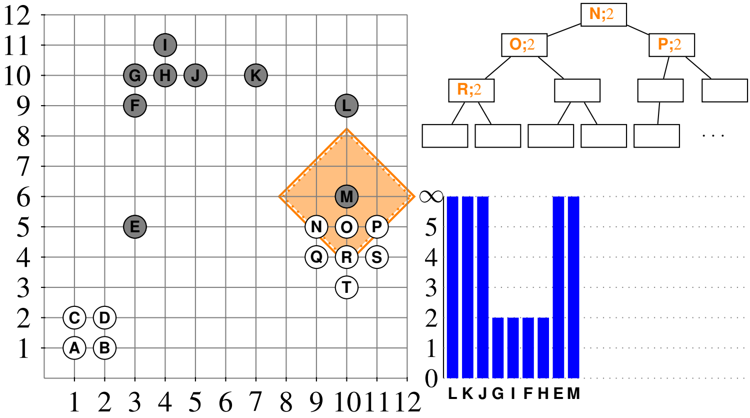
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example Skip example



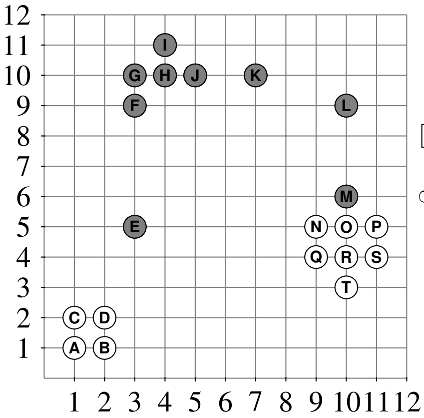
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

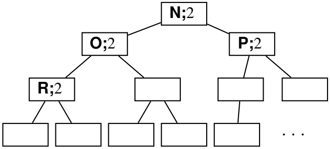
Start example | Skip example

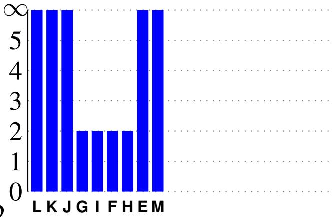


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example Skip example

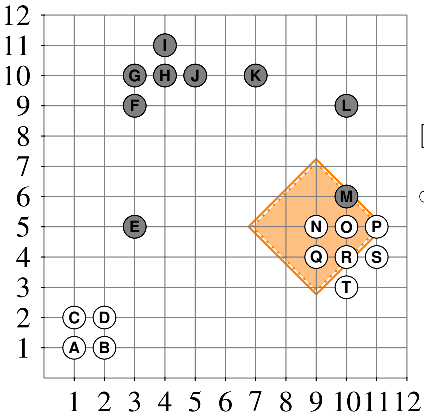


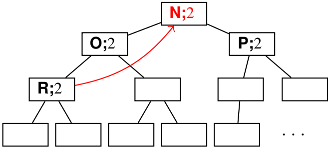


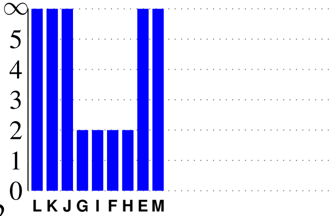


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example

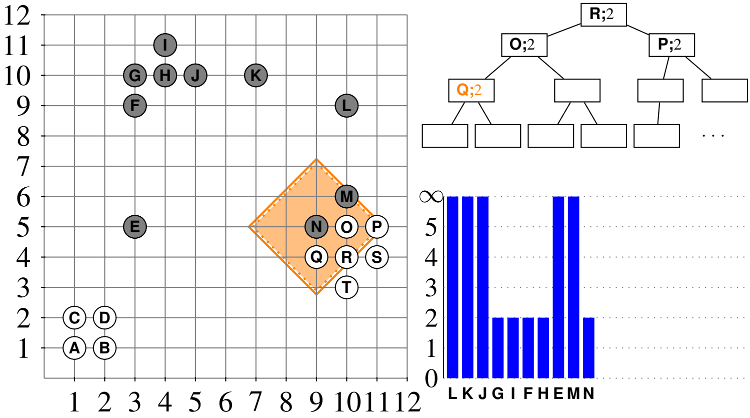






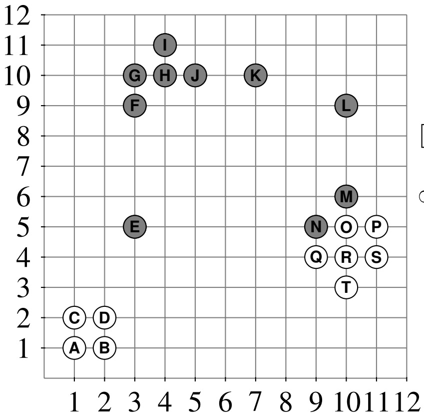
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

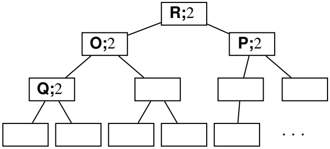
Start example
Skip example

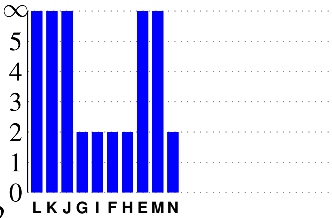


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example | Skip example

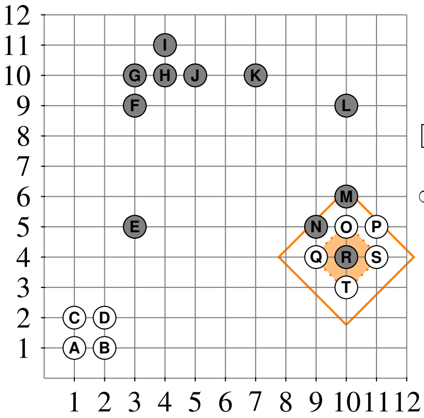


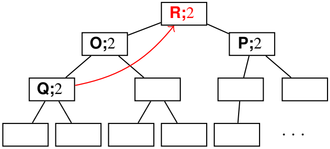


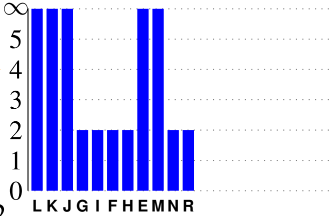


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

▸ Start example ▸▸ Skip example

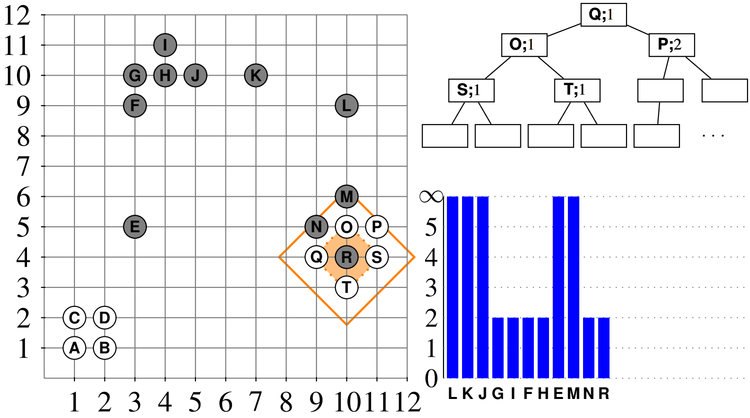






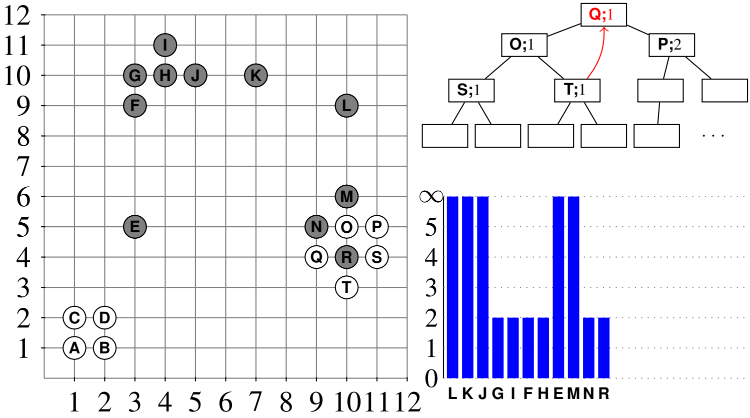
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example



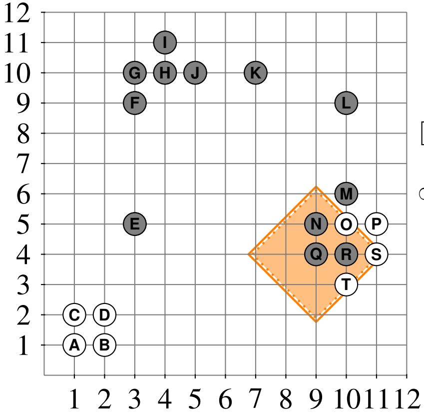
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

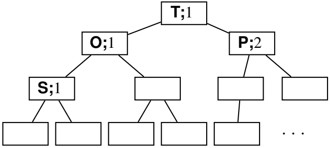
▸ Start example ▸▸ Skip example

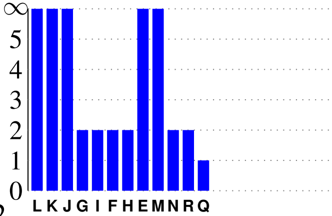


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

▸ Start example ▸▸ Skip example

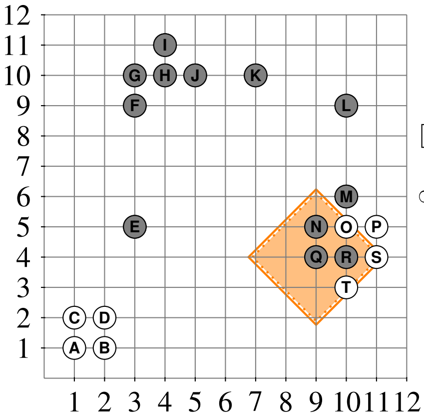


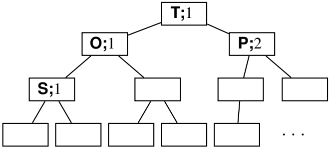


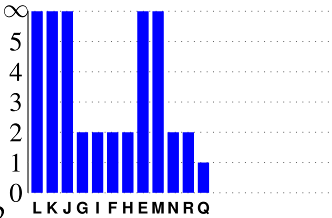


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example Skip example

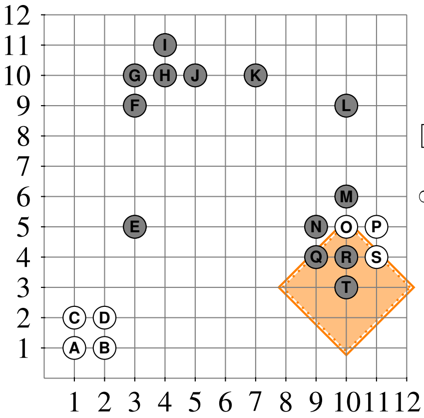


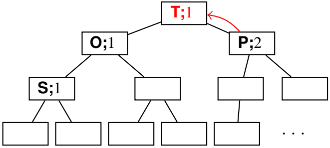


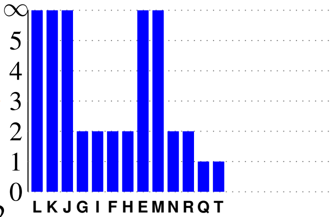


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

▸ Start example ⏩ Skip example

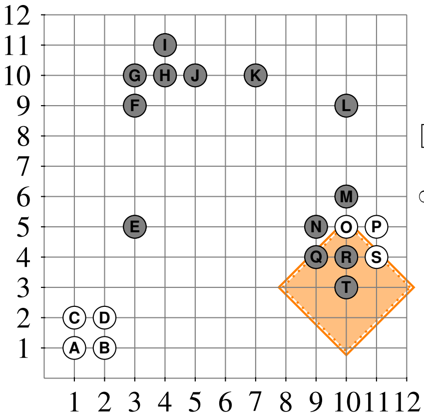


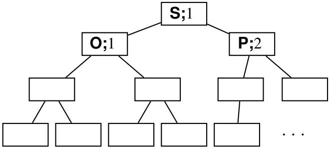


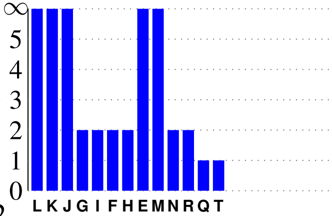


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example   Skip example







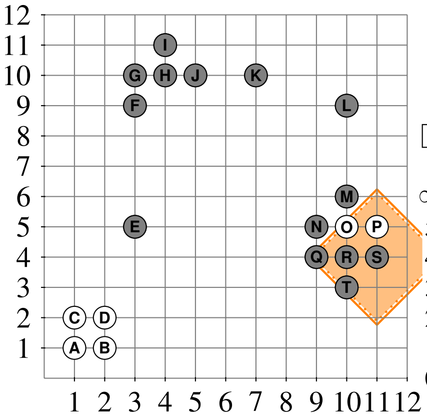
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

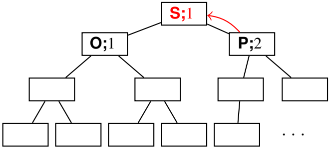
Start example | Skip example

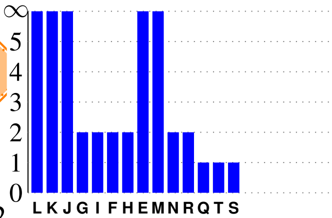
DM566

Melih Kandemir

Introduction
Freq. Pattern Mining
Feature Spaces
Clustering – Basics
Classification – Basics
Bayesian Learning
Learning with Distributions
Discrete Random Variables and Exp.
Variance and Deviations
Continuous Distr.
Bayesian Learning II
Non-Parametric Learning
Non-Parametric Density Estimation
Density-based Clustering
Hierarchical Clustering
Outlier Detection
Summary
References

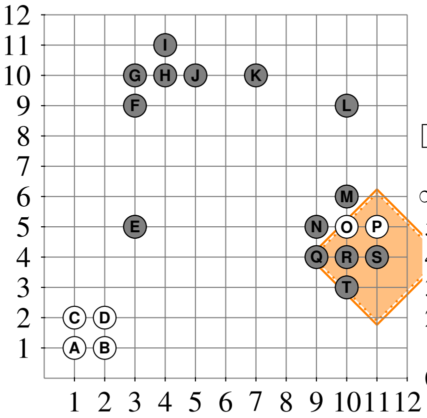


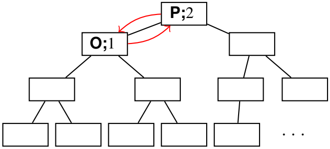


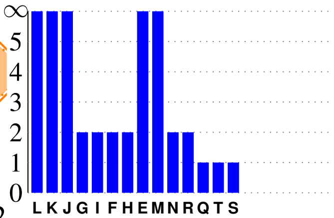


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example   Skip example

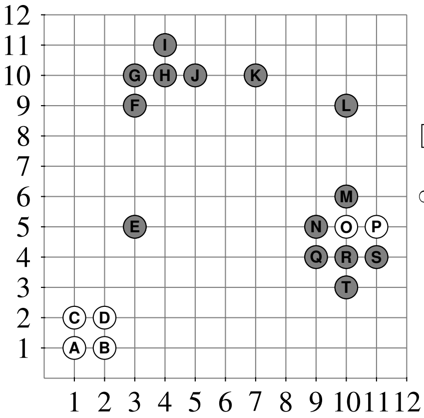


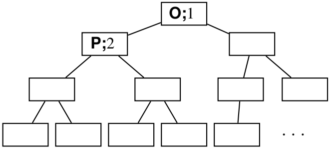


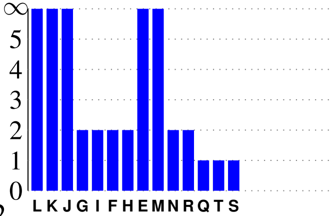


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example   Skip example







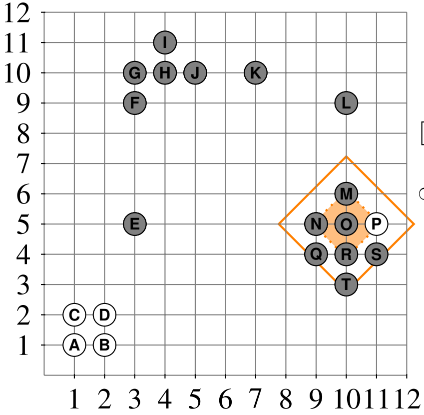
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

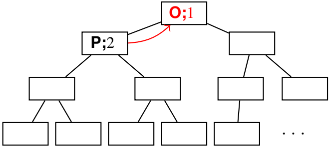
Start example   Skip example

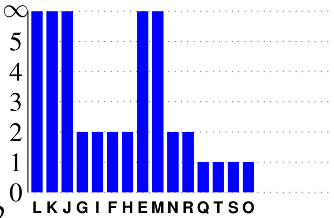
DM566

Melih Kandemir

Introduction
Freq. Pattern Mining
Feature Spaces
Clustering – Basics
Classification – Basics
Bayesian Learning
Learning with Distributions
Discrete Random Variables and Exp.
Variance and Deviations
Continuous Distr.
Bayesian Learning II
Non-Parametric Learning
Non-Parametric Density Estimation
Density-based Clustering
Hierarchical Clustering
Outlier Detection
Summary
References

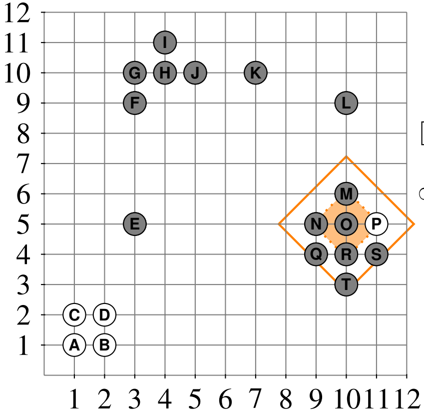


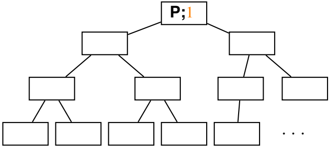


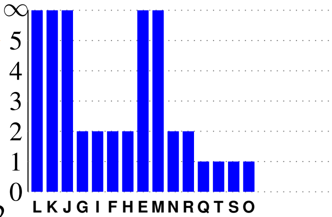


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example

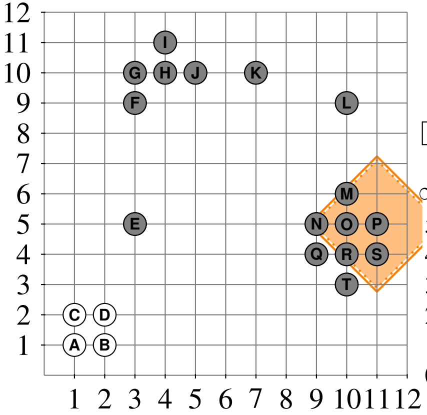


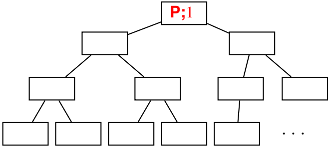


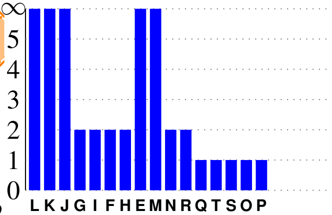


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example

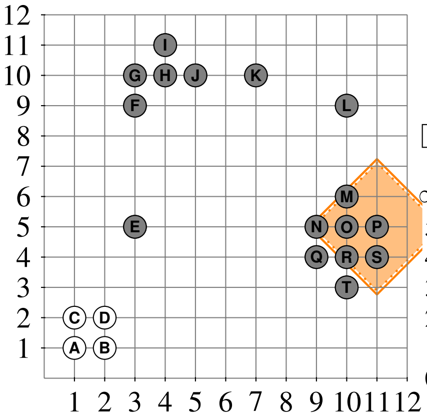


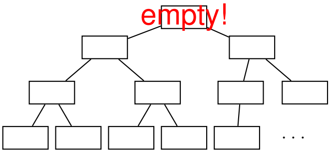


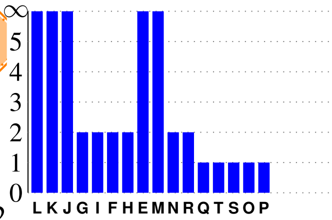


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example   Skip example

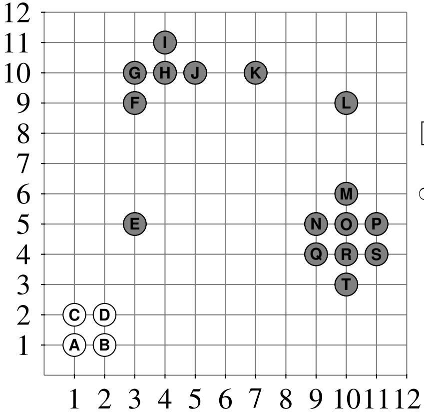


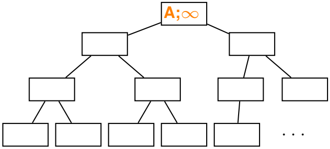


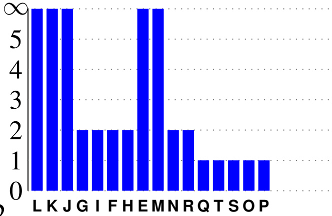


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example Skip example

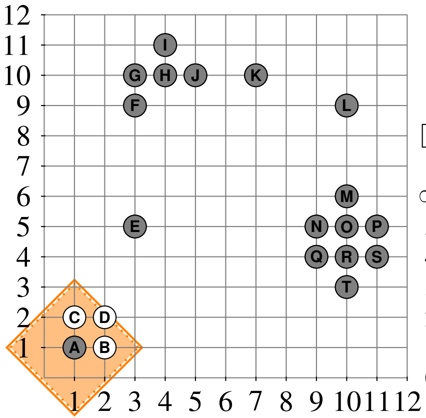


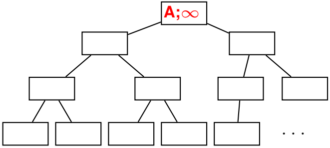


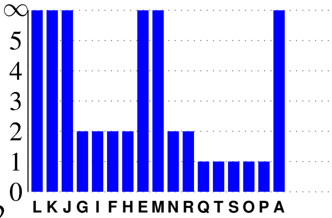


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example Skip example

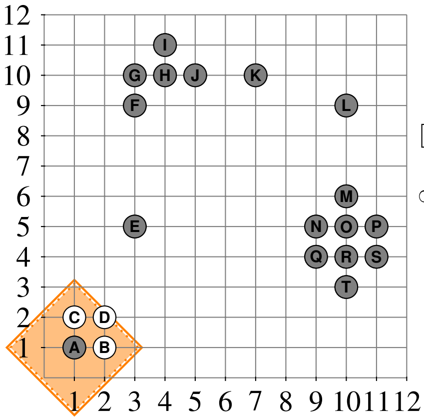


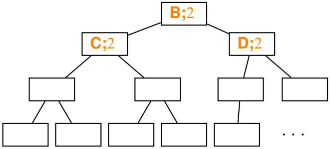


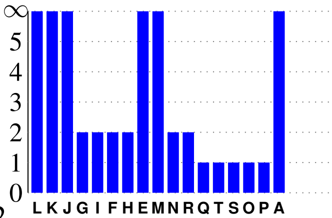


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example   Skip example

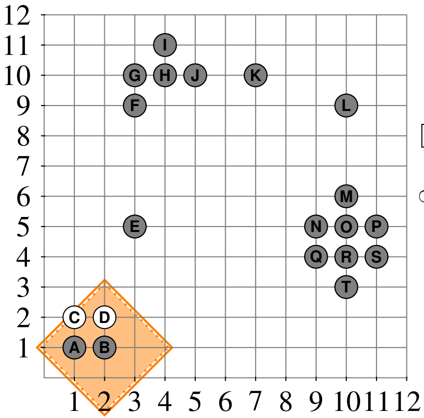


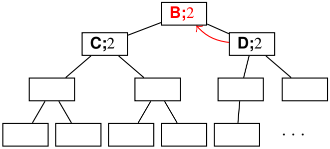


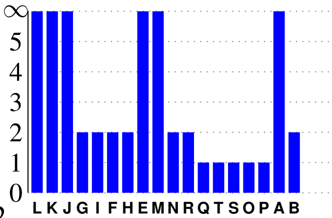


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example   Skip example

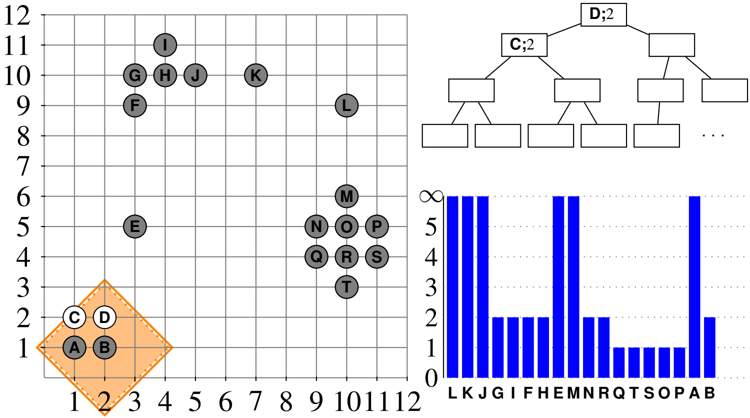






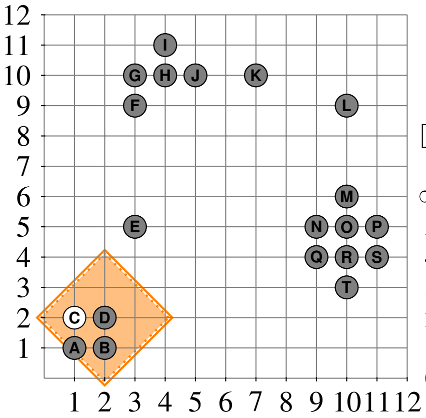
OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

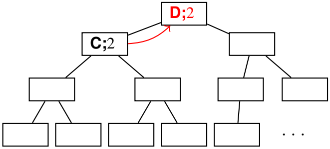
Start example
Skip example

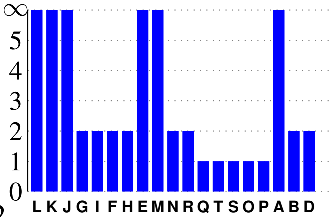


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example

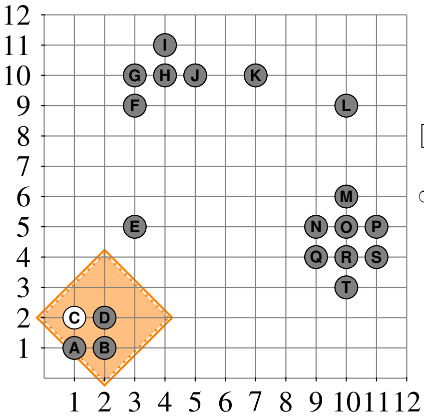


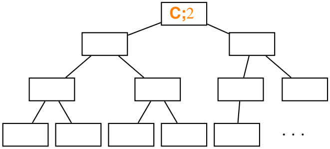


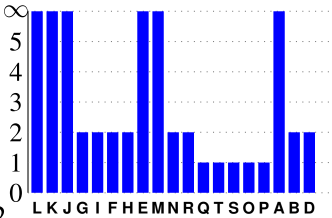


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example
Skip example

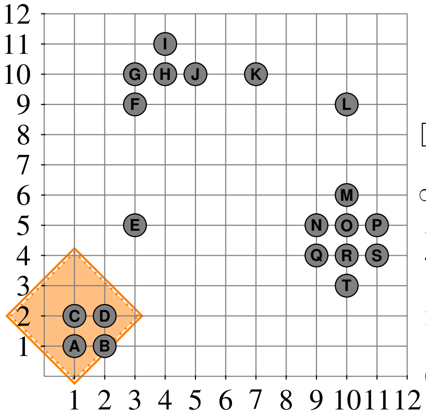


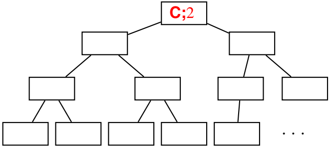


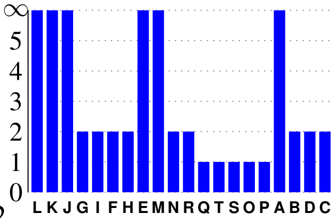


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example Skip example

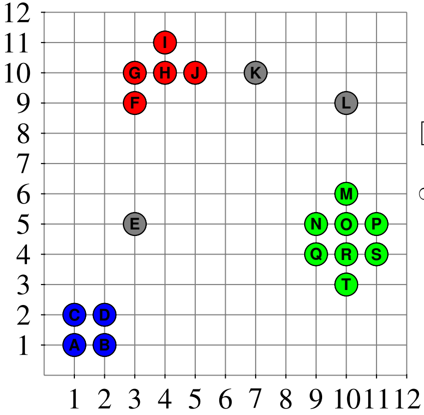


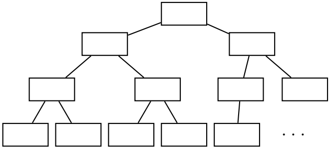


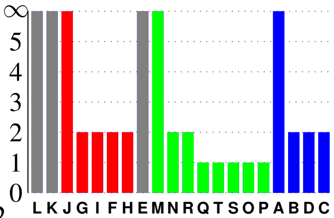


OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

▸ Start example ▸▸ Skip example





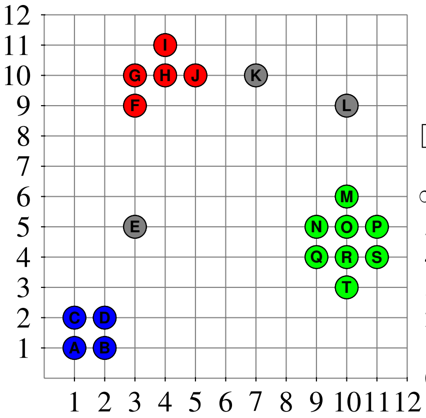


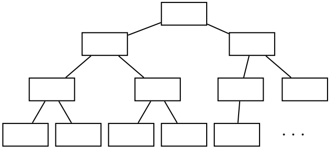
(Intuitively: cluster $\approx$ valley)

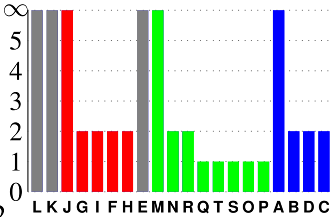
Previous “hill” would belong to the cluster, subsequent “hill” does not!

OPTICS – Example $\varepsilon = 2, \mathrm{MinPts} = 4$

Start example　Skip example

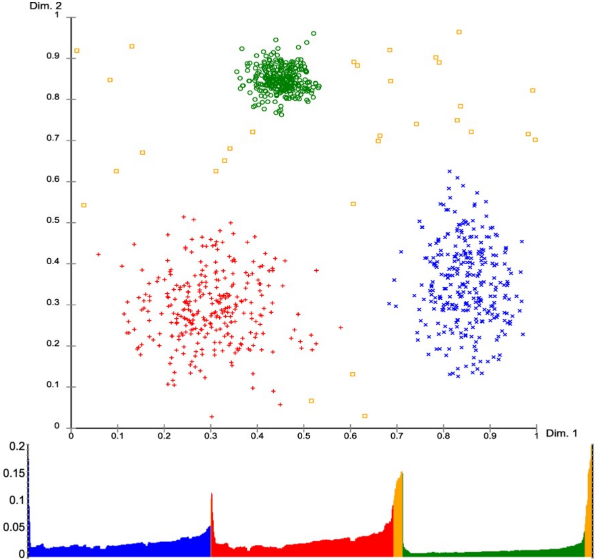






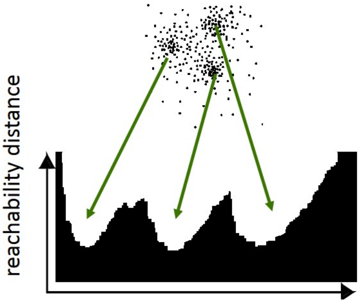
radius too small!

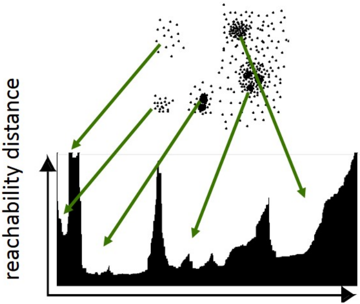
OPTICS Plot: Reachability Diagram



OPTICS Plot: Reachability Diagram

The cluster order (or: reachability diagram) shows the reachability distances w.r.t. ε and MinPts of the objects as bars the order (x-axis) given by the "cluster order"

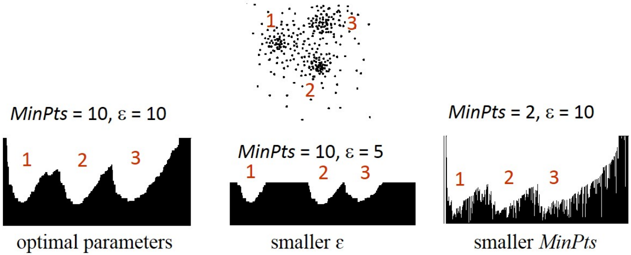




cut: $\approx$ density-level as ε for DBSCAN

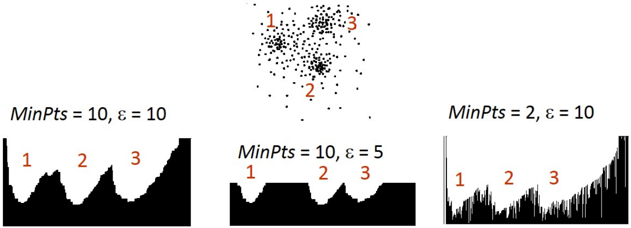
OPTICS Parameter Sensitivity

- cluster order relatively robust against parameter choices
- reasonable result when parameters are "big enough"



OPTICS Parameter Sensitivity

- cluster order relatively robust against parameter choices
- reasonable result when parameters are "big enough"



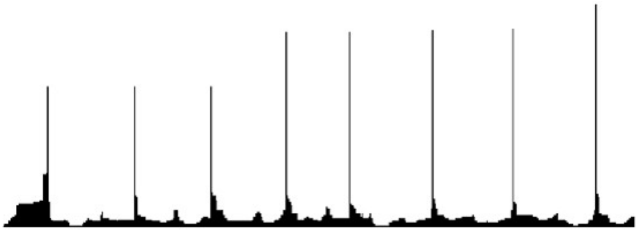
optimal parameters — smaller ε — smaller *MinPts*

Note that:

The meaning and effect of ε is quite different from the meaning of the equally named parameter in DBSCAN.

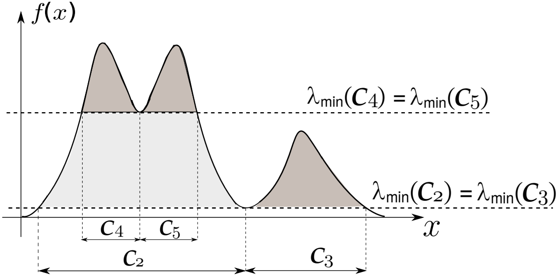
Manual Data Analysis with OPTICS

- are there clusters?
- how many?
- are the clusters hierarchically nested?
- how big are the clusters?



Cluster Extraction from Hierarchies

- Hierarchical DBSCAN* (HDBSCAN*) [Campello et al., 2013a, 2015]: formulates a criterion of cluster stability, optimizes overall quality of clusters extracted by *local* cuts
- framework for cluster extraction [Campello et al., 2013b]: finds the globally optimal *local* cut, based on additional criteria – HDBSCAN* becomes special case [Campello et al., 2015]



Videos

Recommended Reading:

A research group in Canada implemented HDBSCAN in scikit-learn [McInnes and Healy, 2017].
They talk about it in some conference presentations:

- *comparing discussion of various clustering methods and a demonstration of HDBSCAN**
  `https://www.youtube.com/watch?v=AgPQ76RIi6A`
- *from the same group, similar content, but a bit more context and technical explanation:*
  `https://www.youtube.com/watch?v=dGsxd67IFiU`

SDU SYDDANSK UNIVERSITET

Outline

Recommended Reading

Recommended Reading:

- *textbooks:*
  - *Tan et al. [2006], Chapter 10.*
  - *Tan et al. [2020], Chapter 9.*
  - *Han et al. [2011], Chapter 12.*
- *research articles:*
  - *General overview, interpretation, and categories: Zimek and Filzmoser [2018]*
  - *Advanced reading on locality and density-estimates: Schubert et al. [2014a], Schubert et al. [2014b].*
  - *On the problems of evaluation: Campos et al. [2016], Schubert et al. [2012].*
  - *Advanced reading on specializations to high-dimensional data: Zimek et al. [2012].*

Outline

Probabilistic Outlier Model: Statistical Tests

general idea

- given a certain kind of statistical distribution (e.g., Gaussian)
- compute the parameters assuming all data points have been generated by such a statistical distribution (e.g., mean and standard deviation)
- outliers are points that have a low probability to be generated by the overall distribution (e.g., deviate more than 3 times the standard deviation from the mean)

basic assumptions

- normal data objects follow a (known) distribution and occur in a high probability region of this model
- outliers deviate strongly from this distribution

Probabilistic Outlier Model: Statistical Tests

Many statistical tests are available in the literature, see classic textbooks (e.g., by Barnett [1978] or by Hawkins [1980]). Such tests differ in assumptions on:

- the type of data distribution (e.g., Gaussian)
- the number of variables, i.e., dimensions of the data objects (univariate/multivariate)
- the number of distributions (mixture models)
- parametric versus non-parametric (e.g., histogram-based)

example on the following slides:

- Gaussian distribution
- multivariate
- 1 model
- parametric

Probabilistic Outlier Model: Statistical Tests

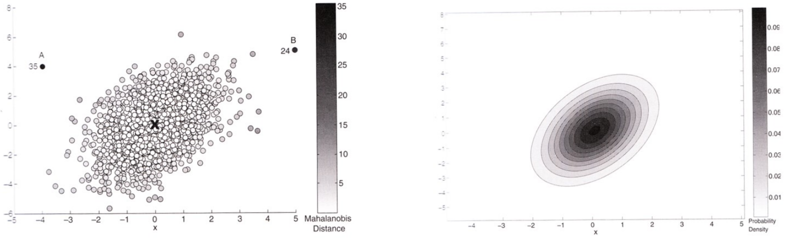
probability density function (multivariate normal distribution):

$$f(x) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} e^{-\frac{1}{2}(x-\mu)\cdot(\Sigma)^{-1}\cdot(x-\mu)^\mathsf{T}}$$

- μ is the mean value of all points (usually data are normalized such that $\mu = 0$)
- Σ is the covariance matrix centered at the mean
- MDist$= (x-\mu)\cdot(\Sigma)^{-1}\cdot(x-\mu)^\mathsf{T}$ is the Mahalanobis distance of point x to μ
- MDist follows a χ^2-distribution with d degrees of freedom ($d =$ data dimensionality)
- all points x, with MDist$(x,\mu) > \chi^2(0.975)$ $[\approx 3\cdot\sigma]$ are considered outliers

Visualization

Example (2D)



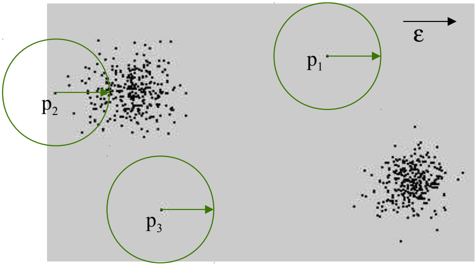
(Figure from Tan et al. [2006].)

Outline

Distance-based Outliers

DB(ε, π)-outlier [Knorr and Ng, 1997]

- given ε, π
- A point p is considered an outlier if at most π percent of all other points have a distance to p less than ε

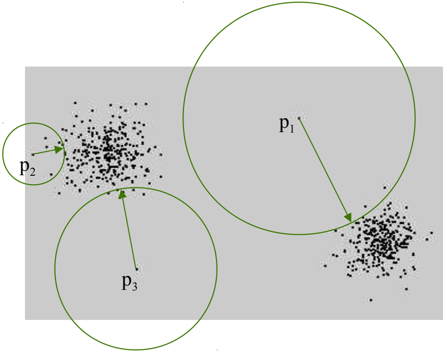


$$OutlierSet(\varepsilon, \pi) = \left\{ p \middle| \frac{Cardinality(q \in \mathcal{D} \mid \mathrm{dist}(q,p) < \varepsilon)}{Cardinality(\mathcal{D})} \leq \pi \right\}$$

Distance-based Outliers

Outlier scoring based on kNN distances:

- Take the kNN distance of a point as its outlier score [Ramaswamy et al., 2000]



- Variant: Aggregate the distances for the 1-NN, 2-NN, . . . , kNN (sum, average) [Angiulli and Pizzuti, 2002]

Choosing Parameters

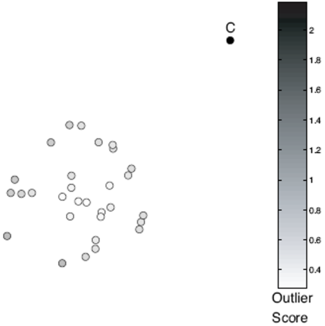


Figure 10.4. Outlier score based on the distance to fifth nearest neighbor.

(Figures from Tan et al. [2006].)

Choosing Parameters

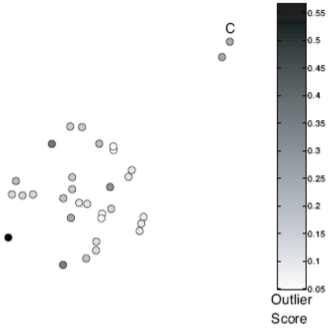


Figure 10.5. Outlier score based on the distance to the first nearest neighbor. Nearby outliers have low outlier scores.

(Figures from Tan et al. [2006].)

Choosing Parameters

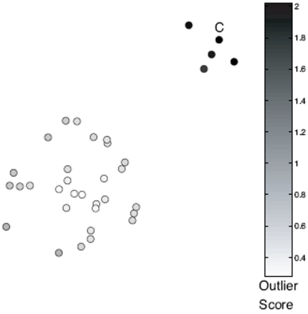


Figure 10.6. Outlier score based on distance to the fifth nearest neighbor. A small cluster becomes an outlier.

(Figures from Tan et al. [2006].)

Choosing Parameters

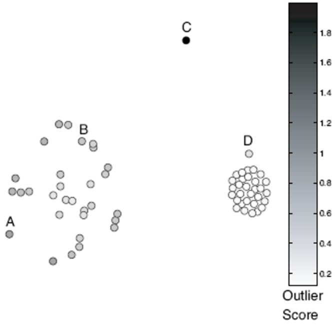


Figure 10.7. Outlier score based on the distance to the fifth nearest neighbor. Clusters of differing density.

(Figures from Tan et al. [2006].)

Density-based Local Outliers

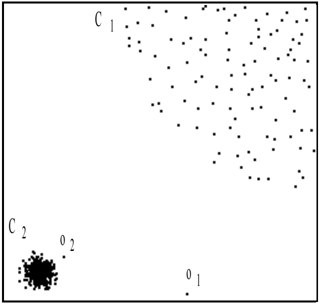


Figure from Breunig et al. [2000].

- DB-outlier model: no parameters ε, π such that o_2 is an outlier but none of the points of C_1 is an outlier
- kNN-outlier model: kNN-distances of points in C_1 are larger than kNN-distances of o_2

Density-based Local Outliers

Local Outlier Factor (LOF) [Breunig et al., 2000]:

- reachability distance (smoothing effect):
  $\mathrm{reachdist}_k(p,o) = \max\{k\,\mathrm{dist}(o), \mathrm{dist}(p,o)\}$
- local reachability density (lrd)
  $\mathrm{lrd}_k(p) = 1/\frac{\sum_{o\in kNN(p)} \mathrm{reachdist}_k(p,o)}{Cardinality(kNN(p))}$
- Local outlier factor (LOF) of point p: average ratio of lrds of neighbors of p and lrd of p

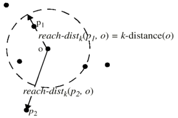


Figure from Breunig et al. [2000]

$$LOF_k(p) = \frac{\sum_{o\in kNN(p)} \frac{\mathrm{lrd}_k(o)}{\mathrm{lrd}_k(p)}}{Cardinality(kNN(p))}$$

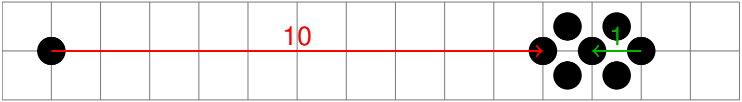
- LOF ≈ 1: homogeneous density
- LOF $\gg 1$: point is an outlier (meaning of "$\gg$" ?)

Outline

Global vs. Local Outliers

Distance based outliers:
"Outliers are farther away from the remainder of the data than most other objects."

- Distance to nearest neighbor



Global vs. Local Outliers

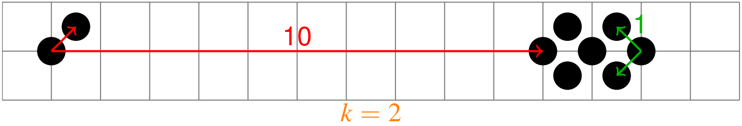
Distance based outliers:
"Outliers are farther away from the remainder of the data than most other objects."

- Distance to nearest neighbor
  $\Rightarrow$ misses paired outliers

Global vs. Local Outliers

Distance based outliers:
"Outliers are farther away from the remainder of the data than most other objects."

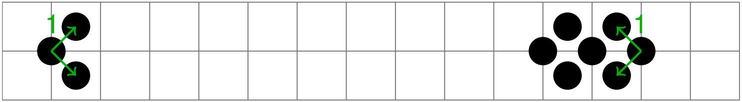
- Distance to k nearest neighbor



Global vs. Local Outliers

Distance based outliers:
"Outliers are farther away from the remainder of the data than most other objects."

- Distance to k nearest neighbor

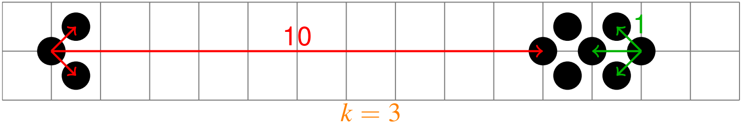


$k = 2$

Global vs. Local Outliers

Distance based outliers:
"Outliers are farther away from the remainder of the data than most other objects."

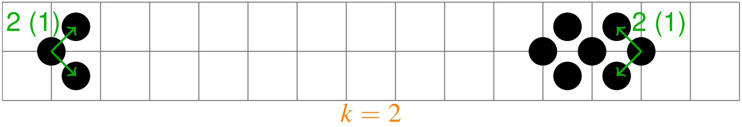
- Distance to k nearest neighbor
  $\Rightarrow$ micro clusters ($|C| < k+1$) become outliers



Global vs. Local Outliers

Distance based outliers:
"Outliers are farther away from the remainder of the data than most other objects."

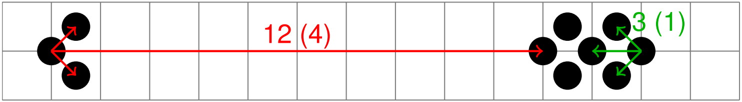
- Sum (average) of distances to the first k nearest neighbors
  More robust with respect to k and micro clusters



Global vs. Local Outliers

Distance based outliers:
“Outliers are farther away from the remainder of the data than most other objects.”

- Sum (average) of distances to the first k nearest neighbors

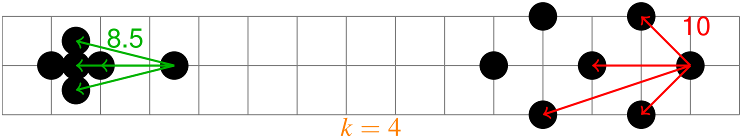


$k = 3$

Global vs. Local Outliers

Distance based outliers:
"Outliers are farther away from the remainder of the data than most other objects."

- Sum (average) of distances to the first k nearest neighbors



Local Density Estimate, Global Comparison

Note that:

- *Both*
  - *the k nearest neighbor distance*
  - *and the average k nearest neighbor distance*

  are variants of local density estimates.
- *All local density estimates are ranked, the lowest density estimate is the strongest outlier.*
- *We cannot handle areas of different densities, because all density estimates are compared globally, against all others.*

Global vs. Local Outliers

Local outlier factor:

Idea:

Outliers exhibit a lower local density than their neighbors.

Global vs. Local Outliers

Local outlier factor:

Idea:

Outliers exhibit a lower local density than their neighbors.

Local Outlier Factor

$$LOF(p) = \underbrace{\frac{1}{|\mathcal{N}|}\sum_{o\in\mathcal{N}}}_{\text{average}} \underbrace{\frac{lrd_k(o)}{lrd_k(p)}}_{\text{relative density}}$$

- Same density $\Leftrightarrow$ relative density $= 1$
- Less dense $\Leftrightarrow$ relative density > 1
- $LOF(p) \gg 1$ for outliers!

Outline

Some Typical Local Outlier Detection Methods

Example: Local Outlier Factor [Breunig et al., 2000]

- Compute neighborhoods of points:

$$\mathcal{N}(p) = k \text{ nearest neighbors of } p$$

- Compute for each point:

$$\mathsf{lrd}(p) = \frac{1}{\mathsf{mean}_{o \in \mathcal{N}(p)} \, \mathsf{reach\text{-}dist}(p, o)}$$

- Compare 'density' of each point to the neighbor 'densities':

$$\mathsf{LOF}(p) = \mathsf{mean}_{o \in \mathcal{N}(p)} \frac{\mathsf{lrd}(o)}{\mathsf{lrd}(p)}$$

Some Typical Local Outlier Detection Methods

Example: SimplifiedLOF [Schubert et al., 2014a]

- Compute neighborhoods of points:

$$\mathcal{N}(p) = k \text{ nearest neighbors of } p$$

- Compute for each point:

$$\text{dens}(p) = \frac{1}{k\text{-dist}(p)}$$

- Compare ‘density’ of each point to the neighbor ‘densities’:

$$\text{SimplifiedLOF}(p) = \text{mean}_{o \in \mathcal{N}(p)} \frac{\text{dens}(o)}{\text{dens}(p)}$$

Some Typical Local Outlier Detection Methods

Example: Influenced Outlierness [Jin et al., 2006]

- Compute neighborhoods of points:

$$\mathcal{N}(p) = kNN(p) \cap RkNN(p)$$

- Compute for each point:

$$\text{dens}(p) = \frac{1}{k\text{-dist}(p)}$$

- Compare ‘density’ of each point to the neighbor ‘densities’:

$$\text{INFLO}(p) = \text{mean}_{o \in \mathcal{N}(p)} \frac{\text{dens}(o)}{\text{dens}(p)}$$

Outline

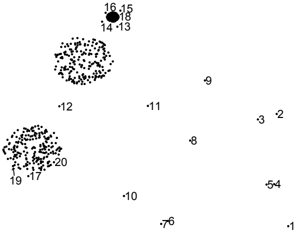
Outlier vs. Inlier – a Classification Problem?

- class imbalance: the “class” outlier is substantially smaller, but often more important than the “class” inlier
- resulting problems:
  - training of a classifier is difficult due to lack of training data
  - evaluation measures are hard to interpret (e.g., typically very low precision values even for a rather good result)
- most (unsupervised) outlier detection methods deliver not a decision but a score (“outlier score”, “outlier factor”)
- outlier detection result based on scores is a ranking

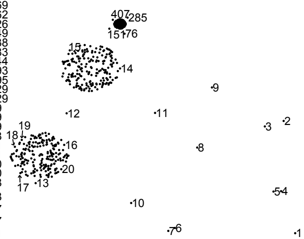
Evaluation of Rankings

- methods return a full ranking of database objects
- user interested in the top-ranked objects

Rank	Score
1	0.996
2	0.995
3	0.994
4	0.994
5	0.994
6	0.993
7	0.993
8	0.992
9	0.991
10	0.989
11	0.985
12	0.974
13	0.891
14	0.877
15	0.854
16	0.803
17	0.680
18	0.573
19	0.568
20	0.562
⋮	⋮
615	0.000



Rank	Score
1	56.069
2	47.862
3	43.726
4	43.449
5	41.938
6	37.283
7	36.444
8	32.193
9	27.095
10	23.429
11	16.629
12	9.339
13	4.029
14	3.369
15	3.208
16	3.091
17	3.091
18	3.020
19	3.020
20	2.988
⋮	⋮
76	2.268
⋮	⋮
151	2.017
⋮	⋮
285	1.687
⋮	⋮
407	1.251
⋮	⋮
615	0.171



examples taken from Campello et al. [2015]

Evaluation of Rankings

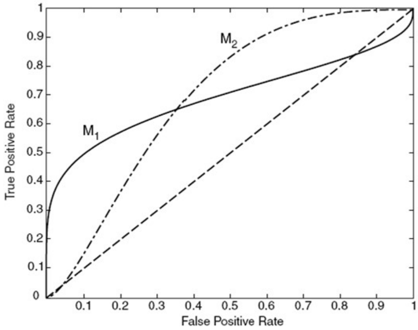
- two-class confusion matrix ($C(o)$: class, $P(o)$: prediction):

$$\begin{array}{rcc} C(o)\backslash P(o) & outlier & inlier \\ outlier & TP & FN \\ inlier & FP & TN \end{array}$$

- ranking is a continuum between strongest outlier and weakest outlier (i.e., strongest inlier): we consider only the first column (everything is predicted being an outlier, but in an order as defined by the ranking)
- ideal ranking: all TP before any FP

ROC: Receiver Operating Characteristic

- each TP in the ranking: one step up
- each FP in the ranking: one step to the right
- comparison of two rankings: area under the curve (ROC AUC)



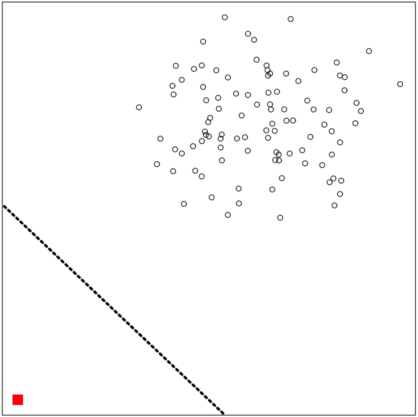
- $0 \leq$ ROC AUC ≤ 1
- interpretation: probability that two randomly chosen objects, one positive example (outlier) and one negative example (inlier), have a correct relative ranking (with the outlier ranked before the inlier) [Hanley and McNeil, 1982]

External vs. Internal Evaluation

- Evaluation with ROC needs class labels (external information).
- This kind of evaluation is therefore called *external evaluation*.
- In an application of an unsupervised method, we would actually not have external information available.
- We need *internal evaluation*, relying only on the data itself.

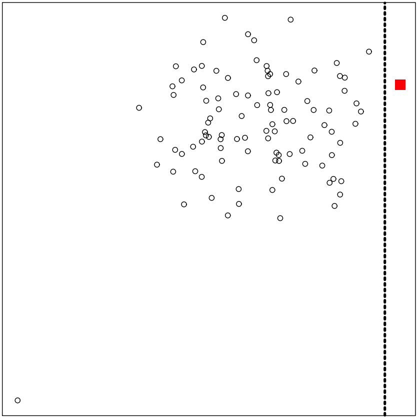
Internal Evaluation

Marques et al. [2015] proposed the first internal evaluation measure, based on a notion of classification hardness.



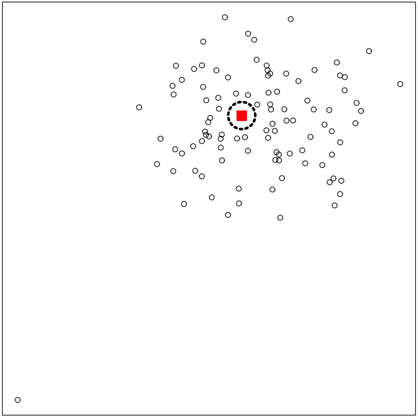
Internal Evaluation

Marques et al. [2015] proposed the first internal evaluation measure, based on a notion of classification hardness.



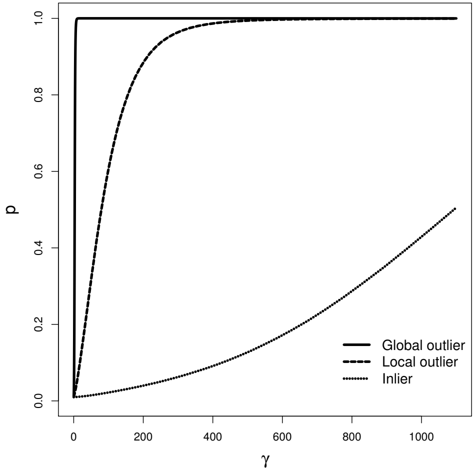
Internal Evaluation

Marques et al. [2015] proposed the first internal evaluation measure, based on a notion of classification hardness.



Internal Evaluation

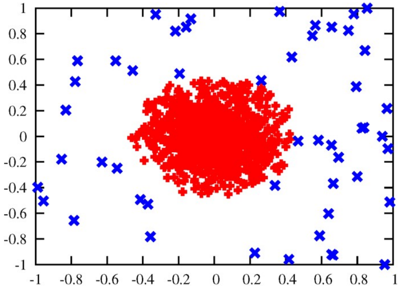
Marques et al. [2015] proposed the first internal evaluation measure, based on a notion of classification hardness.



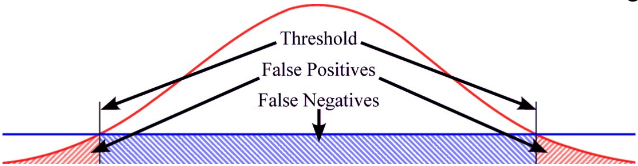
Outline

Unavoidable False Positives and False Negatives

- objects truly generated by a different mechanism can fit very well to the "normal" data distribution
- inliers in the tail of the "normal" distribution must appear suspicious

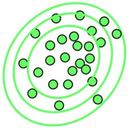


decision on true outlierness based on domain knowledge



Masking and Swamping

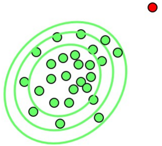
- mean and standard deviation are sensitive to outliers themselves
- these values are computed for the complete data (including potential outliers) – masking and swamping:



- the MDist is used to determine outliers although the MDist values are influenced by these outliers (advanced approaches try to alleviate this effect)
- applies to any model (some are more robust, some are more susceptible to these effects)
- thus statistical research on “robust methods” [Rousseeuw and Hubert, 2011]

Masking and Swamping

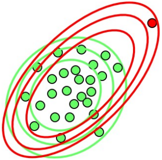
- mean and standard deviation are sensitive to outliers themselves
- these values are computed for the complete data (including potential outliers) – masking and swamping:



- the MDist is used to determine outliers although the MDist values are influenced by these outliers (advanced approaches try to alleviate this effect)
- applies to any model (some are more robust, some are more susceptible to these effects)
- thus statistical research on “robust methods” [Rousseeuw and Hubert, 2011]

Masking and Swamping

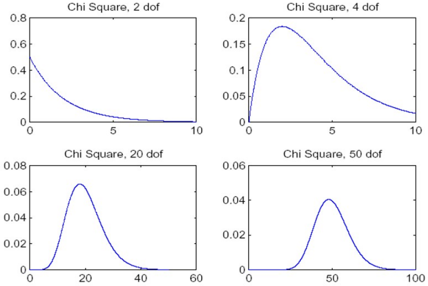
- mean and standard deviation are sensitive to outliers themselves
- these values are computed for the complete data (including potential outliers) – masking and swamping:



- the MDist is used to determine outliers although the MDist values are influenced by these outliers (advanced approaches try to alleviate this effect)
- applies to any model (some are more robust, some are more susceptible to these effects)
- thus statistical research on "robust methods" [Rousseeuw and Hubert, 2011]

Curse of Dimensionality

- curse of dimensionality: the larger the degree of freedom, the more similar the MDist values are for all points



x-axis: observed *MDist* values
y-axis: frequency of observation

Outline

Summary

You learned in this section:

- *discrete random variables*
- *expectation, variance, standard deviation, covariance, correlation*
- *continuous distributions (uniform and normal) and their parameters*
- *Bayesian learning with distributions*
- *algebraic vs. probabilistic view on data*
  - *EM clustering and a probabilistic interpretation of k-means*
  - *LDA*

Summary

You learned in this section:

- *parametric vs. non parametric learning*
- *non-parametric density estimation*
- *basic ideas of density-based clustering and the algorithm DBSCAN*
  - *core point*
  - *(direct) density-reachability*
  - *density-connectivity*
  - *border points*
  - *noise*
- *variants*
  - *include border points?*
  - *SNN-based clustering*
  - *GDBSCAN*

Summary

You learned in this section:

- *dendrogram as representation of a hierarchical clustering solution*
- *distances for clusters and objects and the basic agglomerative greedy algorithm*
- *density-based hierarchical clustering*
  - *formalization*
  - *algorithm OPTICS*
  - *cluster order*
    - *construction*
    - *interpretation*
    - *parameters*

Summary

You learned in this section:

- *probabilistic parametric model for outlier detection*
- *non-parametric models:*
  - *distance-based models*
    - *DB-outlier model*
    - k*NN-based model*
  - *density-based model*
    - *LOF: motivation, model*
    - *pointers to the literature: many variants of LOF*
- *evaluation of outlier detection*
- *problems and challenges*

References I

C. C. Aggarwal and J. Han, editors. *Frequent Pattern Mining*. Springer, 2014.

C. C. Aggarwal and C. K. Reddy, editors. *Data Clustering: Algorithms and Applications*. CRC Press, 2013.

F. Angiulli and C. Pizzuti. Fast outlier detection in high dimensional spaces. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD), Helsinki, Finland*, pages 15–26, 2002. doi: 10.1007/3-540-45681-3_2.

M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Philadelphia, PA*, pages 49–60, 1999. doi: 10.1145/304182.304187.

D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), New Orleans, LA*, pages 1027–1035, 2007. doi: 10.1145/1283383.1283494.

V. Barnett. The study of outliers: Purpose and model. *Applied Statistics*, 27(3): 242–250, 1978.

References II

T. Bernecker, M. E. Houle, H.-P. Kriegel, P. Kröger, M. Renz, E. Schubert, and A. Zimek. Quality of similarity rankings in time series. In *Proceedings of the 12th International Symposium on Spatial and Temporal Databases (SSTD), Minneapolis, MN*, pages 422–440, 2011. doi: 10.1007/978-3-642-22922-0_25.

F. Birzele and S. Kramer. A new representation for protein secondary structure prediction based on frequent patterns. *Bioinformatics*, 22(21):2628–2634, 2006. doi: 10.1093/bioinformatics/btl453.

M. M. Breunig, H.-P. Kriegel, R.T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Dallas, TX*, pages 93–104, 2000. doi: 10.1145/342009.335388.

R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Gold Coast, Australia*, pages 160–172. 2013a. doi: 10.1007/978-3-642-37456-2_14.

R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander. A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. *Data Mining and Knowledge Discovery*, 27(3):344–371, 2013b. doi: 10.1007/s10618-013-0311-4.

References III

R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):5:1–51, 2015. doi: 10.1145/2733381.

R. J. G. B. Campello, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2), 2020.

G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle. On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30:891–927, 2016. doi: 10.1007/s10618-015-0444-8.

D. Defays. An efficient algorithm for the complete link cluster method. *The Computer Journal*, 20(4):364–366, 1977. doi: 10.1093/comjnl/20.4.364.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(1):1–31, 1977.

J. L. Devore and K. N. Berk. *Modern Mathematical Statistics with Applications*. Springer, 2nd edition, 2012.

References IV

M. M. Deza and E. Deza. *Encyclopedia of Distances*. Springer, 3rd edition, 2009. ISBN 9783662443415.

P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *Proceedings of the 13th International Conference on Machine Learning (ICML), Bari, Italy*, pages 105–112, 1996.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley&Sons, 2nd edition, 2001.

C. Elkan. Using the triangle inequality to accelerate k-means. In *Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, DC*, pages 147–153, 2003.

L. Ertöz, M. Steinbach, and V. Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the 3rd SIAM International Conference on Data Mining (SDM), San Francisco, CA*, 2003. doi: 10.1137/1.9781611972733.5.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR*, pages 226–231, 1996.

References V

U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge discovery and data mining: Towards a unifying framework. In *Proceedings of the 2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR*, pages 82–88, 1996.

U. M. Fayyad, C. Reina, and P. S. Bradley. Initialization of iterative refinement clustering algorithms. In *Proceedings of the 4th ACM International Conference on Knowledge Discovery and Data Mining (KDD), New York City, NY*, pages 194–198, 1998.

E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–769, 1965.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *ACM SIGMOD Record*, 29(2):1–12, 2000. doi: 10.1145/342009.335372.

J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3rd edition, 2011.

J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143:29–36, 1982.

J. A. Hartigan. *Clustering Algorithms*. John Wiley&Sons, New York, London, Sydney, Toronto, 1975.

References VI

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. Springer, 2001.

D. Hawkins. *Identification of Outliers*. Chapman and Hall, 1980.

M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Can shared-neighbor distances defeat the curse of dimensionality? In *Proceedings of the 22nd International Conference on Scientific and Statistical Database Management (SSDBM), Heidelberg, Germany*, pages 482–500, 2010. doi: 10.1007/978-3-642-13818-8_34.

Z. Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. In *Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD), Tucson, AZ*, 1997.

R. A. Jarvis and E. A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, C-22(11):1025–1034, 1973.

W. Jin, A. K. H. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Singapore*, pages 577–593, 2006. doi: 10.1007/11731139_68.

References VII

E. M. Knorr and R. T. Ng. A unified notion of outliers: Properties and computation. In *Proceedings of the 3rd ACM International Conference on Knowledge Discovery and Data Mining (KDD), Newport Beach, CA*, pages 219–222, 1997. doi: 10.1145/782010.782021.

H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011. doi: 10.1002/widm.30.

H.-P. Kriegel, E. Schubert, and A. Zimek. The (black) art of runtime evaluation: Are we comparing algorithms or implementations? *Knowledge and Information Systems (KAIS)*, 52(2):341–378, 2017. doi: 10.1007/s10115-016-1004-2.

S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982. doi: 10.1109/TIT.1982.1056489.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematics, Statistics, and Probabilistics*, volume 1, pages 281–297, 1967.

H. O. Marques, R. J. G. B. Campello, A. Zimek, and J. Sander. On the internal evaluation of unsupervised outlier detection. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management (SSDBM), San Diego, CA*, pages 7:1–12, 2015. doi: 10.1145/2791347.2791352.

References VIII

L. McInnes and J. Healy. Accelerated hierarchical density based clustering. In *ICDM Workshops*, pages 33–42, 2017.

T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

M. Mitzenmacher and E. Upfal. *Probability and Computing. Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, 2nd edition, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Dallas, TX*, pages 427–438, 2000. doi: 10.1145/342009.335437.

P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

References IX

P. J. Rousseeuw and M. Hubert. Robust statistics for outlier detection. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):73–79, 2011. doi: 10.1002/widm.2.

J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998. doi: 10.1023/A:1009745219419.

E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel. On evaluation of outlier rankings and outlier scores. In *Proceedings of the 12th SIAM International Conference on Data Mining (SDM), Anaheim, CA*, pages 1047–1058, 2012. doi: 10.1137/1.9781611972825.90.

E. Schubert, A. Zimek, and H.-P. Kriegel. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery*, 28(1):190–237, 2014a. doi: 10.1007/s10618-012-0300-z.

E. Schubert, A. Zimek, and H.-P. Kriegel. Generalized outlier detection with flexible kernel density estimates. In *Proceedings of the 14th SIAM International Conference on Data Mining (SDM), Philadelphia, PA*, pages 542–550, 2014b. doi: 10.1137/1.9781611973440.63.

References X

R. Sibson. SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973. doi: 10.1093/comjnl/16.1.30.

R. Silberzahn and E. L. Uhlmann. Many hands make tight work. *Nature*, 526: 189–191, 2015. doi: 10.1038/526189a.

B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman & Hall/CRC, 1986.

R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Montreal, QC, Canada*, pages 1–12, 1996. doi: 10.1145/233269.233311.

N. N. Taleb. *Fooled by Randomness: The Hidden Role of Chance in Life and in the Markets*. Random House, 2001.

N. N. Taleb. *The Black Swan. The Impact of the Highly Improbable*. Random House, 2007.

P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.

P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2nd edition, 2020.

References XI

L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka. Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, 3(4): 209–235, 2010. doi: 10.1002/sam.10080.

J. Vreeken and N. Tatti. Interesting patterns. In C. C. Aggarwal and J. Han, editors, *Frequent Pattern Mining*, chapter 5, pages 105–134. Springer, 2014. doi: 10.1007/978-3-319-07821-2_5.

I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 3rd edition, 2011.

M. J. Zaki and W. Meira Jr. *Data Mining and Analysis. Fundamental Concepts and Algorithms*. Cambridge University Press, 2014.

M. J. Zaki and W. Meira Jr. *Data Mining and Analysis. Fundamental Concepts and Algorithms*. Cambridge University Press, 2nd edition, 2020.

A. Zimek and P. Filzmoser. There and back again: Outlier detection between statistical reasoning and data mining algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1280, 2018. doi: 10.1002/widm.1280.

A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5): 363–387, 2012. doi: 10.1002/sam.11161.

References XII

A. Zimek, I. Assent, and J. Vreeken. Frequent pattern mining algorithms for data clustering. In C. C. Aggarwal and J. Han, editors, *Frequent Pattern Mining*, chapter 16, pages 403–423. Springer, 2014. doi: 10.1007/978-3-319-07821-2_16.