

Clustering

Fabrice Rossi

CEREMADE
Université Paris Dauphine

2021

Introduction

Hierarchical clustering

K-means and related methods

DBSCAN

Fuzzy and probabilistic models

Setting

- ▶ $\mathcal{D} = ((\mathbf{X}_i)_{1 \leq i \leq N})$
- ▶ no target value!
- ▶ goal: “understanding” the data
- ▶ in practice, many concrete goals such as
 - ▶ finding clusters
 - ▶ finding frequent patterns
 - ▶ finding outliers
 - ▶ modeling the data distribution
 - ▶ etc.

Setting

- ▶ $\mathcal{D} = ((\mathbf{X}_i)_{1 \leq i \leq N})$
- ▶ no target value!
- ▶ goal: “understanding” the data
- ▶ in practice, many concrete goals such as
 - ▶ finding clusters
 - ▶ finding frequent patterns
 - ▶ finding outliers
 - ▶ modeling the data distribution
 - ▶ etc.

Definition

In a data set, a *cluster* is a group of objects that are more similar to each other than to objects from the rest of the data set.

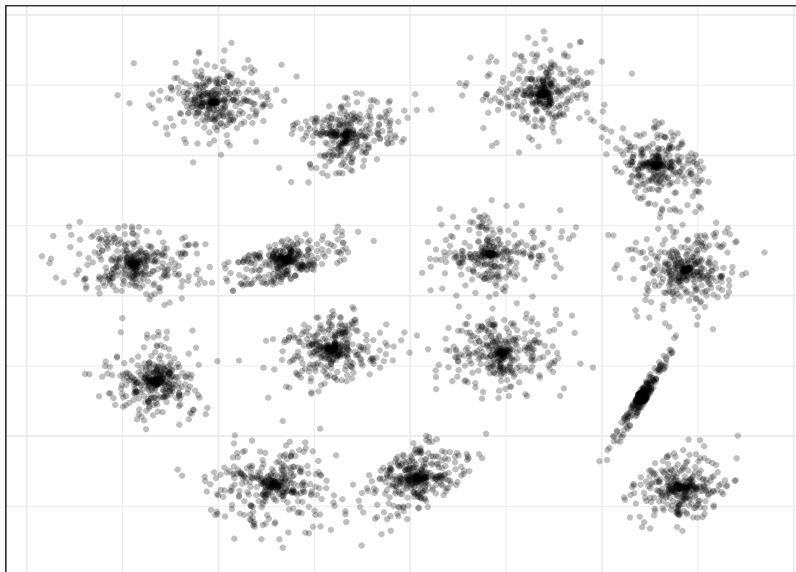
Definition

In a data set, a *cluster* is a **group** of objects that are **more similar** to each other **than** to objects from the rest of the data set.

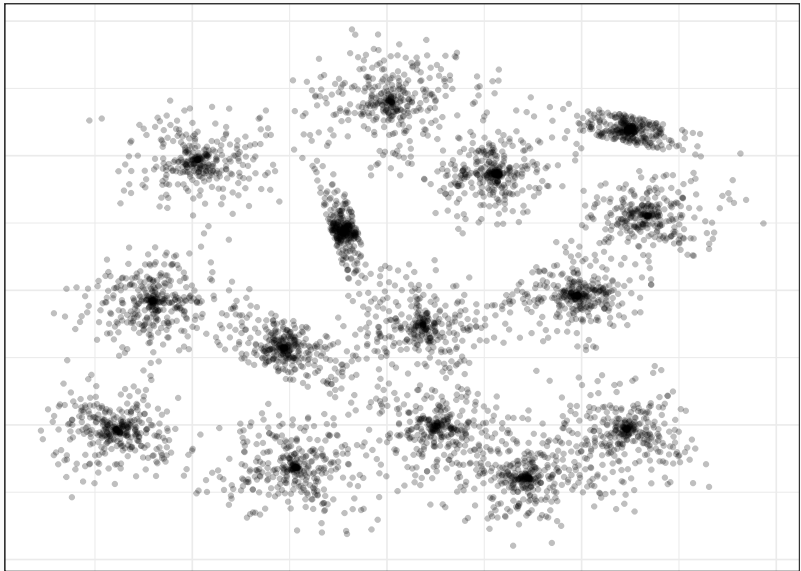
Difficulties

- ▶ group?
- ▶ similar?
- ▶ more than?

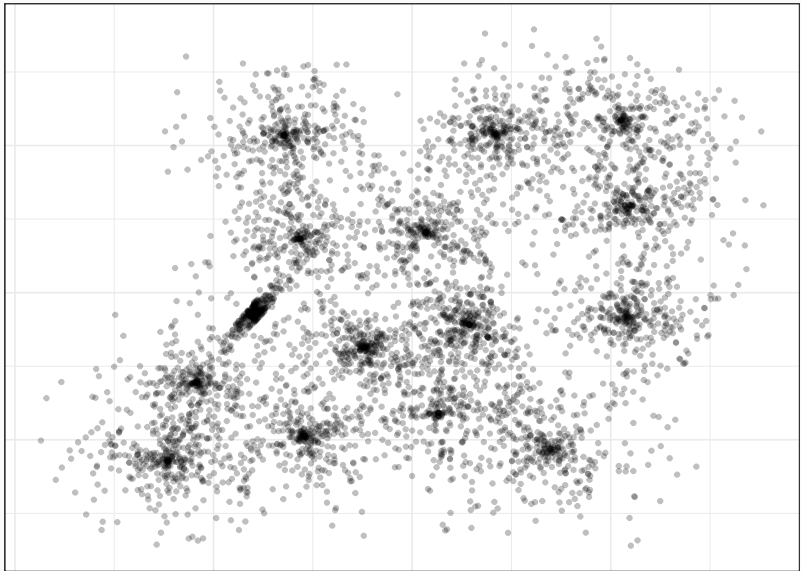
Non obvious task



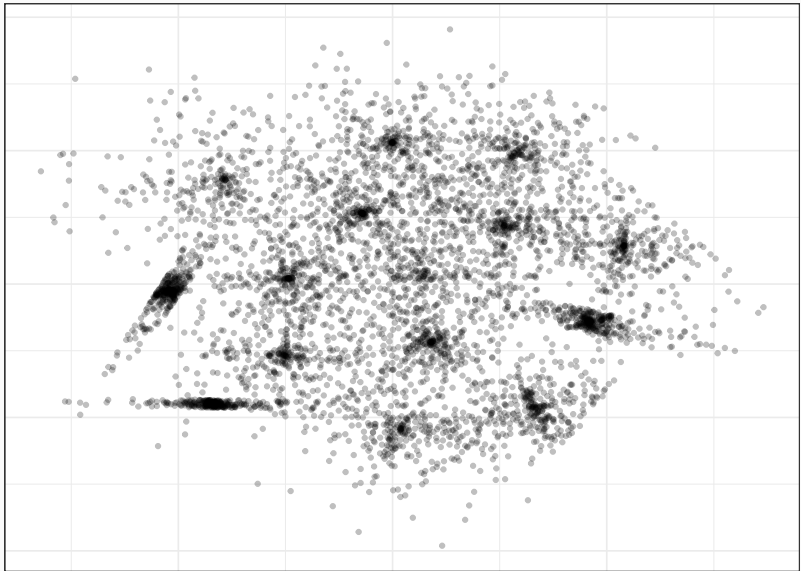
Non obvious task



Non obvious task



Non obvious task



Simplification (pre-processing)

- ▶ large scale data analysis: replace a group of objects by a single typical object
- ▶ coarse grain analysis before a finer grain one (see e.g. [Yippy](#))

Simplification (pre-processing)

- ▶ large scale data analysis: replace a group of objects by a single typical object
- ▶ coarse grain analysis before a finer grain one (see e.g. [Yippy](#))

Knowledge discovery

- ▶ consumer analysis
- ▶ image analysis (zone extraction)
- ▶ evolutionary biology
- ▶ etc.

Yippy example

The screenshot shows a web browser window with the Yippy search engine interface. The search bar contains the text "machine learning" and a "Search" button. Below the search bar, there are navigation links for "Sources", "Sites", "Time", and "Topics". A list of sources is displayed, including "Top 500 Results", "Language, Natural (55)", "Management (41)", "Marketing (40)", "Neural networks (49)", "Image (26)", "Security, Cyber (21)", "Microsoft, Azure Machine Learning (14)", "Digital (21)", "Reviews (21)", "Developers (11)", "Paper (12)", "Library (12)", "Without being explicitly programmed (3)", "Machine Learning, And Scientific Data (10)", "Reasoning (12)", "Twitter (5)", "Face-swap (3)", "Support vector machines (8)", "Machine Learning Journal (7)", "Conference on Machine Learning (11)", "Download (6)", "Tutorials (8)", "Magazine (8)", "Planets (8)", "Arvys, Marketwired (3)", "Machine Learning Platform (5)", "Funds (4)", "Nvidia, Competition (4)", "Advertising (10)", and "YouTube (3)".

Below the sources list, there are several search results for "machine learning":

- Machine learning - Wikipedia** [new window](#) [review](#)
Machine learning is a field of computer science that gives computer systems the ability to "learn" (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed.
https://en.wikipedia.org/wiki/Machine_learning - - Yippy Index V
- Machine Learning | Microsoft Azure** [new window](#) [review](#)
Get started now with Azure **Machine Learning** for powerful cloud-based analytics, now part of Cortana Intelligence Suite.
<https://azure.microsoft.com/en-us/services/machine-learning-studio> - - Yippy Index V
- Machine Learning | Coursera** [new window](#) [review](#)
Machine Learning from Stanford University. **Machine learning** is the science of getting computers to act without being explicitly programmed. In the past decade, **machine learning** has given us self-driving cars, practical speech recognition, ...
<https://www.coursera.org/learn/machine-learning> - - Yippy Index V
- Machine Learning by Tom Mitchell - amazon.com** [new window](#) [review](#)
Machine Learning [Tom M. Mitchell] on Amazon.com. "FREE" shipping on qualifying offers. This book covers the field of **machine learning**, which is the study of algorithms that allow computer programs to automatically improve through experience.
<https://www.amazon.com/Machine-Learning-Tom-M-Mitchell/dp/0070428077> - - Yippy Index V
- Machine Learning: What it is and why it matters | SAS** [new window](#) [review](#)
Find out what **machine learning** is, what kinds of algorithms and processes are used, and some of the many ways that **machine learning** is being used today.
https://www.sas.com/en_us/insights/analytics/machine-learning.html - - Yippy Index V
- VantageScore White Paper Explains VantageScore 4.0's Use of Machine Learning** [new window](#) [review](#)
Date: December 07, 2017 09:00 ET
STAMFORD, CN—(Marketwired - December 07, 2017) - VantageScore Solutions, LLC, developer of the VantageScore credit score model, released a new white paper that showcases the benefits of using **machine learning** to score more people with greater accuracy. The white paper, "Scoring Credit Invisibles: Using **machine learning** techniques to score consumers with ..."
www.marketwired.com - - use of machine-learning-2242858.htm - [cache](#) - Yippy News
- Google launches new Pixel 2 and Pixel 2 XL smartphones - video** [new window](#) [review](#)
Date: 2017-10-04T22:08:53.000Z, 2017-10-04T22:08:53.000Z
... Apple, Google's new high-end smartphones have 64GB of storage, front-facing speakers and 12-megapixel cameras supported by **machine learning** ... Apple, Google's new high-end smartphones have 64GB of storage, front-facing speakers and 12-megapixel cameras supported by **machine learning**
<https://www.theguardian.com/.../el-2-and-pixel-2-xl-smartphones-video> - [cache](#) - Yippy News
- What is machine learning? - Definition from WhatsIt.com** [new window](#) [review](#)
Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.
<https://www.techtarget.com/definition/machine-learning> - - Yippy Index V
- Deepfakes: The face-swapping software explained** [new window](#) [review](#)
www.bbc.co.uk/.../takes-the-face-swapping-software-explained - - Yippy News
- Novoheart Publishes Landmark Study Demonstrating the Use of Machine Learning to Accelerate Drug Screening** [new window](#) [review](#)
Date: December 05, 2017 07:30 ET
VANCOUVER, BRITISH COLUMBIA—(Marketwired - Dec. 5, 2017) - Novoheart ("Novoheart" or the "Company") (TSX VENTURE:NVH), a global stem cell biotechnology company, is pleased to announce the publication of its study in Stem Cell Reports, the official journal of the International Society for Stem Cell Research. The study demonstrates how **machine** ...
www.marketwired.com/.../rate-drug-bis-venture-nvh-2242648.htm - [cache](#) - Yippy News

more | all

Yippy example

The screenshot shows a web browser window with the Yippy search engine interface. The search query is "machine learning". The results page is titled "Results 1-28 of 49 in Neural networks". On the left side, there is a navigation menu with categories like "Sources Sites Time Topics", "Top 583 Results", and "Neural networks (40)". The main content area displays a list of search results, each with a title, a brief description, and a link to the source. The results include articles from various sources such as "European Radiology and Imaging News", "Cam Forx", "Random Acts of Chooch", "D3 - Home", "Innovation | Cloud-Based Analytics & Data-Driven Platforms", "Database of Free Online Computer Science and Programming Books, Textbooks, and Lecture Notes", "Virtual Exchange Deal", and "Schoolinfosystem.org | Curated Education Information".

Yippy machine learning Search

Results 1-28 of 49 in Neural networks

Sources Sites Time Topics

Top 583 Results

- + Language, Natural (53)
- + Management (41)
- + Marketing (40)
- **Neural networks (40)**
 - + Pattern recognition (15)
 - + Imaging (7)
 - + Knowledge, Intelligent (7)
 - Deep (3)
 - + Science, Programming (5)
 - + Artificial Neural Networks (4)
 - Neural Networks, And Genetic Algorithms (3)
 - Random (2)
 - Artificial intelligence, distributed (2)
 - Industrial automation (2)
 - Other Topics (8)
- + Image (26)
- + Security, Cyber (21)
- + Microsoft, Azure Machine Learning (14)
- + Digital (21)
- + Reviews (21)
- + Developers (11)
- + Paper (12)
- + Library (12)
 - Without being explicitly programmed (3)
- + Machine Learning, And Scientific Data (10)
- + Reasoning (12)
- + Twitter (5)
- + Face-swap (3)
- + Support vector machines (3)
- + Machine Learning Journal (7)
- + Conference on Machine Learning (11)
- + Download (10)
- Tutorials (8)
- + Magazine (8)
- Planets (8)
- Araya, Marketwired (3)

European Radiology and Imaging News, Information, Education and Services [new window](#) [preview](#)
... CER and the physics community – are needed. Discuss **Machine learning** can help assess atherosclerosis February 7, 2018 – **Machine learning** techniques analyze imaging measurements to automatically stratify patients ...
[www.astronmri.europa.com/index.aspx?sec=def - cache](#) - Yippy Index

(No Title) [new window](#) [preview](#)
... latest research builds on the pioneering use of **machine learning** algorithms with brain imaging technology to “mind read.” ... replays memories faster than they are stored. A **machine learning** algorithm shows that during sleep, the brain actively ...
[https://inside-the-brain.com - cache](#) - Yippy Index

Cam Forx – Random Musings on MT4, Expert Advisors & Trading [new window](#) [preview](#)
... typical EA builder it uses **Neural Networks** or **Machine Learning** to create EAs. How does it do ... level. Included in their plethora of features are: **machine learning** techniques and genetic programming to automatically generate EA ...
[www.camforx.com - cache](#) - Yippy Index

Random Acts of Chooch - Random Acts Random Acts of Chooch [new window](#) [preview](#)
... fast and furious and the system includes a **machine learning** component; it identifies music and processes it separately ... export your files in any way you like. **Machine Learning** – the system includes **machine learning** components to constantly ...
[chooch.us - cache](#) - Yippy Index

D3 - Home [new window](#) [review](#)
... MMI communication, real-time analytics and control, and **machine learning**. It powers the Industrial Internet of Things (IIoT) and is driving Industry 4.0. Applications range from more efficient industrial motor systems to ... vision, factory automation, and the Industrial IoT. Embedded ...
[https://d3engineering.com - cache](#) - Yippy Index

Innovation | Cloud-Based Analytics & Data-Driven Platforms [new window](#) [review](#)
... Center for Health Information & Decision Systems Collaborate on **Machine Learning** and **Neural Network** Applications November 20, 2017 Innovation ...
[www.innovation.com - cache](#) - Yippy Index

Database of Free Online Computer Science and Programming Books, Textbooks, and Lecture Notes [new window](#) [review](#)
... Algorithms and Data Structures Artificial Intelligence Computer Vision **Machine Learning** **Neural Networks** Game Development and Multimedia Data Communication ...
[www.freetechnobooks.com - cache](#) - Yippy Index

Virtual Exchange Deal [new window](#) [review](#)
... infrastructure reveal the secrets of deep **neural networks**, **machine learning**, substrate, shards, and much more. They also share ... infrastructure reveal the secrets of deep **neural networks**, **machine learning**, substrate, shards, and much more. They also share ... infrastructure reveal the secrets of deep **neural networks**, **machine learning**, substrate, shards, and much more. They also share ...
[https://en.com/virtual_exchange_deal?from=virtualexchangeideal.com - cache](#) - Yippy Index

Schoolinfosystem.org | Curated Education Information [new window](#) [review](#)
... later became the groundwork for “deep learning” or “**machine learning**”—had already been disproven. In the late ‘50s, a Cornell scientist named Frank Rosenblatt had proposed the world’s first **neural network machine**. It was called the Perceptron, and it had ...
[www.schoolinfosystem.org - cache](#) - Yippy Index

Welcome [new window](#) [preview](#)
... pro 2016 \$25 Multidimensional Particle Swarm Optimization for **Machine Learning** and Pattern Recognition PDF eBook \$10 The C ...
[f5t9wz-ws - cache](#) - Yippy Index



Image as data set

- ▶ a pixel: a vector in \mathbb{R}^3
- ▶ image: set of pixels a.k.a. vectors
- ▶ clustering: comparable pixels

Image quantization



Pixels

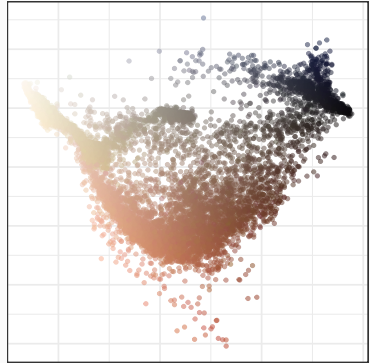


Image quantization



50 typical pixels

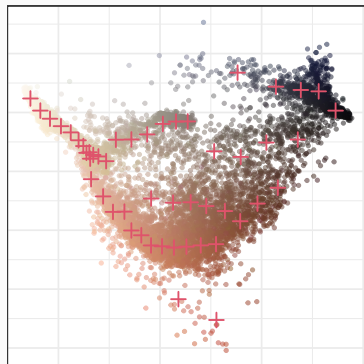
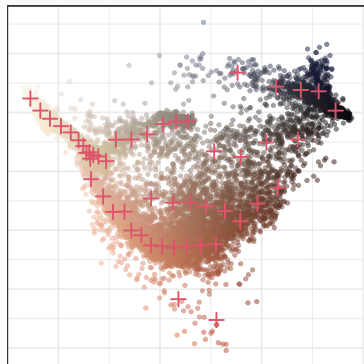


Image quantization



50 typical pixels



How to build a clustering method?

Ingredients

- ▶ similarity or dissimilarity: how to compare objects?
- ▶ group structure: set, “fuzzy” set, probabilistic membership?
- ▶ clustering structure: disjoint sets, overlapping sets, complete partition?

Classical example: standard k-means

- ▶ dissimilarity: euclidean distance on numerical variables (i.e. $\mathcal{X} = \mathbb{R}^P$)
- ▶ groups: standard sets
- ▶ clustering structure: groups for a partition of the data set

Clustering is ill posed

Problems

- ▶ vague definition with possible extensions (e.g. find groups that reflect the “underlying structure” of the data set)
- ▶ vastly different practical goals (from pre-processing to knowledge discovery)
- ▶ no universal quality criterion:
 - ▶ almost one criterion per method!
 - ▶ lack of task oriented criterion
- ▶ impossibility result in some situations

Clustering is ill posed

Problems

- ▶ vague definition with possible extensions (e.g. find groups that reflect the “underlying structure” of the data set)
- ▶ vastly different practical goals (from pre-processing to knowledge discovery)
- ▶ no universal quality criterion:
 - ▶ almost one criterion per method!
 - ▶ lack of task oriented criterion
- ▶ impossibility result in some situations

Clustering as a process

- ▶ for knowledge discovery
- ▶ usefulness rather than quality

Using a clustering

- ▶ cluster analysis: what do they contain?
 - ▶ list of the members
 - ▶ representative element(s)
- ▶ cluster “positioning”:
 - ▶ relative positioning in space
 - ▶ significant differences

Tasks oriented analysis

- ▶ can the analyst understand the clusters?
- ▶ can the clustering summarize the data? In what sense?

Families of clustering algorithms

Four main families...

- ▶ hierarchical algorithms: hierarchical clustering
- ▶ centroid based algorithms: k-means
- ▶ density based algorithms: DBSCAN
- ▶ probabilistic algorithms: mixture models and EM

Families of clustering algorithms

Four main families...

- ▶ hierarchical algorithms: hierarchical clustering
- ▶ centroid based algorithms: k-means
- ▶ density based algorithms: DBSCAN
- ▶ probabilistic algorithms: mixture models and EM

but a complex landscape

- ▶ k-means as a limit case of some mixture models
- ▶ DBSCAN is somewhat related to single linkage hierarchical clustering
- ▶ numerous hybrid techniques
- ▶ etc.

Minimal assumption

- ▶ \mathcal{X} is equipped with a dissimilarity d
- ▶ d is a **dissimilarity** on \mathcal{X} iff:
 1. d is a function from $\mathcal{X} \times \mathcal{X}$ to \mathbb{R}^+
 2. $\forall \mathbf{X}, \mathbf{X}', d(\mathbf{X}, \mathbf{X}') = d(\mathbf{X}', \mathbf{X})$
 3. $\forall \mathbf{X}, \mathbf{X}', \mathbf{X} \neq \mathbf{X}' \Leftrightarrow d(\mathbf{X}, \mathbf{X}') > 0$

Links with clustering

- ▶ many algorithms can work with arbitrary dissimilarities
- ▶ density, separability, compactness, in general quality metrics are expressed in terms of the dissimilarity
- ▶ results **strongly** depend on the dissimilarity
- ▶ plain \mathbb{R}^P : euclidean dissimilarity!

Introduction

Hierarchical clustering

K-means and related methods

DBSCAN

Fuzzy and probabilistic models

Clustering and partition

$$\mathcal{D} = ((\mathbf{X}_i)_{1 \leq i \leq N})$$

- ▶ group: a subset of $\{1, \dots, N\}$
- ▶ clustering structure: each observation (i.e. index in $\{1, \dots, N\}$) is in one unique group + no empty group
- ▶ building a clustering \Leftrightarrow choosing a partition!
- ▶ *here a cluster is also a class*

Clustering and partition

$$\mathcal{D} = ((\mathbf{X}_i)_{1 \leq i \leq N})$$

- ▶ group: a subset of $\{1, \dots, N\}$
- ▶ clustering structure: each observation (i.e. index in $\{1, \dots, N\}$) is in one unique group + no empty group
- ▶ building a clustering \Leftrightarrow choosing a partition!
- ▶ *here a cluster is also a class*

Remarks

- ▶ this is only *one* possible structure among others!
- ▶ one can have $\mathbf{X}_i = \mathbf{X}_j$ for $i \neq j$
 - ▶ we expect i and j to be in the same group!
 - ▶ groups are described at the index level

Partial order on partitions

- ▶ a partition P is **finer** than a partition Q ($P \leq Q$) if any class of P is a subset of a class of Q
- ▶ example

$$P = \{\{1, 2\}, \{3, 4\}, \{5, 6\}\} \leq \{\{1, 2, 3, 4\}, \{5, 6\}\} = Q$$

Hierarchy

A hierarchy for $\mathcal{D} = ((\mathbf{X}_i)_{1 \leq i \leq N})$ is a **fully ordered** set of partitions containing the trivial partitions

1. $\{\{1, \dots, N\}\}$
2. $\{\{1\}, \{2\}, \dots, \{N\}\}$

Graphical representation of a hierarchy by a tree

Graphical representation of a hierarchy by a tree

A B C D E F

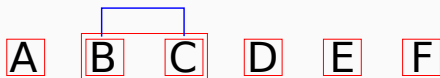
Graphical representation of a hierarchy by a tree

- ▶ most refined partition (trivial)

A B C D E F

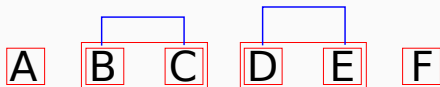
Graphical representation of a hierarchy by a tree

- ▶ most refined partition (trivial)
- ▶ B and C are in the same class at level 2: leaves of a common node

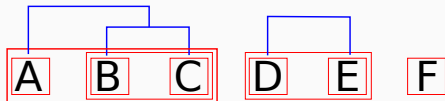


Graphical representation of a hierarchy by a tree

- ▶ most refined partition (trivial)
- ▶ B and C are in the same class at level 2: leaves of a common node
- ▶ same thing for D and E, but a level 3

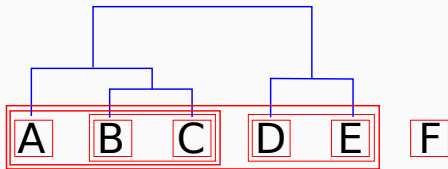


Graphical representation of a hierarchy by a tree



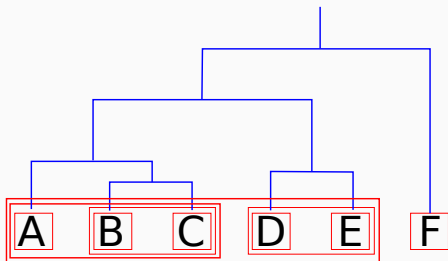
- ▶ most refined partition (trivial)
- ▶ B and C are in the same class at level 2: leaves of a common node
- ▶ same thing for D and E, but a level 3
- ▶ A, B, and C: new node that shows the order between partitions

Graphical representation of a hierarchy by a tree



- ▶ most refined partition (trivial)
- ▶ B and C are in the same class at level 2: leaves of a common node
- ▶ same thing for D and E, but a level 3
- ▶ A, B, and C: new node that shows the order between partitions
- ▶ more nodes until the coarsest partition

Graphical representation of a hierarchy by a tree

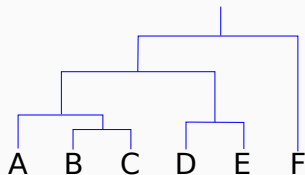


- ▶ most refined partition (trivial)
- ▶ B and C are in the same class at level 2: leaves of a common node
- ▶ same thing for D and E, but a level 3
- ▶ A, B, and C: new node that shows the order between partitions
- ▶ more nodes until the coarsest partition

Dendrogram

Understanding dendrograms

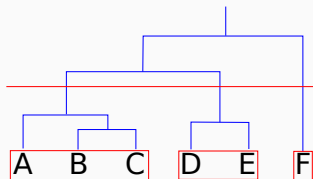
- ▶ provides a summary of the hierarchy
- ▶ a level in the tree (a.k.a. a node) corresponds to merging two classes
- ▶ the height of a level/node gives the “quality” of the merging



Dendrogram

Understanding dendrograms

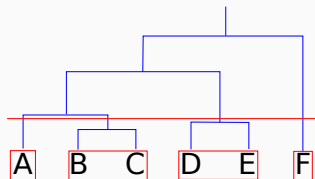
- ▶ provides a summary of the hierarchy
- ▶ a level in the tree (a.k.a. a node) corresponds to merging two classes
- ▶ the height of a level/node gives the “quality” of the merging
- ▶ horizontal cut \Rightarrow partition



Dendrogram

Understanding dendrograms

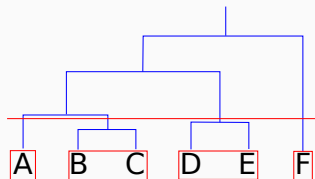
- ▶ provides a summary of the hierarchy
- ▶ a level in the tree (a.k.a. a node) corresponds to merging two classes
- ▶ the height of a level/node gives the “quality” of the merging
- ▶ horizontal cut \Rightarrow partition



Dendrogram

Understanding dendrograms

- ▶ provides a summary of the hierarchy
- ▶ a level in the tree (a.k.a. a node) corresponds to merging two classes
- ▶ the height of a level/node gives the “quality” of the merging
- ▶ horizontal cut \Rightarrow partition

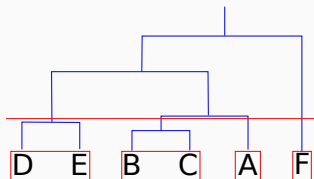


Common errors

- ▶ the dendrogram summarizes the **hierarchy** not the data
- ▶ do not interpret the order of the leaves: it is almost arbitrary (2^{N-2} orders for N leaves)

Understanding dendrograms

- ▶ provides a summary of the hierarchy
- ▶ a level in the tree (a.k.a. a node) corresponds to merging two classes
- ▶ the height of a level/node gives the “quality” of the merging
- ▶ horizontal cut \Rightarrow partition

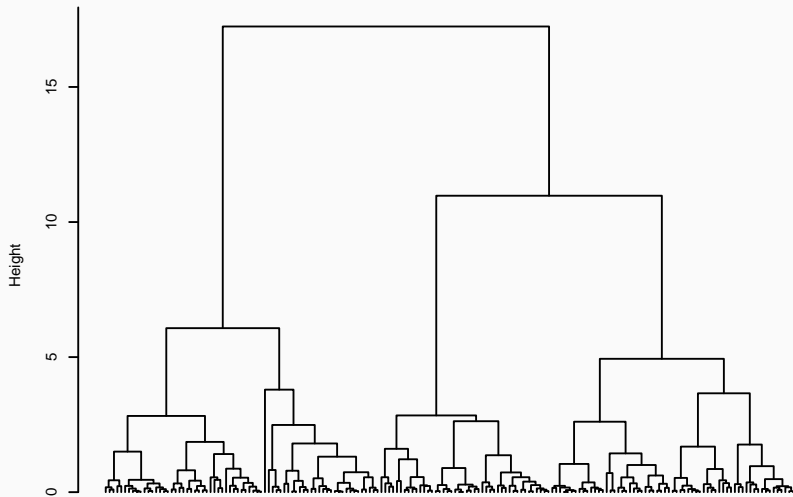


Common errors

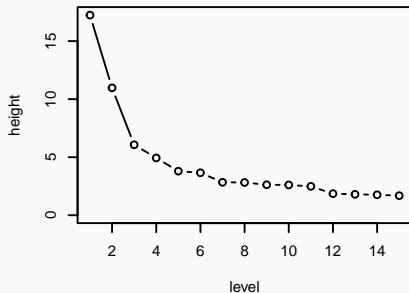
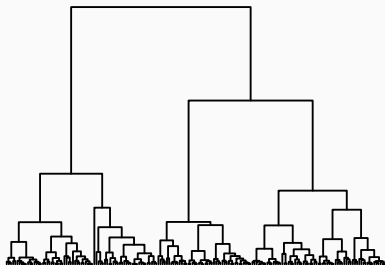
- ▶ the dendrogram summarizes the **hierarchy** not the data
- ▶ do not interpret the order of the leaves: it is almost arbitrary (2^{N-2} orders for N leaves)

Example

Cluster Dendrogram

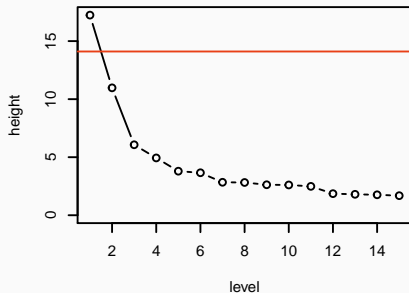
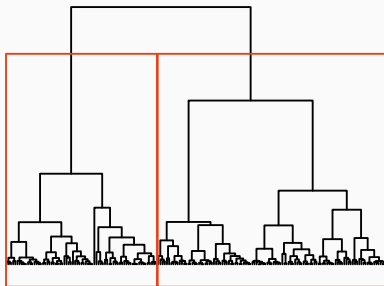


Example



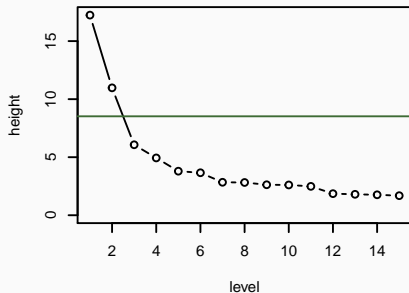
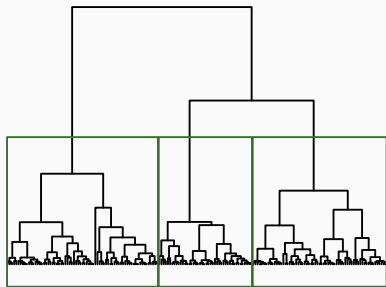
- ▶ look for “gaps” between levels: potential candidates for interesting partitions
- ▶ local partitions (i.e. branches) might also be interesting

Example



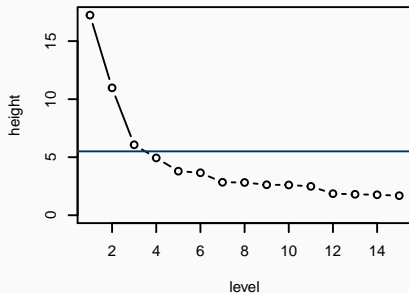
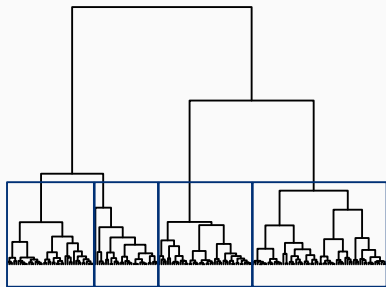
- ▶ look for “gaps” between levels: potential candidates for interesting partitions
- ▶ local partitions (i.e. branches) might also be interesting

Example



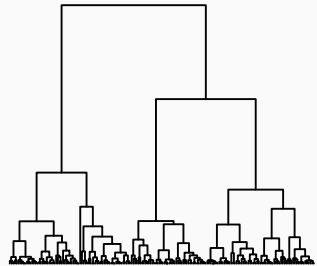
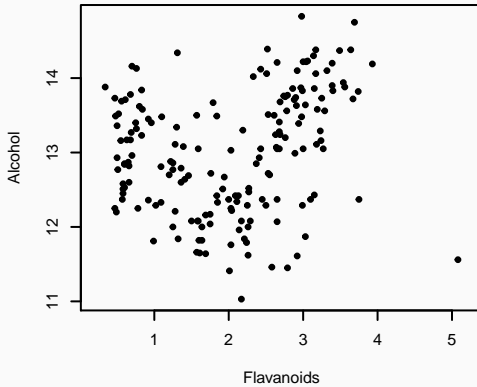
- ▶ look for “gaps” between levels: potential candidates for interesting partitions
- ▶ local partitions (i.e. branches) might also be interesting

Example

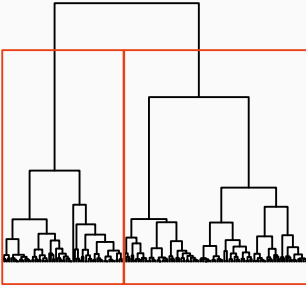
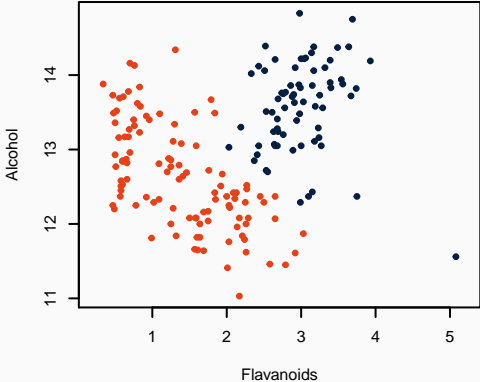


- ▶ look for “gaps” between levels: potential candidates for interesting partitions
- ▶ local partitions (i.e. branches) might also be interesting

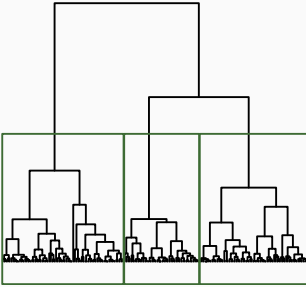
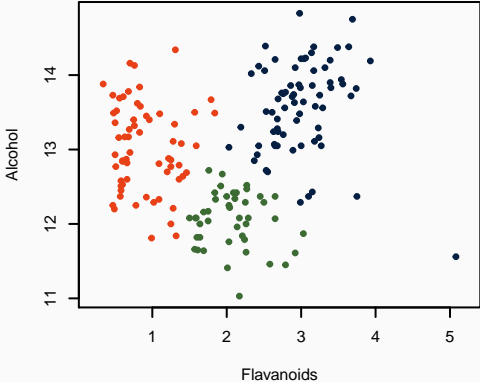
Example



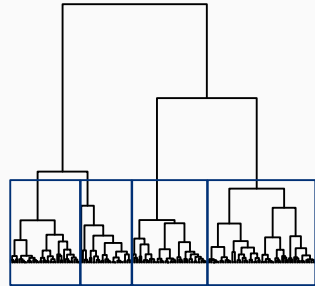
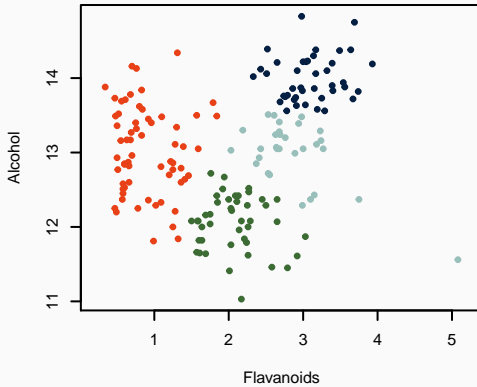
Example



Example



Example



Agglomerative methods

- ▶ start with $P = \{\{1\}, \{2\}, \dots, \{N\}\}$
- ▶ merge two classes in such a way that the resulting class is a *cluster*
- ▶ keep doing that until reaching $Q = \{\{1, \dots, N\}\}$

Divisive methods

- ▶ start with $Q = \{\{1, \dots, N\}\}$
- ▶ split one class into two sub-classes that are *clusters*
- ▶ keep doing that until reaching $P = \{\{1\}, \{2\}, \dots, \{N\}\}$

Common elements

- ▶ those methods produce a hierarchy with N partitions
- ▶ they need a way to assess suitability of the classes as clusters
- ▶ dissimilarity based algorithms

Differences

- ▶ numerous agglomerative methods
- ▶ relatively few divisive ones
- ▶ very different computational problems:
 - ▶ agglomerative: $\Theta(N^2)$ potential merges at each step
 - ▶ divisive: $\Theta(2^{N-1})$ possible splits at the first step

Agglomerative methods

Core principle

- ▶ merge classes when they contain *similar* objects
- ▶ for singleton classes $\{\mathbf{X}_i\}$ and $\{\mathbf{X}_j\}$ use $d(\mathbf{X}_i, \mathbf{X}_j)$ to judge similarity
- ▶ key point: extend the dissimilarity to **groups of objects**

General algorithm

- ▶ initial partition: $\mathcal{P}^1 = \{\{1\}, \{2\}, \dots, \{N\}\}$
- ▶ for k from 2 to N :
 - ▶ compute the dissimilarity between all current classes in \mathcal{P}^{k-1}
 - ▶ build \mathcal{P}^k from \mathcal{P}^{k-1} by merging the two least dissimilar classes

Aggregation functions

Let A and B be two classes (of indices)

- ▶ single linkage (min)

$$d_S(A, B) = \min_{i \in A, j \in B} d(\mathbf{X}_i, \mathbf{X}_j)$$

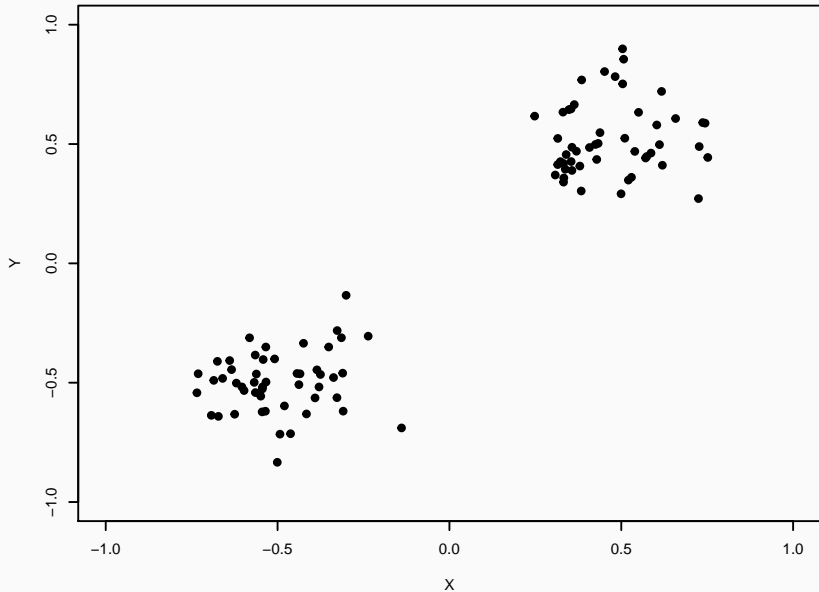
- ▶ complete linkage (max)

$$d_S(A, B) = \max_{i \in A, j \in B} d(\mathbf{X}_i, \mathbf{X}_j)$$

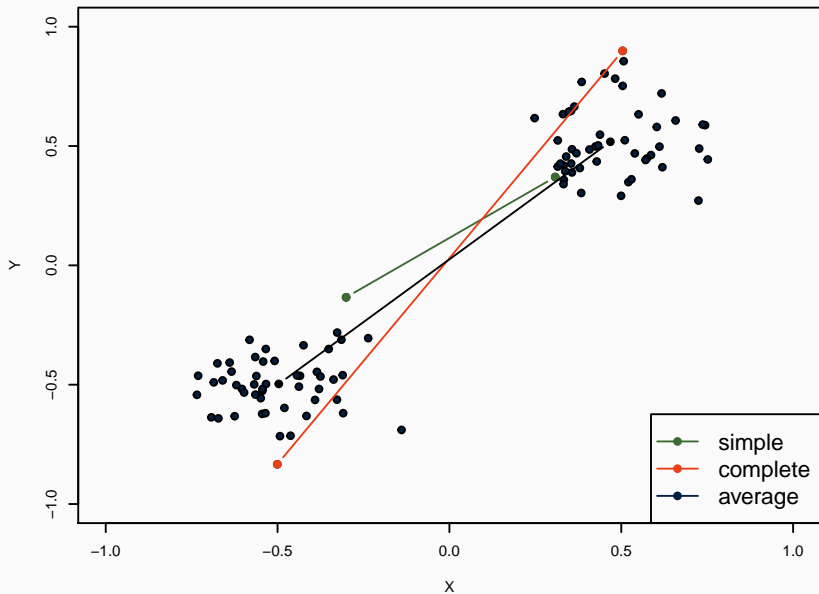
- ▶ average linkage

$$d_S(A, B) = \frac{1}{|A||B|} \sum_{i \in A, j \in B} d(\mathbf{X}_i, \mathbf{X}_j)$$

Illustration



Illustration



Naive approach

- ▶ $N - 1$ steps
- ▶ step k searches for the best merge over $\frac{(N-k+1)(N-k)}{2}$ pairs of classes
- ▶ overall $\Theta(N^3)$ with adapted dissimilarity calculations
- ▶ standard implementation in many packages

Optimized solution

- ▶ simple priority list based solutions in $\Theta(N^2 \log N)$
- ▶ optimized algorithms in $\Theta(N^2)$ (see [fastcluster](#) in R)
- ▶ notice that the dissimilarity must be calculated! For $\mathcal{X} = \mathbb{R}^P$, this adds in general a $\Theta(N^2 P)$ cost.

Outline

1. choose a dissimilarity
2. choose an aggregation function
3. build the hierarchy
4. study the dendrogram:
 - ▶ the heights are the dissimilarities between merged classes
 - ▶ gaps can be seen as abrupt changes in merge qualities
 - ▶ visualization methods can be used to display the classes (e.g. Principal Component Analysis)

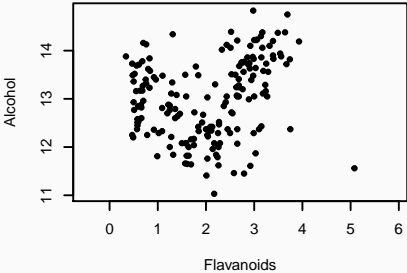
Outline

1. choose a dissimilarity
2. choose an aggregation function
3. build the hierarchy
4. study the dendrogram:
 - ▶ the heights are the dissimilarities between merged classes
 - ▶ gaps can be seen as abrupt changes in merge qualities
 - ▶ visualization methods can be used to display the classes (e.g. Principal Component Analysis)

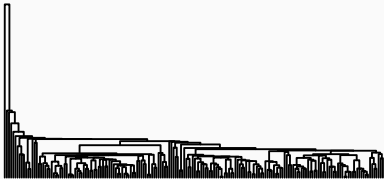
Limitations

- ▶ results strongly depend on both the dissimilarity and the aggregation function
- ▶ somewhat slow
- ▶ exploratory tool rather than clustering tool

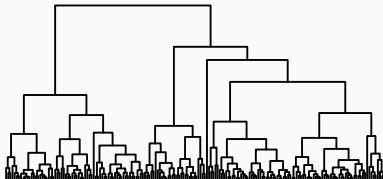
Example



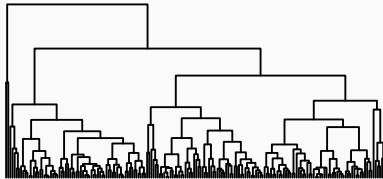
Single



Complete

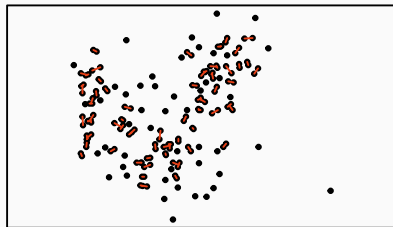
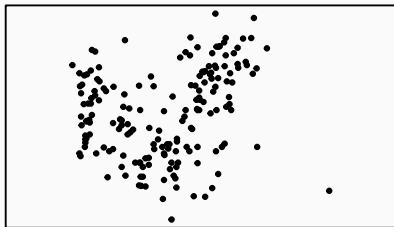


Average

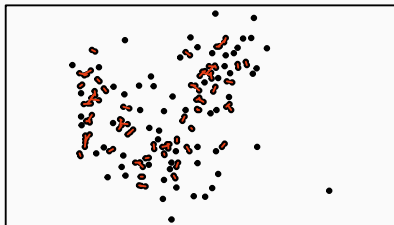


Example

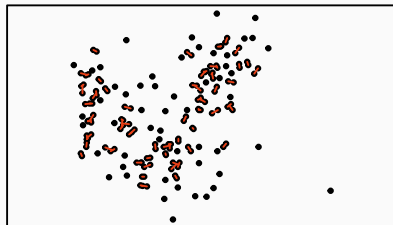
Complete



Single

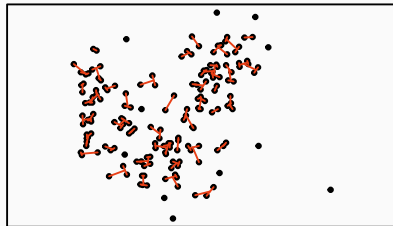
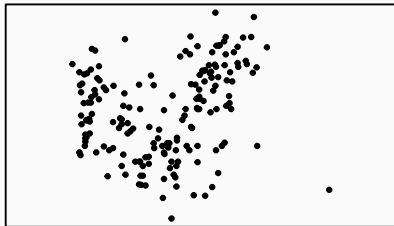


Average

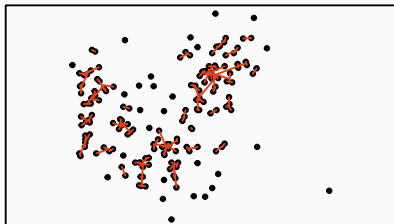


Example

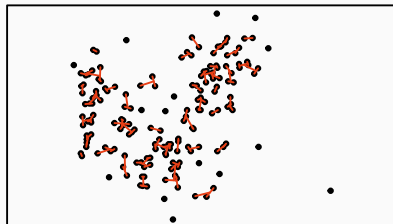
Complete



Single

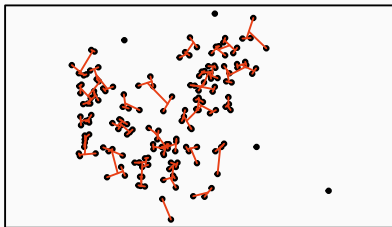
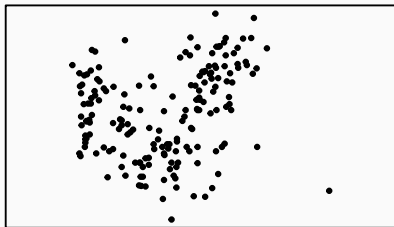


Average

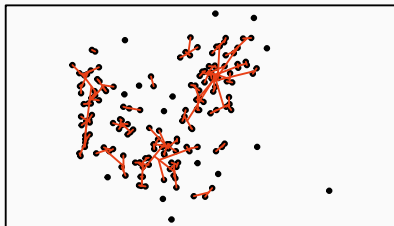


Example

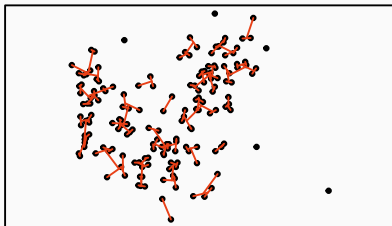
Complete



Single

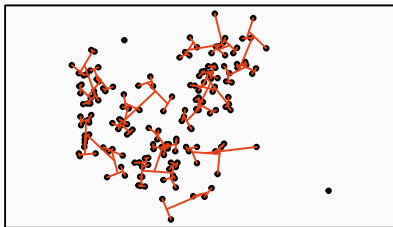
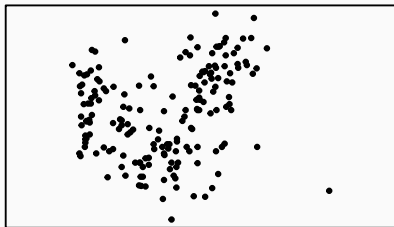


Average

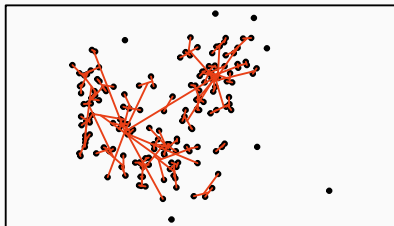


Example

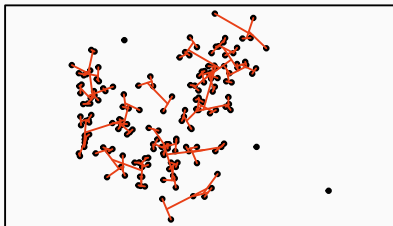
Complete



Single

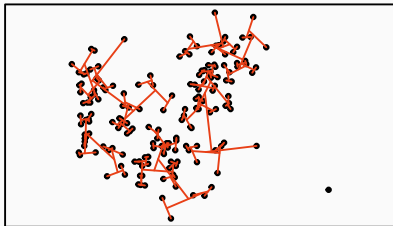
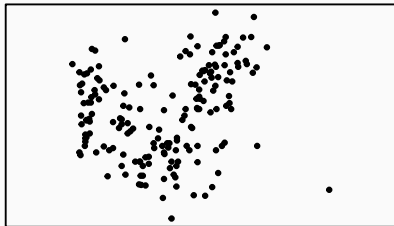


Average

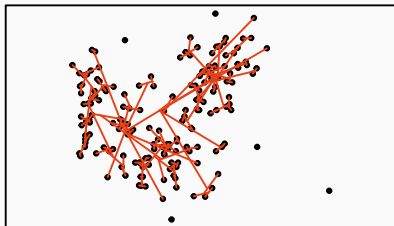


Example

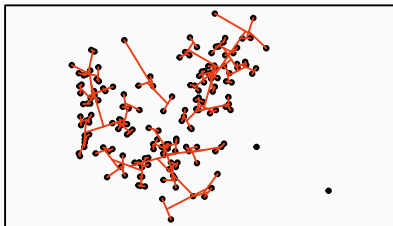
Complete



Single

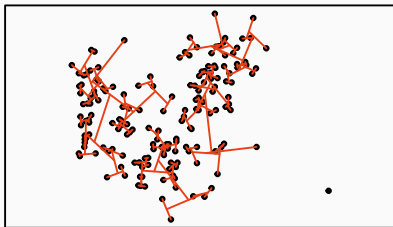
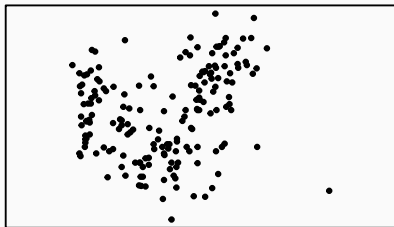


Average

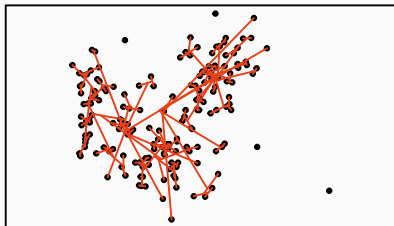


Example

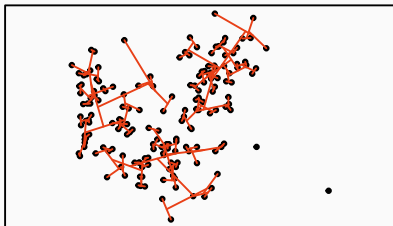
Complete



Single

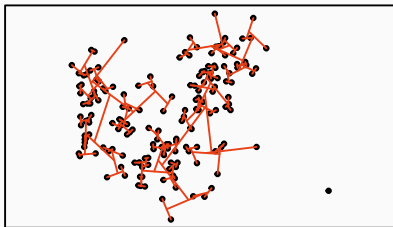
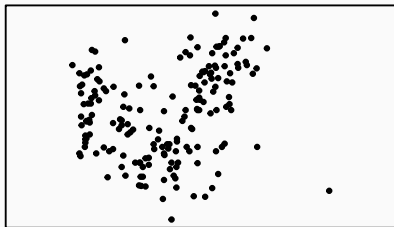


Average

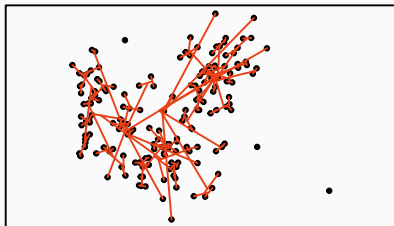


Example

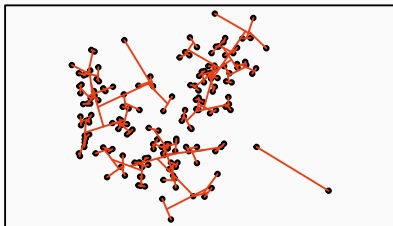
Complete



Single

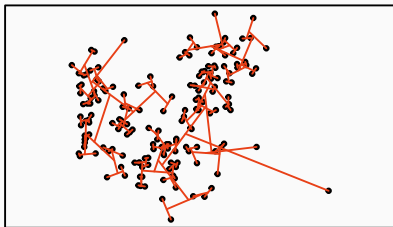
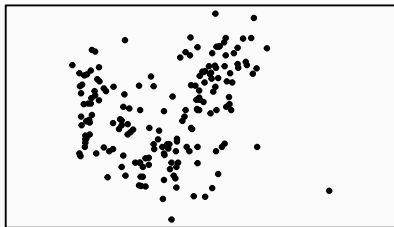


Average

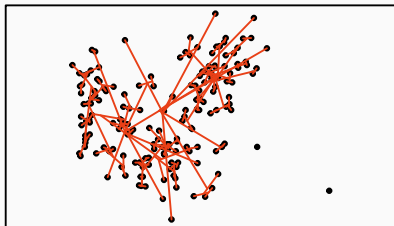


Example

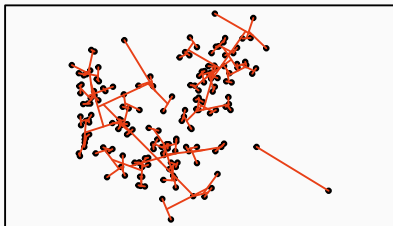
Complete



Single

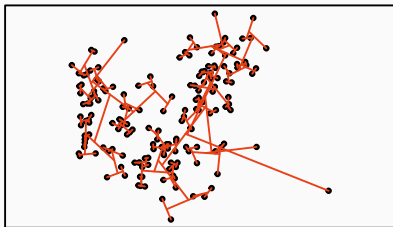
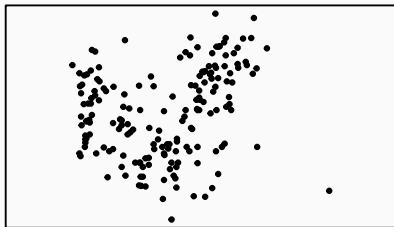


Average

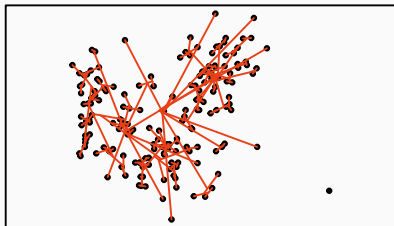


Example

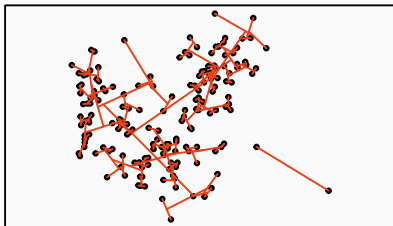
Complete



Single

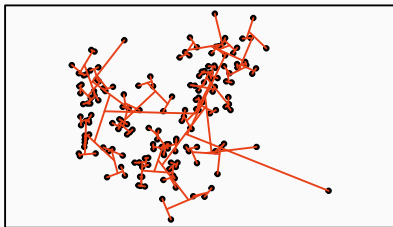
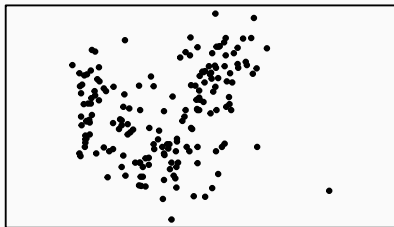


Average

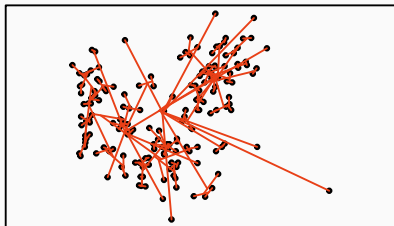


Example

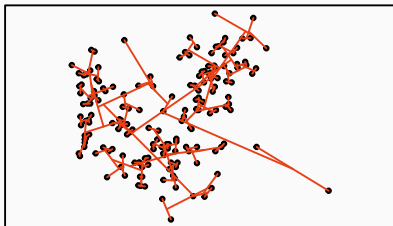
Complete



Single



Average



Remarks

- ▶ initial behaviors are quite independent from the aggregation function
- ▶ single linkage:
 - ▶ tendency to produce “long” classes (chaining)
 - ▶ outlier isolation
- ▶ complete linkage:
 - ▶ balanced class sizes
 - ▶ classes can be close one to another (crowding)
- ▶ average linkage
 - ▶ average behavior...
 - ▶ resistant to noise but sensitive to dissimilarity transformation

Classes as clusters

A partition provides a good *clustering* if its classes are

- ▶ homogeneous
- ▶ well separated

Criteria

- ▶ homogeneity via the diameter

$$D(A) = \max_{i \in A, j \in A} d(\mathbf{X}_i, \mathbf{X}_j)$$

- ▶ separation via dissimilarity
- ▶ a good clustering: small diameters and large dissimilarities!

Single linkage

- ▶ no control over $D(A)$ during the merge
- ▶ could end up with classes with very different diameters
- ▶ distances between classes can be very small relatively to their diameter

Complete linkage

- ▶ the diameter is the quality measure of a merge!
- ▶ however, no control at all over separability

Two extreme cases

with the average linkage in between...

General rules of thumb

- ▶ single linkage
 - ▶ gives frequently poor results
 - ▶ might be useful to spot outliers
- ▶ average linkage
 - ▶ gives generally good clusters
 - ▶ interesting compromise between diameter and separation
- ▶ complete linkage
 - ▶ maximally homogeneous classes
 - ▶ useful when there are no real clusters (i.e., clusters that can be easily separated)

Partition quality

Quality versus distance

- ▶ aggregation functions express the quality of a merge in terms of distances (min, max or average)
- ▶ alternative solution: express the quality using the resulting cluster

Within “variance”

- ▶ within variance of a class A (homogeneity measure)

$$W(A) = \frac{1}{|A|} \sum_{i \in A, j \in A} d(\mathbf{x}_i, \mathbf{x}_j)$$

- ▶ total within variance of a partition $\mathcal{P} = \{A_1, \dots, A_K\}$

$$W(\mathcal{P}) = \sum_{k=1}^K W(A_k)$$

Optimizing the within variance

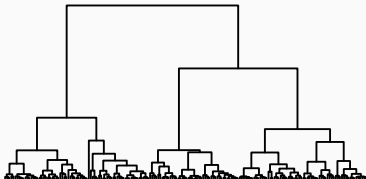
- ▶ same aggregative algorithm than with dissimilarities
- ▶ (non)quality of a merge:
 - ▶ the increase in within variance induced by merging A with B
 - ▶ local computation

$$\Delta_{(A,B) \rightarrow A \cup B} = W(A \cup B) - W(A) - W(B)$$

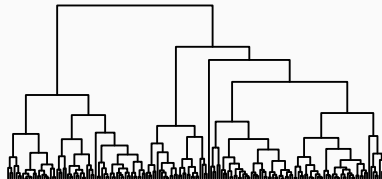
- ▶ can be seen as a greedy optimization of $W(\mathcal{P})$ at each stage of the hierarchy

Example

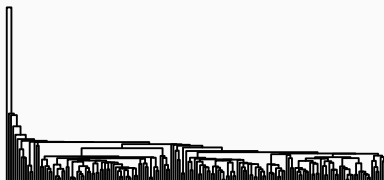
Ward



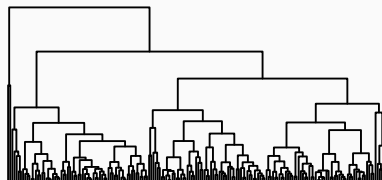
Complete



Single

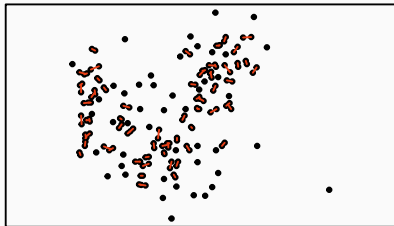


Average

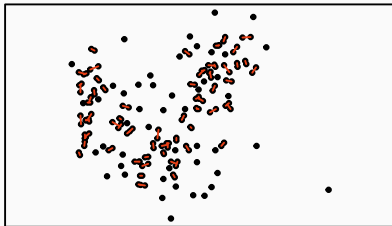


Example

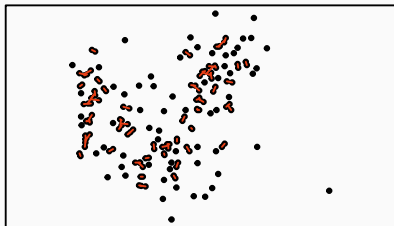
Ward



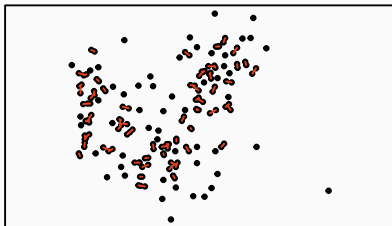
Complete



Single

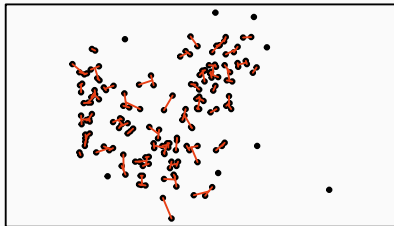


Average

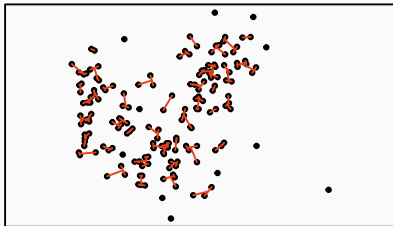


Example

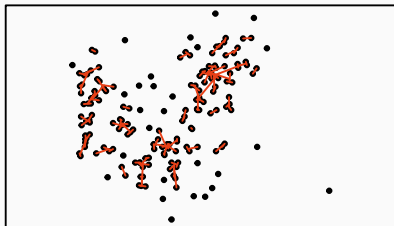
Ward



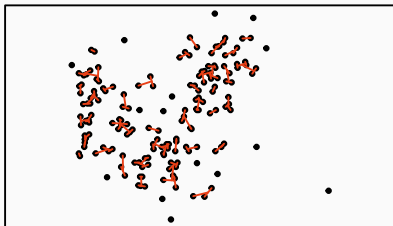
Complete



Single

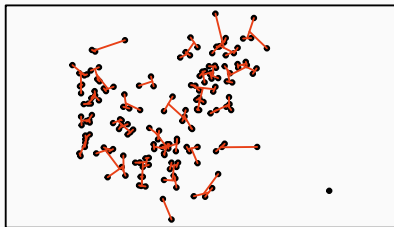


Average

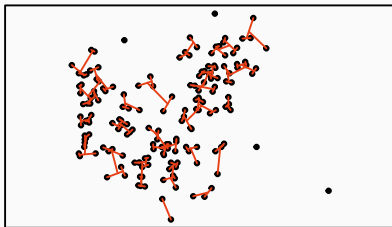


Example

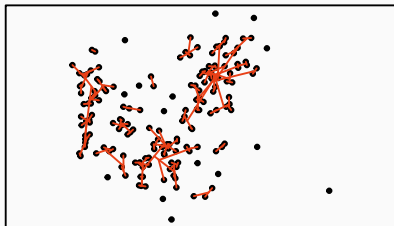
Ward



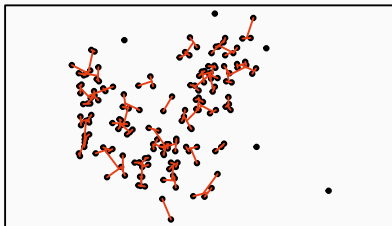
Complete



Single

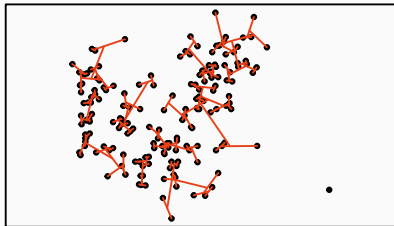


Average

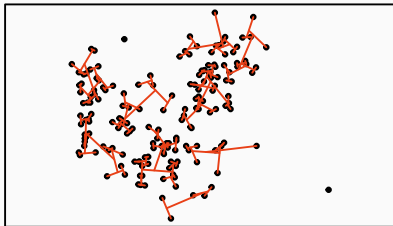


Example

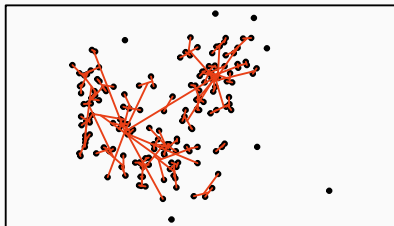
Ward



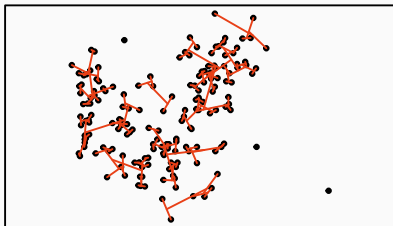
Complete



Single

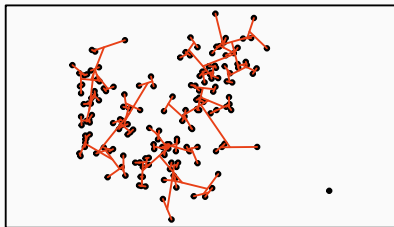


Average

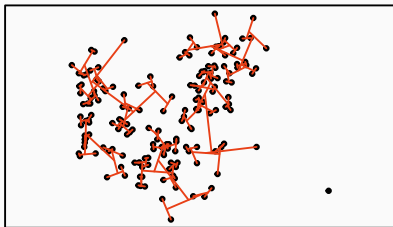


Example

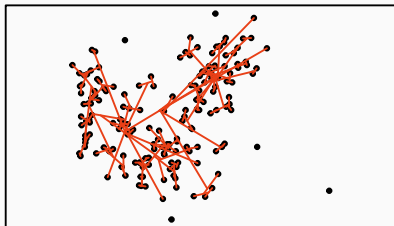
Ward



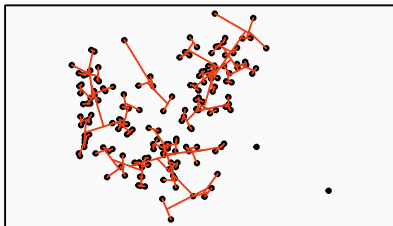
Complete



Single

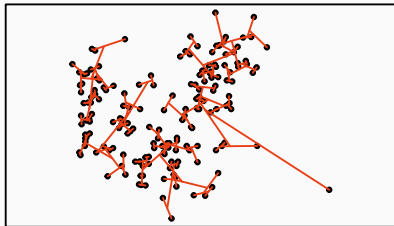


Average

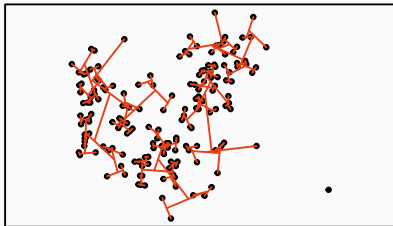


Example

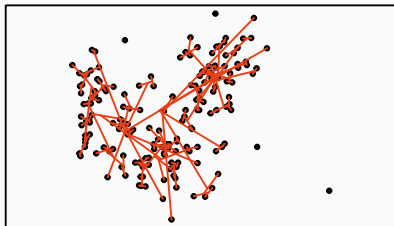
Ward



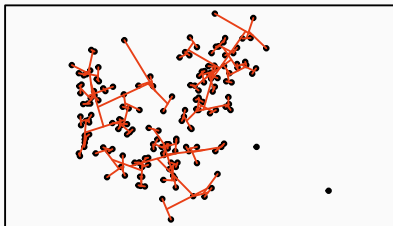
Complete



Single

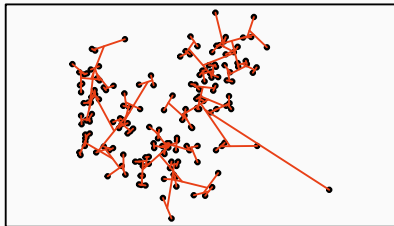


Average

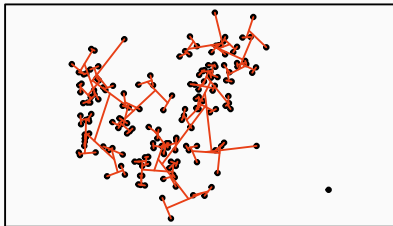


Example

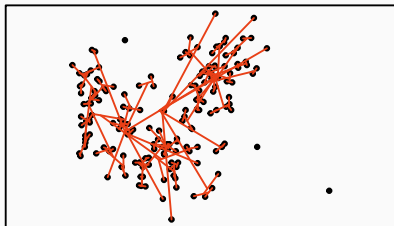
Ward



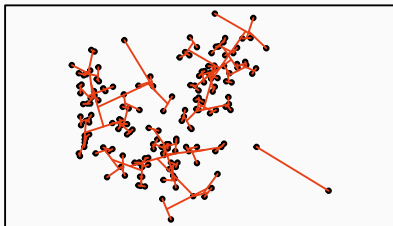
Complete



Single

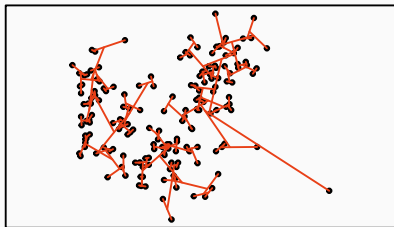


Average

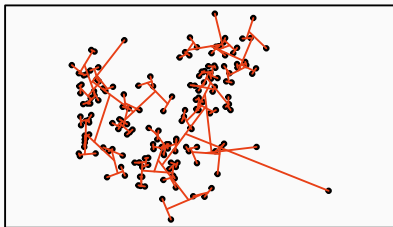


Example

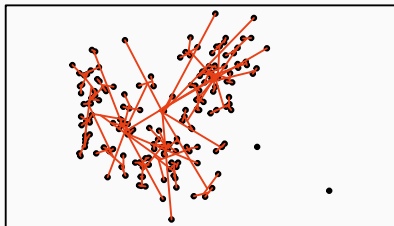
Ward



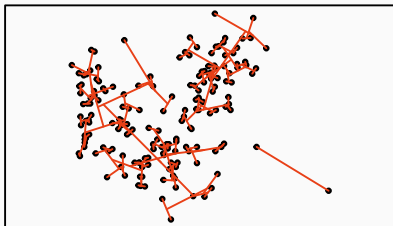
Complete



Single

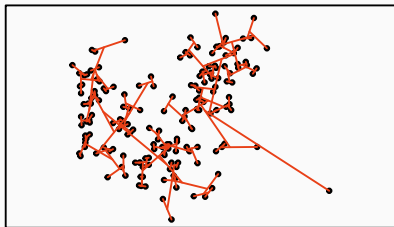


Average

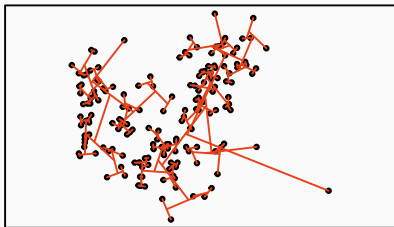


Example

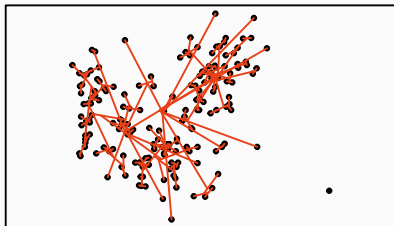
Ward



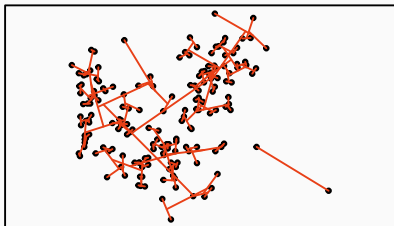
Complete



Single

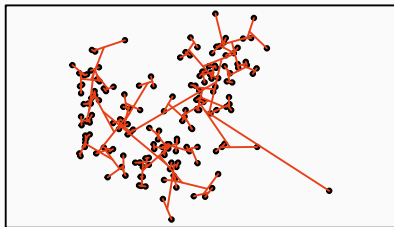


Average

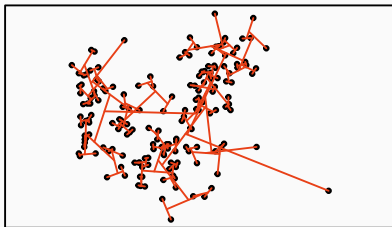


Example

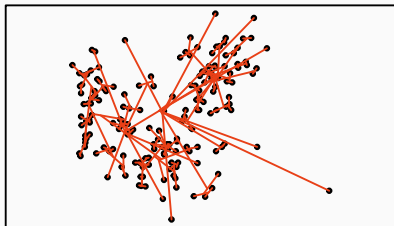
Ward



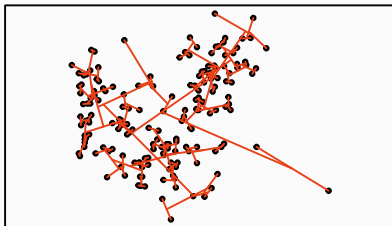
Complete



Single



Average



Ward versus average

- ▶ somewhat related hierarchy
- ▶ Ward's method tend to favor more balanced class size
- ▶ Ward's method is closely related to other methods (e.g. K-means)
- ▶ outliers are more easily aggregated with other points in the Ward's method

Numerous other variants

- ▶ centroid and median linkage (distances between classes induced by prototype based representation)
- ▶ minimax linkage (enclosing ball diameter)
- ▶ etc.

Generic algorithm

- ▶ initial partition: $\mathcal{P}^1 = \{1, \dots, N\}$
- ▶ for k from 2 to N :
 - ▶ chose a class to split
 - ▶ build \mathcal{P}^k from \mathcal{P}^{k-1} by splitting the chosen class into two sub classes

Difficulties

- ▶ choosing the class to split is relatively easy by using e.g. a diameter criterion
- ▶ but splitting is hard: too many possible splits $(2^{N-1} - 1)!$

DIANA

- ▶ Divisive ANALYSIS Clustering (available in R in [cluster](#))
- ▶ reference algorithm by Kaufman & Roussew, 1990

$\mathcal{P}^1 \leftarrow \{1, \dots, N\}$

for k in $2, \dots, N$ **do**

find C_j in \mathcal{P}^{k-1} with the largest diameter

find \mathbf{X}_l for $l \in C_j$ that maximizes $\sum_{l' \in C_j} d(\mathbf{X}_l, \mathbf{X}_{l'})$

$C_j^a \leftarrow \{l\}$ and $C_j^b \leftarrow C_j \setminus \{l\}$

repeat

for all t in C_j compute

$$D(t) = \frac{1}{|C_j^b|} \sum_{u \in C_j^b} d(\mathbf{X}_t, \mathbf{X}_u) - \frac{1}{|C_j^a|} \sum_{u \in C_j^a} d(\mathbf{X}_t, \mathbf{X}_u)$$

if $v = \arg \max_t D(t)$ is such that $D(v) > 0$ move v from C_k^b to C_k^a

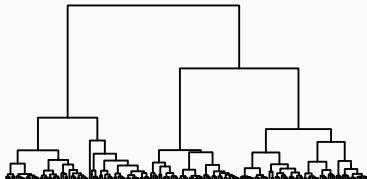
until $\max_t D(t) < 0$

define \mathcal{P}^k as \mathcal{P}^{k-1} in which C_j is replaced by C_j^a and C_j^b

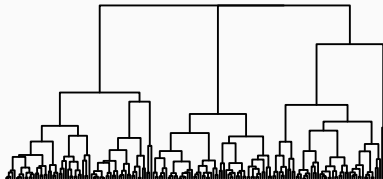
end for

Example

Ward

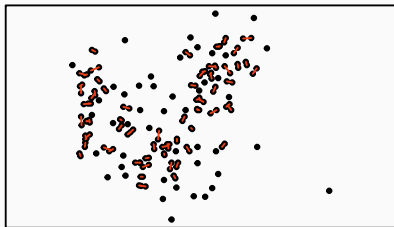


Diana

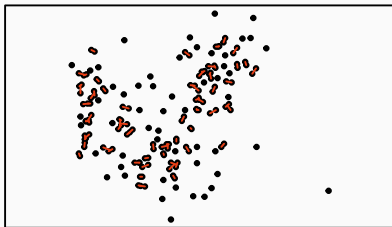


Example

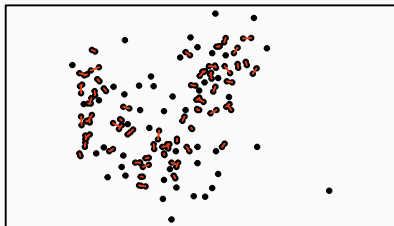
Ward



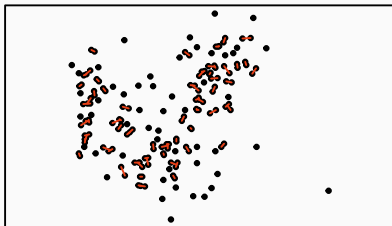
Average



Complete

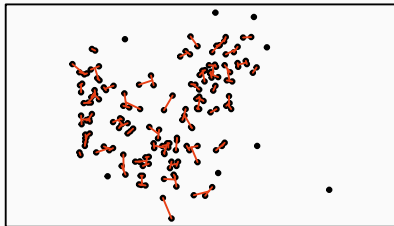


Diana

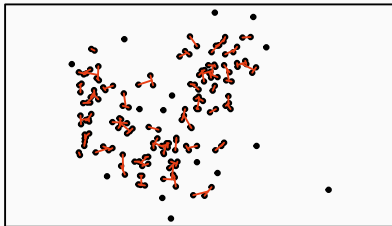


Example

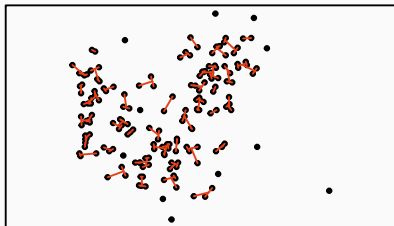
Ward



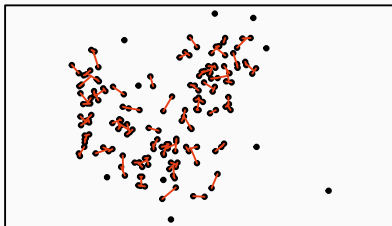
Average



Complete

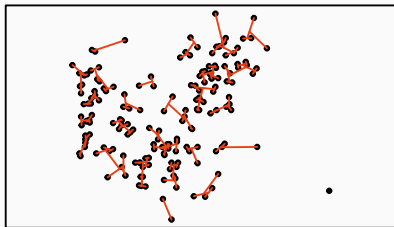


Diana

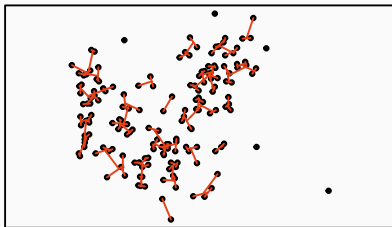


Example

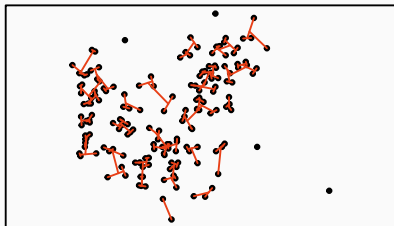
Ward



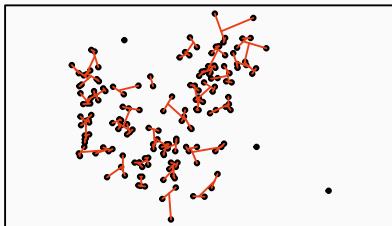
Average



Complete

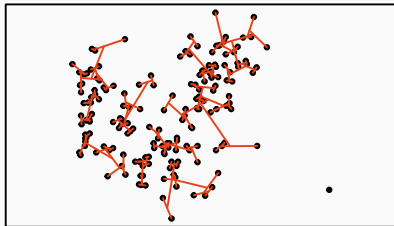


Diana

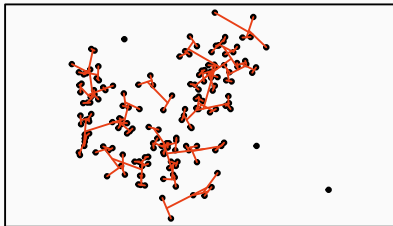


Example

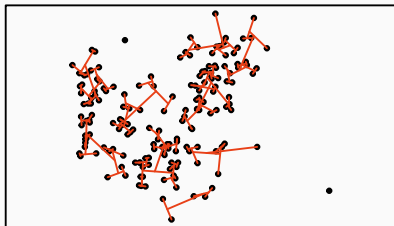
Ward



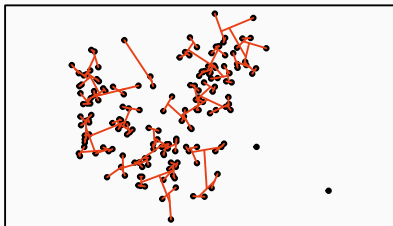
Average



Complete

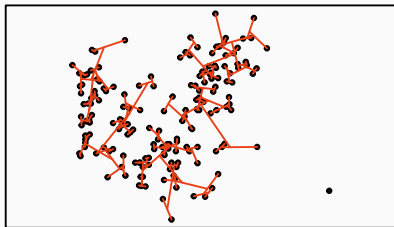


Diana

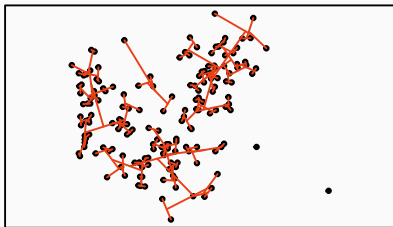


Example

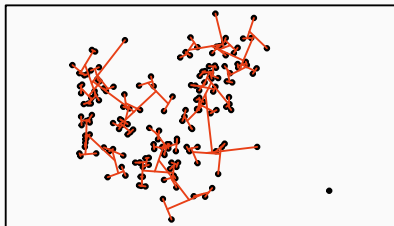
Ward



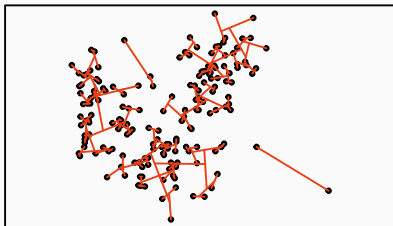
Average



Complete

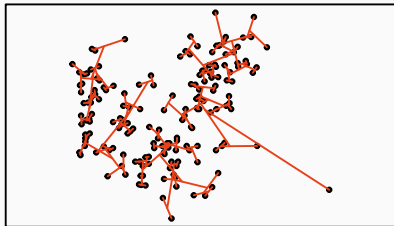


Diana

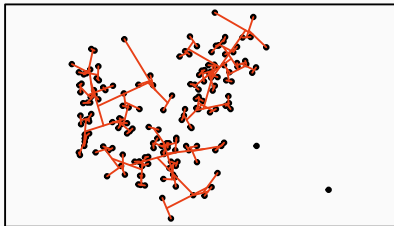


Example

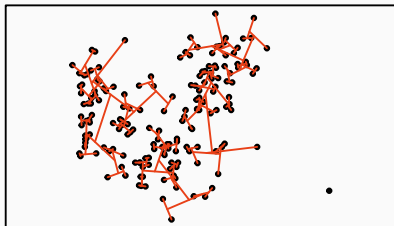
Ward



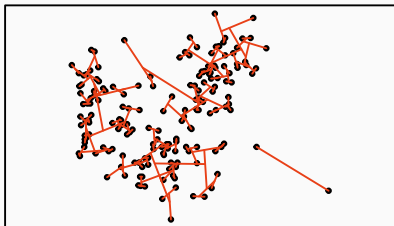
Average



Complete

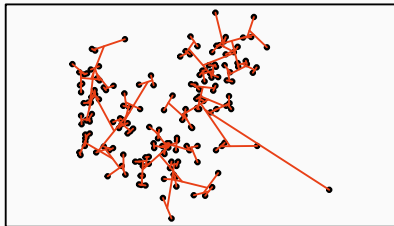


Diana

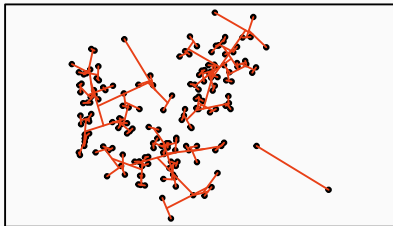


Example

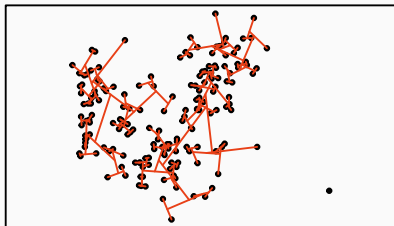
Ward



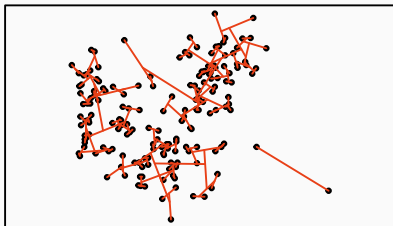
Average



Complete

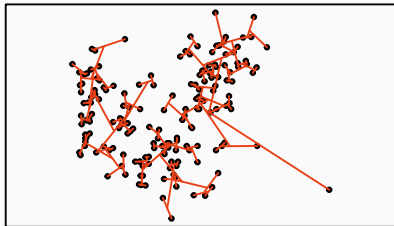


Diana

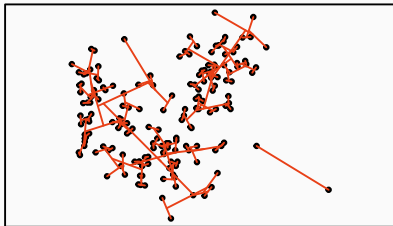


Example

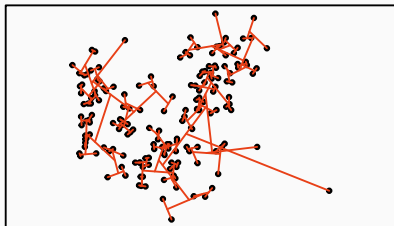
Ward



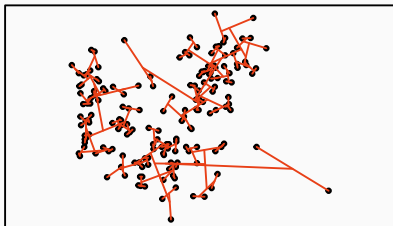
Average



Complete

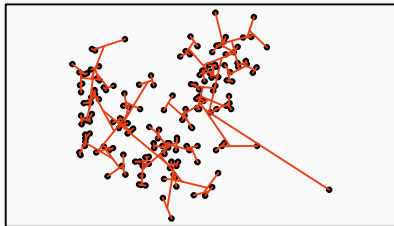


Diana

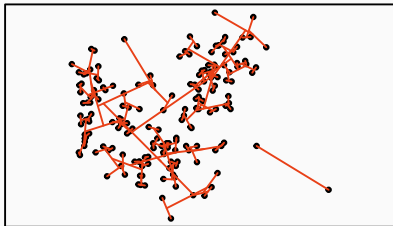


Example

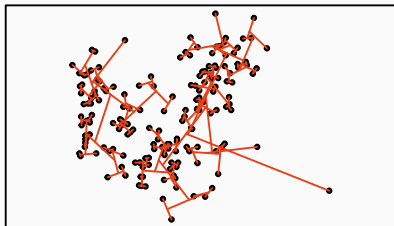
Ward



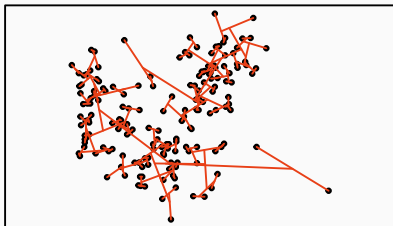
Average



Complete

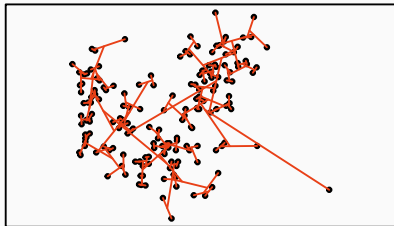


Diana

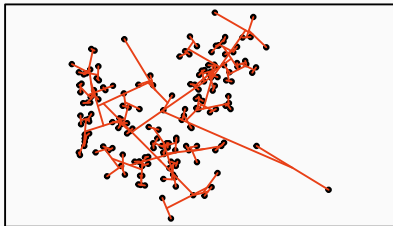


Example

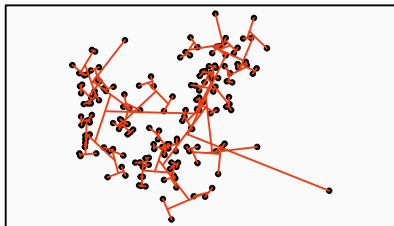
Ward



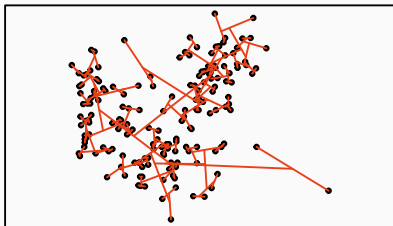
Average



Complete



Diana



Hierarchical clustering

- ☺ provide a hierarchy and a dendrogram
- ☺ applicable to any dissimilarity
- ☹ $\Theta(N^2)$ running time for aggregative methods
- ☹ not much guarantees about the final result
- ☹ inferior results for some methods (e.g. single linkage)

Generic questions

- ▶ dissimilarity?
- ▶ hierarchical method?
- ▶ cluster number?

Introduction

Hierarchical clustering

K-means and related methods

DBSCAN

Fuzzy and probabilistic models

Optimization point of view

- ▶ define a quality criterion for a clustering structure
- ▶ optimize this quality over “all” clustering structures
- ▶ typical example:
 - ▶ within variance as quality measure
 - ▶ optimized over partitions

Difficulties

- ▶ no obvious quality criterion for exploratory tasks
- ▶ very difficult discrete optimization problems (NP-hard in some cases)

Summarizing a data set

- ▶ key idea: represent a data set $\mathcal{D} = (\mathbf{X}_i)_{1 \leq i \leq N}$ by a smaller set of **prototypes** $\mathcal{D} = (\gamma_k)_{1 \leq k \leq K}$
- ▶ \mathbf{X}_i is presented by a prototype: z_i is the index in $\{1, \dots, K\}$ of this prototype
- ▶ natural risk associated to the problem

$$\mathcal{E}(\Gamma, \mathbf{z}) = \sum_{i=1}^N d(\mathbf{X}_i, \gamma_{z_i}),$$

where $\mathbf{z} = (z_1, \dots, z_N)$ and $\Gamma = (\gamma_1, \dots, \gamma_K)$.

Summarizing a data set

- ▶ key idea: represent a data set $\mathcal{D} = (\mathbf{X}_i)_{1 \leq i \leq N}$ by a smaller set of **prototypes** $\mathcal{D} = (\gamma_k)_{1 \leq k \leq K}$
- ▶ \mathbf{X}_i is presented by a prototype: z_i is the index in $\{1, \dots, K\}$ of this prototype
- ▶ natural risk associated to the problem

$$\mathcal{E}(\Gamma, \mathbf{z}) = \sum_{i=1}^N d(\mathbf{X}_i, \gamma_{z_i}),$$

where $\mathbf{z} = (z_1, \dots, z_N)$ and $\Gamma = (\gamma_1, \dots, \gamma_K)$.

- ▶ **natural clustering interpretation**: \mathbf{z} defines a partition of $\{1, \dots, N\}$ into K classes!

Difficult!

- ▶ minimizing $\mathcal{E}(\Gamma, \mathbf{z})$ is a combinatorial problem
- ▶ for standard d (such as the squared Euclidean distance when $\mathcal{X} = \mathbb{R}^P$) this is NP-hard

Simple sub-problems

- ▶ minimizing $\mathcal{E}(\Gamma, \mathbf{z})$ with respect to \mathbf{z} is easy

$$\arg \min_{\mathbf{z}} \mathcal{E}(\Gamma, \mathbf{z}) = \left(\arg \min_{k \in \{1, \dots, K\}} d(\mathbf{X}_1, \gamma_k), \dots, \arg \min_{k \in \{1, \dots, K\}} d(\mathbf{X}_N, \gamma_k) \right)$$

- ▶ if prototypes are restricted to be elements of the data set \mathcal{D} , optimizing with respect to Γ is also easy

$$\arg \min_{\Gamma} \mathcal{E}(\Gamma, \mathbf{z}) = \left(\arg \min_{\gamma_k \in \mathcal{D}} \sum_{i, z_i=k} d(\mathbf{X}_i, \gamma_k) \right)_{1 \leq k \leq K}$$

A.k.a. alternating optimization

a simple minimization algorithm for a function $F(u, v)$:

select u_0 randomly

$k \leftarrow 1$

repeat

$$v_k = \arg \min_v F(u_{k-1}, v)$$

$$u_k = \arg \min_u F(u, v_k)$$

$k \leftarrow k + 1$

until convergence

Properties

- ▶ converges to a local minimum
- ▶ but not to a global one
- ▶ improved by multiple restarts

Algorithm

select Γ as a random subset of \mathcal{D}

repeat

$$z_i \leftarrow \arg \min_{k \in \{1, \dots, K\}} d(\mathbf{X}_i, \gamma_k)$$

$$\gamma_k \leftarrow \arg \min_{\gamma \in \mathcal{D}} \sum_{i, z_i=k} d(\mathbf{X}_i, \gamma)$$

until convergence

- ▷ assignment phase
- ▷ representation phase

Comments

- ▶ one of the most well known clustering algorithm for arbitrary dissimilarities
- ▶ complexity $\Theta(NK) + \Theta(N^2)$ (assuming d is known)
- ▶ complex quantization effects when N is small (“data holes”)

When $\mathcal{X} = \mathbb{R}^P$

- ▶ one uses in general $d(\mathbf{X}_i, \mathbf{X}_j) = \|\mathbf{X}_i - \mathbf{X}_j\|^2$
- ▶ then we do need to restrict γ to elements of \mathcal{D}
 - ▶ we have

$$\mathcal{E}(\Gamma, \mathbf{z}) = \sum_{k=1}^K \sum_{i, z_i=k} \|\mathbf{X}_i - \gamma_k\|^2$$

- ▶ and thus

$$\arg \min_{\Gamma} \mathcal{E}(\Gamma, \mathbf{z}) = \left(\frac{1}{s_k} \sum_{i, z_i=k} \mathbf{X}_i \right)_{1 \leq k \leq K},$$

where $s_k = |\{i | z_i = k\}|$

- ▶ identical results when \mathcal{X} is a Hilbert space (a RHKS for instance)

K-means algorithm

Algorithm

select Γ as a random subset of \mathcal{D}

repeat

$$z_j \leftarrow \arg \min_{k \in \{1, \dots, K\}} \|\mathbf{x}_j - \gamma_k\|^2$$

$$\gamma_k \leftarrow \frac{1}{s_k} \sum_{i, z_i=k} \mathbf{x}_i$$

until convergence

▷ assignment phase

▷ representation phase

Comments

- ▶ one of the most well known clustering algorithm
- ▶ complexity $\Theta(NKP)$
- ▶ numerous variants and improvements (kmeans++ for instance)

Standard solution

- ▶ random prototypes chosen uniformly at random in the data set without replacement
- ▶ random clusters do not work properly

k-means++ (Arthur & Vassilvitskii, 2007 [1])

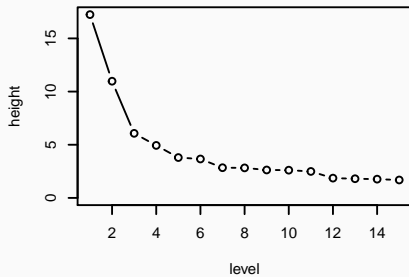
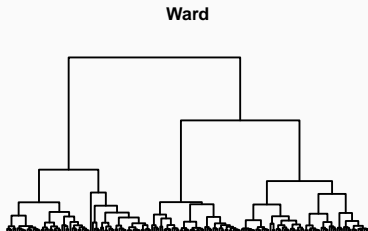
1. γ_1 chosen uniformly at random in the data set
2. for k in 2 to K :

2.1 for $\mathbf{X}_i \in \mathcal{D} \setminus \{\gamma_1, \dots, \gamma_{k-1}\}$ compute $p_i = \frac{\min_{k \in \{1, \dots, k-1\}} d(\mathbf{X}_i, \gamma_k)^2}{\sum_{j \neq i} \min_{k \in \{1, \dots, k-1\}} d(\mathbf{X}_j, \gamma_k)^2}$

2.2 chose γ_k in $\mathcal{D} \setminus \{\gamma_1, \dots, \gamma_{k-1}\}$ according to the probabilities $(p_i)_i$

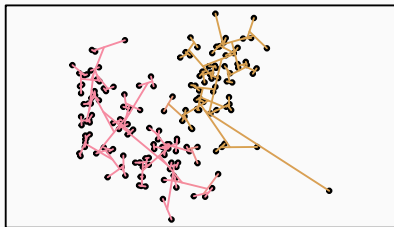
- ▶ theoretical guarantees
- ▶ practical efficiency

Example

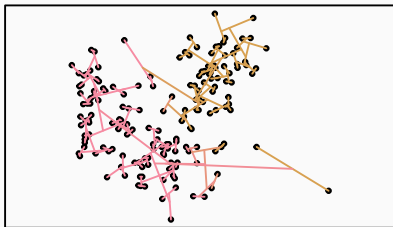


Example

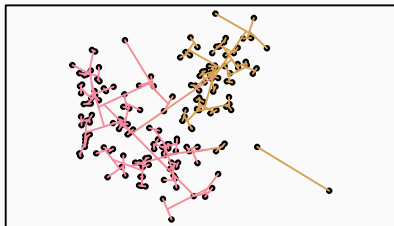
Ward



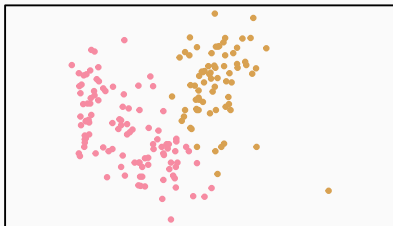
Diana



Average

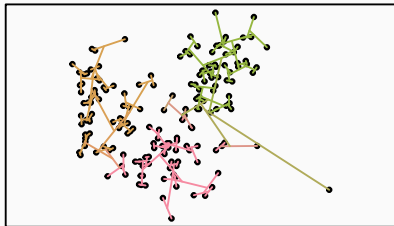


K-means

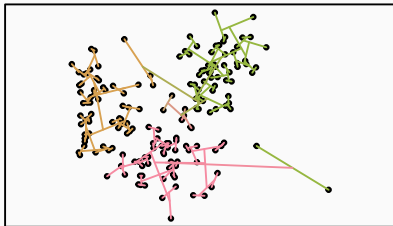


Example

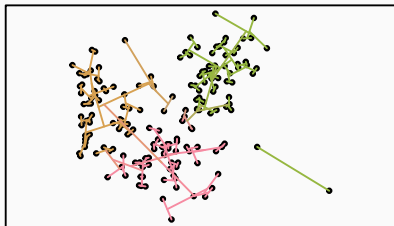
Ward



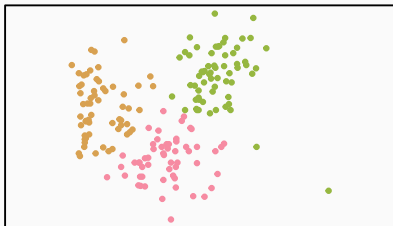
Diana



Average

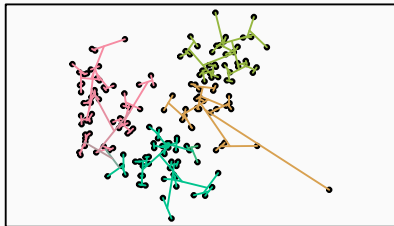


K-means

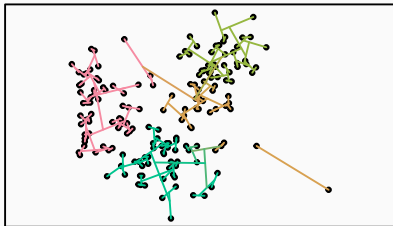


Example

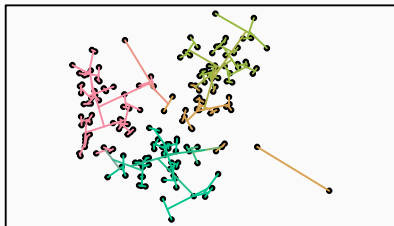
Ward



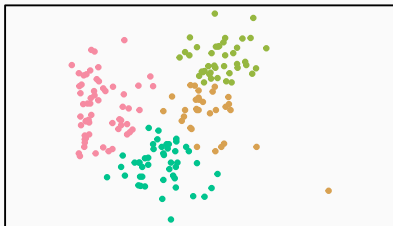
Diana



Average

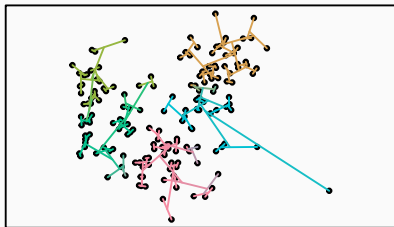


K-means

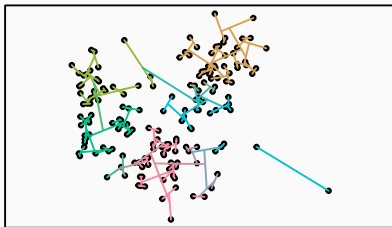


Example

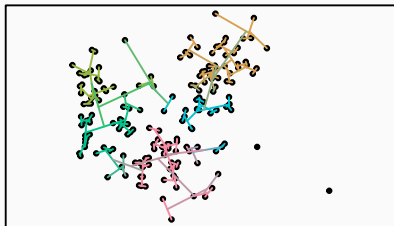
Ward



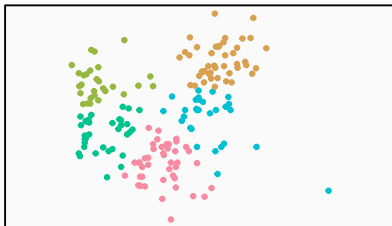
Diana



Average

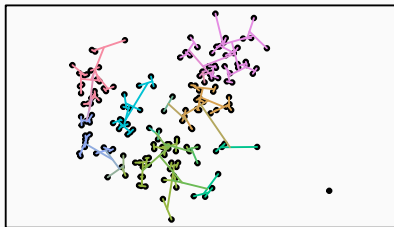


K-means

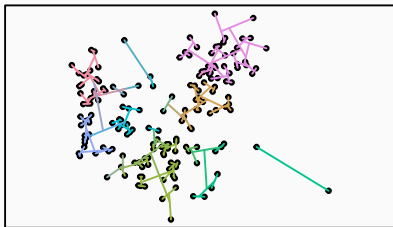


Example

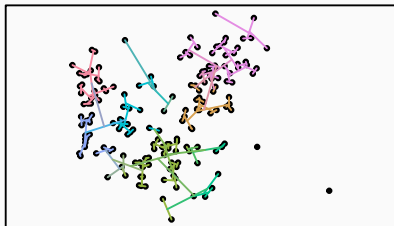
Ward



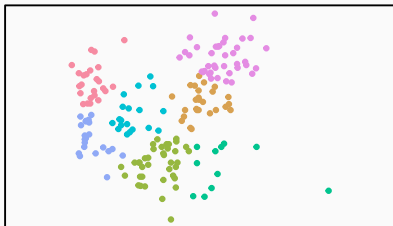
Diana



Average



K-means



Variance decomposition

- ▶ for any $C \subset \{1, \dots, N\}$

$$\sum_{i \in C} \sum_{j \in C} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = 2|C| \sum_{i \in C} \left\| \mathbf{x}_i - \hat{\mathbf{x}}_C \right\|^2,$$

where

$$\hat{\mathbf{x}}_C = \frac{1}{|C|} \sum_{i \in C} \mathbf{x}_i$$

- ▶ and therefore if $C_k = \{i \in \{1, \dots, N\} \mid z_i = k\}$

$$\begin{aligned} \min_{\Gamma} \sum_{i=1}^N \|\mathbf{x}_i - \gamma_{z_i}\|^2 &= \frac{1}{2} \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i \in C_k} \sum_{j \in C_k} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \\ &= \frac{1}{2} W(\{C_1, \dots, C_k\}) \end{aligned}$$

Quantization and within variance

- ▶ minimizing the quantization error is equivalent to maximizing the total within variance
- ▶ K-means is very close to Ward's method!

Total variance

- ▶ the total variance is $\sum_{i=1}^N \left\| \mathbf{x}_i - \hat{\mathbf{x}}_{\mathcal{D}} \right\|^2$
- ▶ we have

$$\begin{aligned} \sum_{i=1}^N \left\| \mathbf{x}_i - \hat{\mathbf{x}}_{\mathcal{D}} \right\|^2 &= \min_{\mathbf{r}} \sum_{i=1}^N \left\| \mathbf{x}_i - \hat{\mathbf{x}}_{C_{\gamma z_i}} \right\|^2 + \sum_{k=1}^K |C_k| \left\| \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{\mathcal{D}} \right\|^2 \\ &= \frac{1}{2} W(\{C_1, \dots, C_k\}) + \frac{1}{2} \sum_{k=1}^K \sum_{k'=1}^K |C_k| |C_{k'}| \left\| \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k'} \right\|^2 \end{aligned}$$

Between variance

- ▶ $B(C_1, \dots, C_k) = \frac{1}{2} \sum_{k=1}^K \sum_{k'=1}^K |C_k| |C_{k'}| \left\| \hat{\mathbf{X}}_k - \hat{\mathbf{X}}_{k'} \right\|^2$
- ▶ weighted pairwise distances between prototypes
- ▶ measures how spread the clusters are

Within and between variance

- ▶ total variance = within variance + between variance
- ▶ the total variance does not depend on the clustering
- ▶ by minimizing the within variance, one maximizes the between variance!
- ▶ clusters are both compact and well separated (at least at the prototype level)

K-means

- 😊 clear quantification interpretation
- 😊 compact clusters with separated prototypes
- 😊 $\Theta(NKP)$ running time
- 😊 efficient initialization strategy (k-means++)
- 😞 $K?$
- 😞 spherical clusters which might be very close one to another

Introduction

Hierarchical clustering

K-means and related methods

DBSCAN

Fuzzy and probabilistic models

Density based clustering

Cluster = dense region

- ▶ clusters are areas of high density compared to other areas
- ▶ density based separation: clusters are separated by low density areas
- ▶ no direct assumption on cluster shape and on relative distances

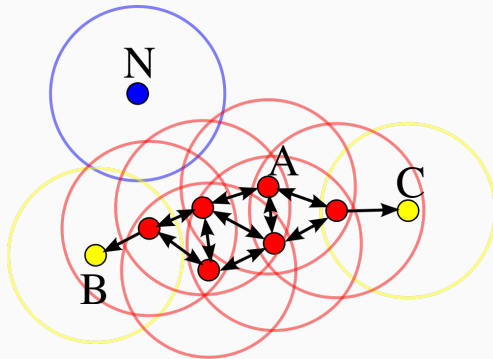
DBSCAN

- ▶ Ester, Kriegel, Sander & Xu, 1996 [2]
- ▶ most well known density based algorithm
- ▶ simple density model:
 - ▶ enough points in a tight region
 - ▶ two parameters:
 - ▶ minPts: minimal number of points in a region
 - ▶ ϵ : radius of the region

Several types of points

- ▶ *core point*: \mathbf{X}_i such that $|\{\mathbf{X} \in \mathcal{D} \mid d(\mathbf{X}_i, \mathbf{X}) \leq \varepsilon\}| \geq \text{minPts}$
- ▶ *directly density reachable point* from a core point \mathbf{X}_i : \mathbf{X}_j such that $d(\mathbf{X}_i, \mathbf{X}_j) \leq \varepsilon$
- ▶ *density reachable point* from a core point \mathbf{X}_i : \mathbf{X}_j such that there is a chain of core points $\mathbf{X}_{k_1}, \dots, \mathbf{X}_{k_t}$ with $d(\mathbf{X}_{k_t}, \mathbf{X}_{k_{t+1}}) \leq \varepsilon$, $\mathbf{X}_{k_1} = \mathbf{X}_i$ and $\mathbf{X}_{k_t} = \mathbf{X}_j$
- ▶ *border point*: *density reachable points* that are not core points
- ▶ *density connected points*: two points are density connected if they are density reachable from the same (core) point
- ▶ *noise point*: points that are not density reachable from a core point

Types of points



- ▶ A and red points: core points
- ▶ B and C: border points
- ▶ N: noise point

illustration from <https://commons.wikimedia.org/wiki/File:DBSCAN-Illustration.svg>

DBSCAN clusters

A cluster in DBSCAN is a maximal set of density connected points.

Noise

Points that do not belong to DBSCAN clusters form the noise.

Clustering model

- ▶ DBSCAN produces a partition of \mathcal{D} into C_1, \dots, C_K, N , where the C_k are the clusters and N is the noise
- ▶ K is not specified directly but only a consequence of minPts and ε
- ▶ notice that $K \leq \frac{N}{\text{minPts}}$ as a cluster must contain a core point and has therefore a minimal size of minPts
- ▶ border points can belong to several clusters and a tie breaking criterion is used to assign them to a single one

Algorithm

```
for all  $\mathbf{X} \in \mathcal{D}$  do  
  if  $\mathbf{X}$  is not labelled then  
     $N \leftarrow \varepsilon$ -neighborhood of  $\mathbf{X}$   
    if  $|N| < minPts$  then  
      label  $\mathbf{X}$  as noise  
    else  
      label  $\mathbf{X}$  with a new cluster label  $k$   
      label with  $k$  all density reachable points from  $\mathbf{X}$  (including  
noise ones)  
    end if  
  end if  
end for
```

Complexity

- ▶ core operation: ε -neighborhood calculation
- ▶ done once for each point in \mathcal{D}
- ▶ naive complexity in $N \Rightarrow \Theta(N^2)$
- ▶ spatial indexing (R* tree, for instance) might decrease the cost but **not to** $\Theta(N \log N)$
- ▶ minimal cost: $\Theta(N^{\frac{4}{3}})$

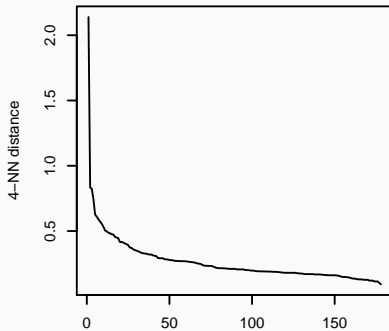
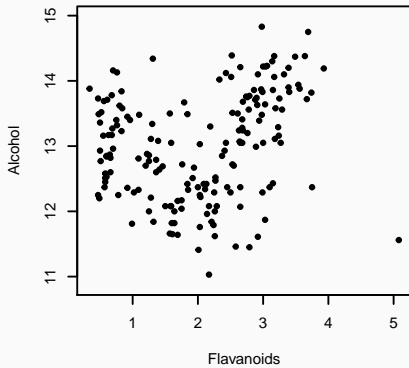
minPts

- ▶ limited impact on the results above a minimal value
- ▶ original recommendation: $minPts = 4$
- ▶ current recommendation: $minPts = 2 \times P$ (larger values for noisy data)

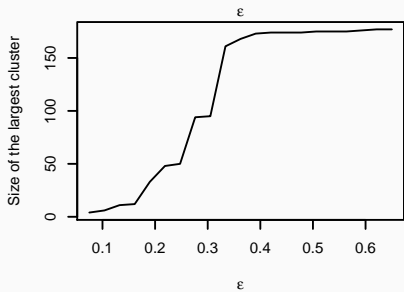
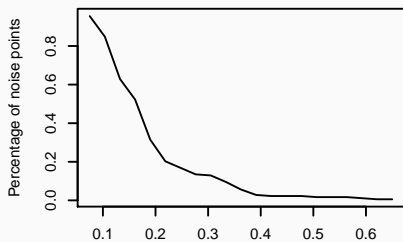
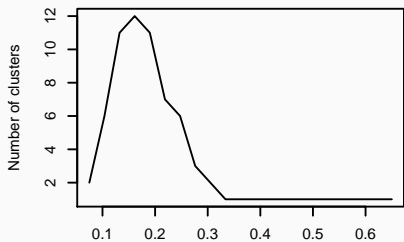
ϵ

- ▶ very difficult to set
- ▶ plays a role similar to the one of k in the k -means
- ▶ one should explore the effects of using different values of ϵ
- ▶ “elbow” approach on the k -nn distance graph

Example

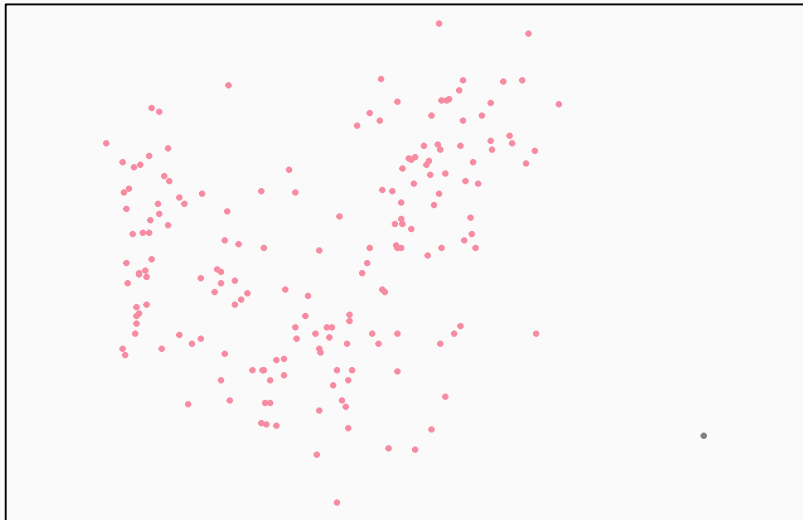


Diagnostic plots



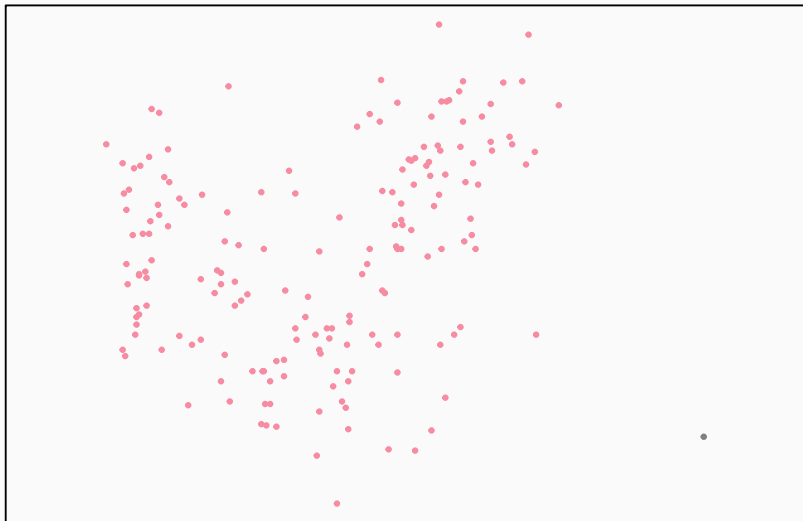
Clusters

$\epsilon = 0.65$ $k = 1$ noise = 1



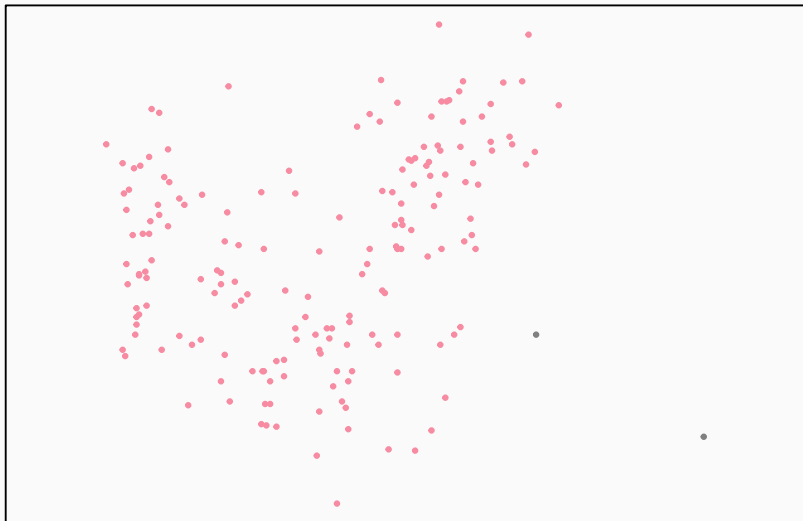
Clusters

$\epsilon = 0.62125$ $k = 1$ noise = 1



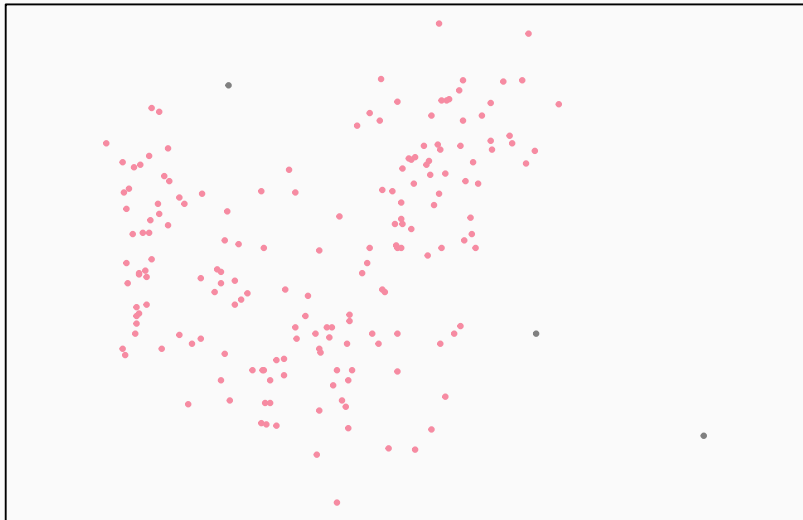
Clusters

$\epsilon = 0.5925$ $k = 1$ noise = 2



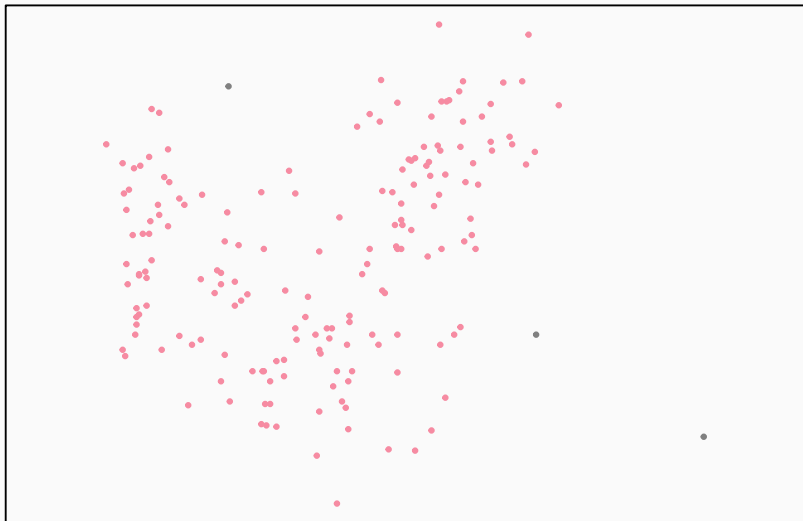
Clusters

$\epsilon = 0.56375$ $k = 1$ noise = 3



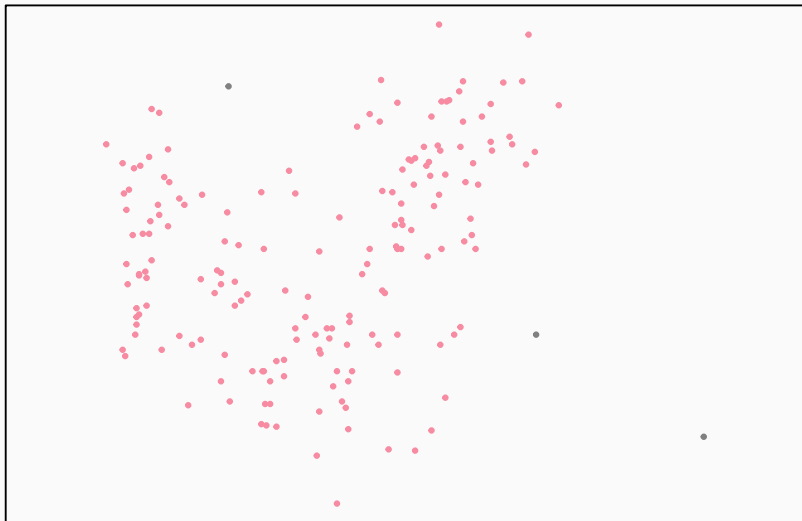
Clusters

$\epsilon = 0.535$ $k = 1$ noise = 3



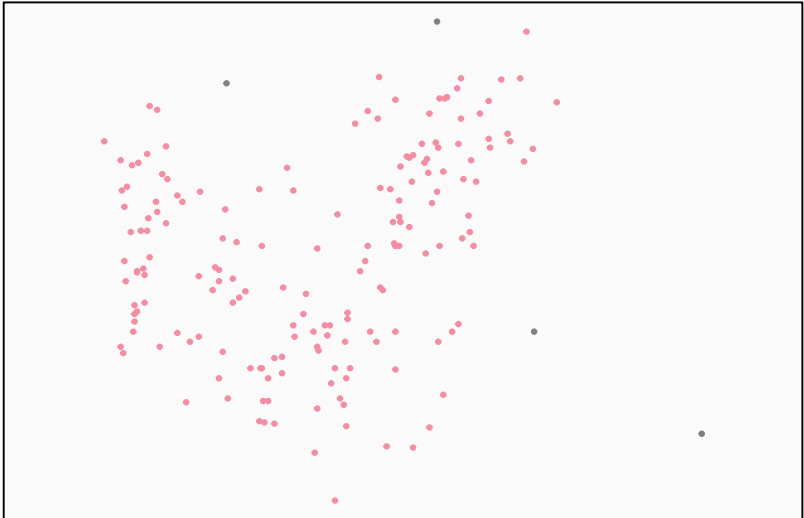
Clusters

$\epsilon = 0.50625$ $k = 1$ noise = 3



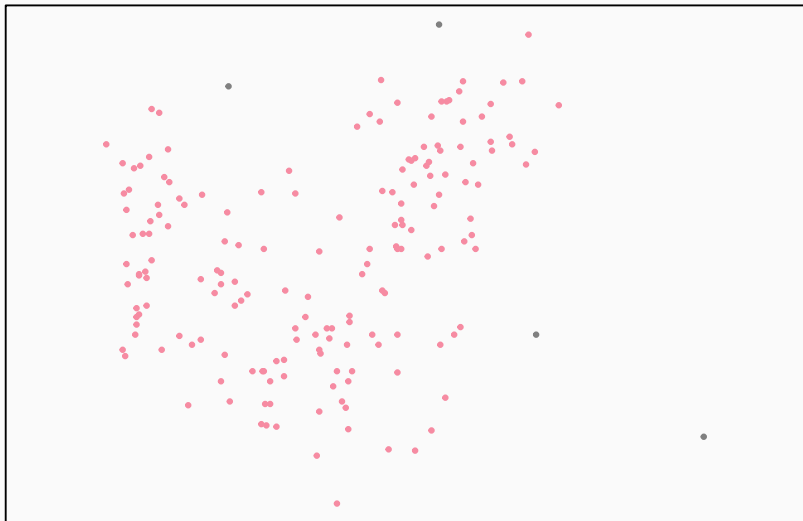
Clusters

$\epsilon = 0.4775$ $k = 1$ noise = 4



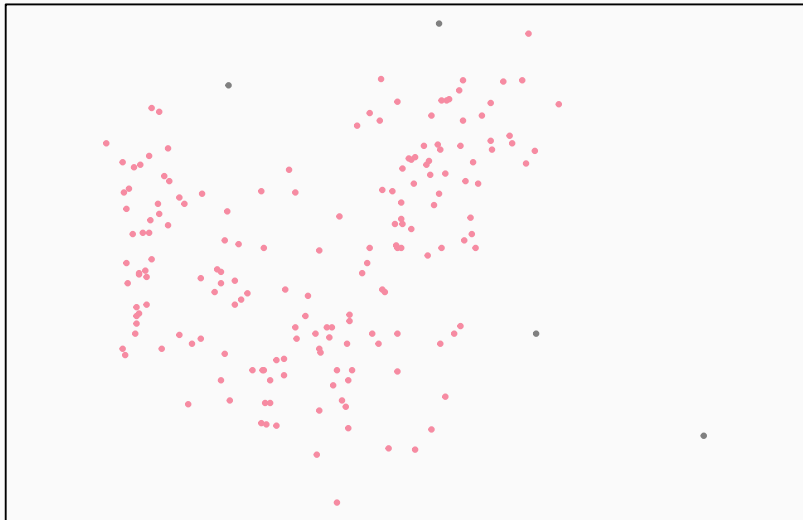
Clusters

$\epsilon = 0.44875$ $k = 1$ noise = 4



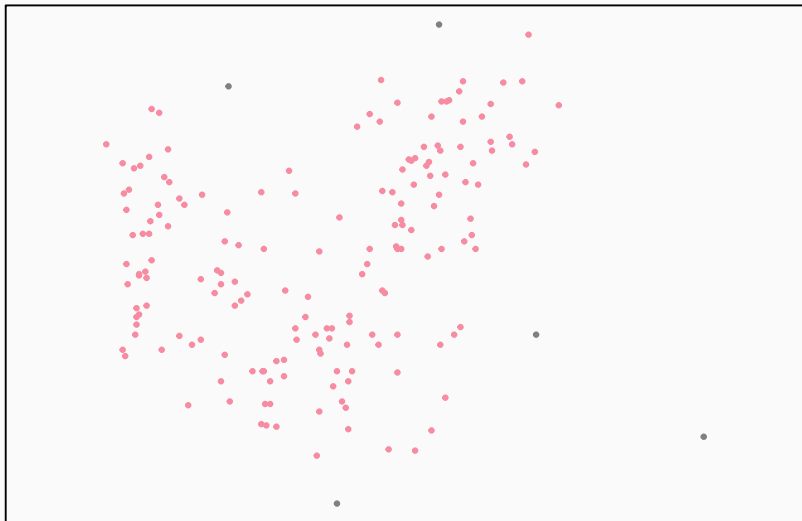
Clusters

$\epsilon = 0.42$ $k = 1$ noise = 4



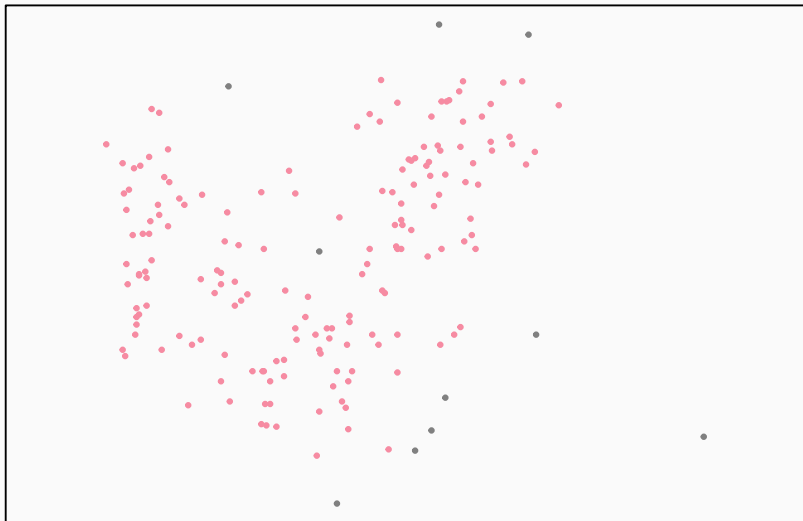
Clusters

$\epsilon = 0.39125$ $k = 1$ noise = 5



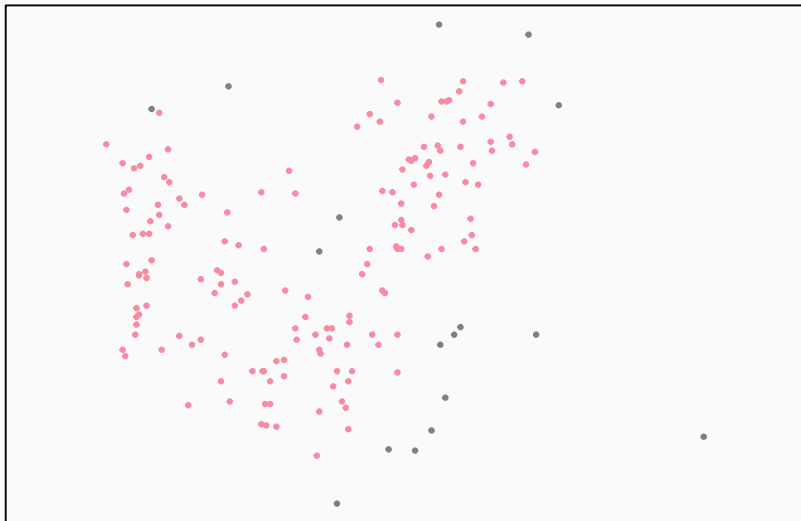
Clusters

$\epsilon = 0.3625$ $k = 1$ noise = 10



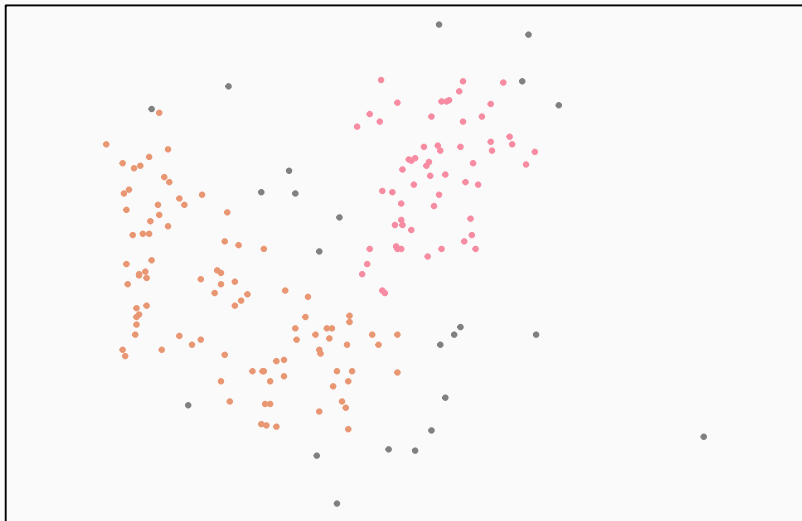
Clusters

$\epsilon = 0.33375$ $k = 1$ noise = 17



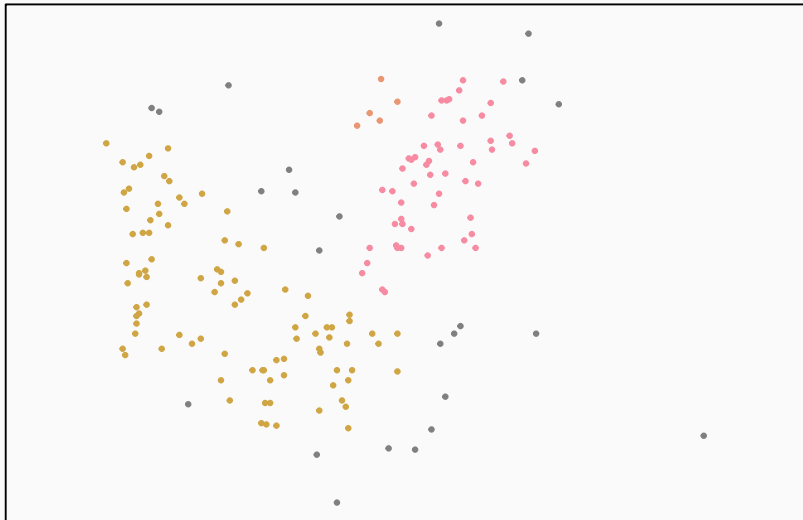
Clusters

$\epsilon = 0.305$ $k = 2$ noise = 23



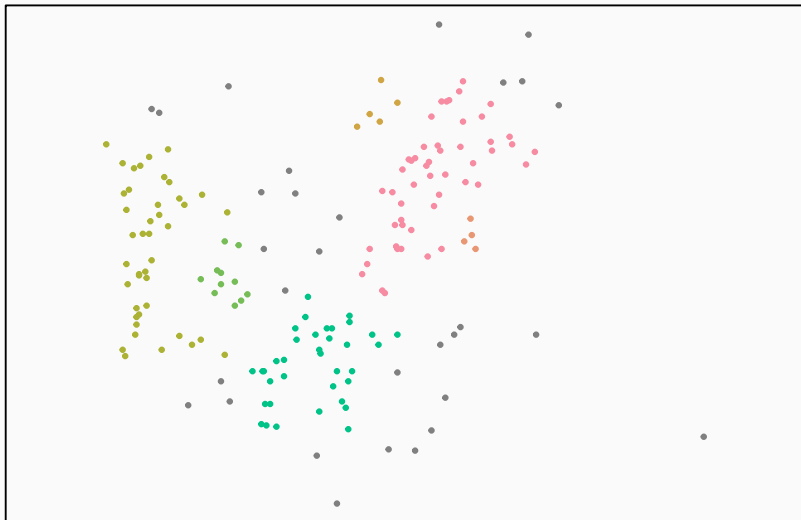
Clusters

$\epsilon = 0.27625$ $k = 3$ noise = 24



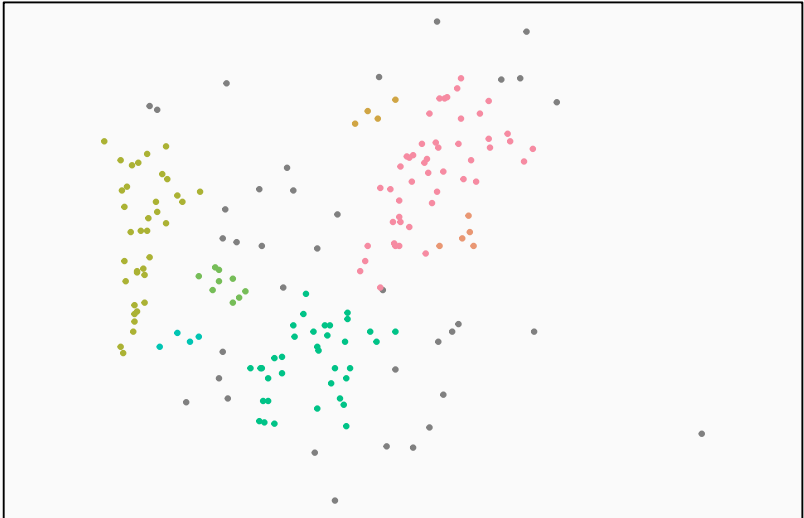
Clusters

$\epsilon = 0.2475$ $k = 6$ noise = 30



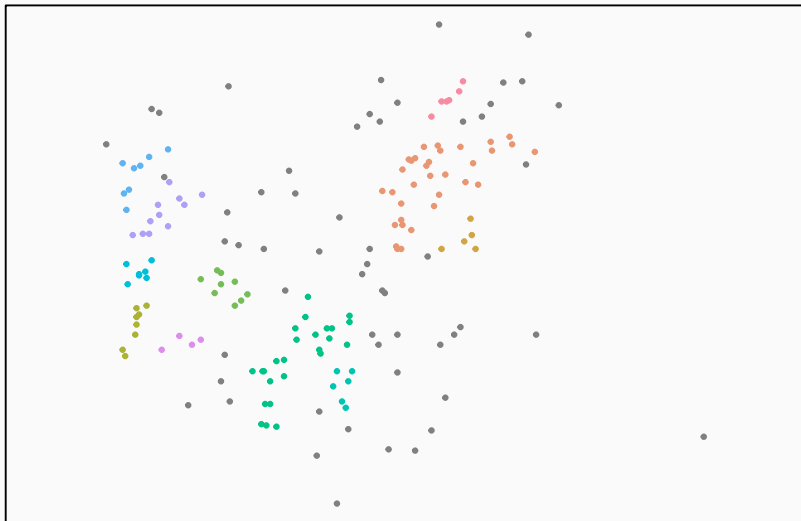
Clusters

$\epsilon = 0.21875$ $k = 7$ noise = 36



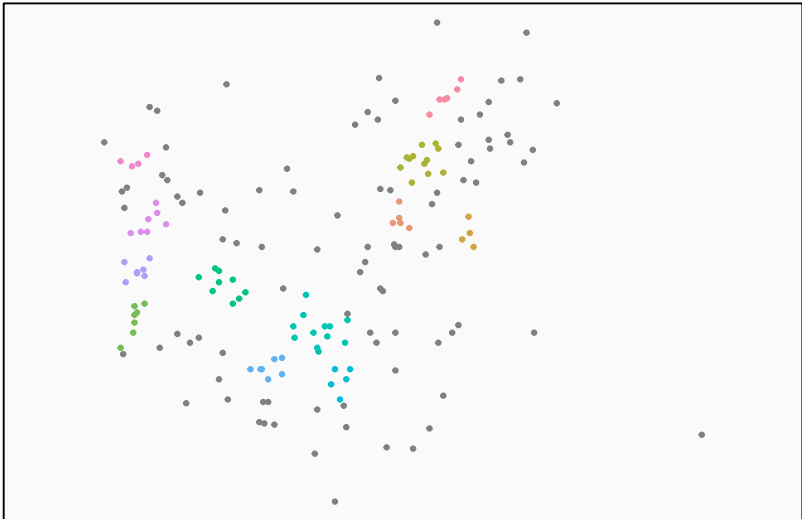
Clusters

$\epsilon = 0.19$ $k = 11$ noise = 56



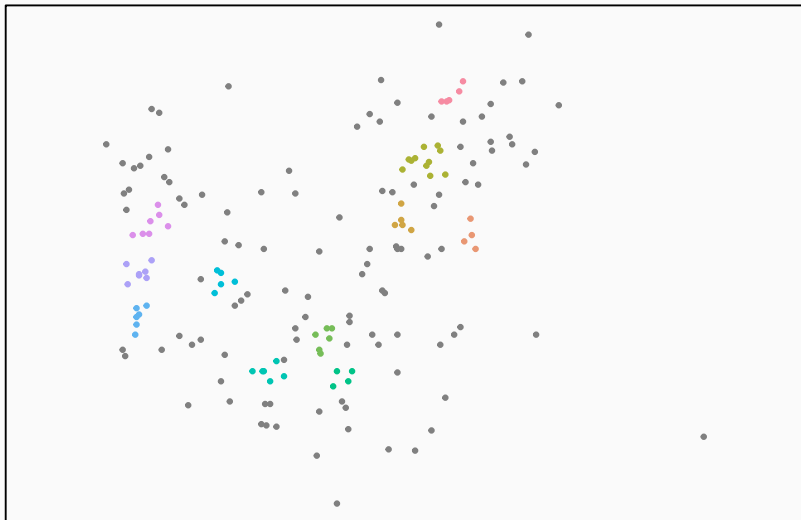
Clusters

$\epsilon = 0.16125$ $k = 12$ noise = 93



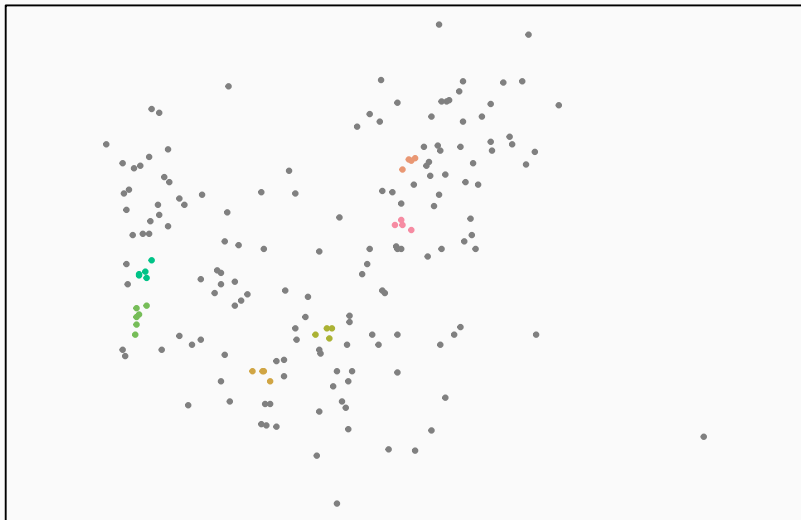
Clusters

$\epsilon = 0.1325$ $k = 11$ noise = 112



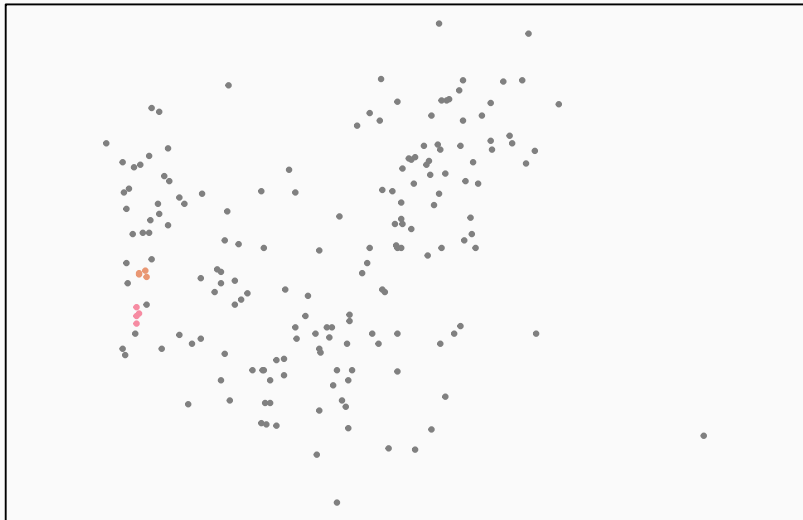
Clusters

$\epsilon = 0.10375$ $k = 6$ noise = 151



Clusters

$\epsilon = 0.075$ $k = 2$ noise = 170



DBSCAN

- 😊 noise detection
- 😊 arbitrary shaped clusters
- 😊 mostly deterministic
- 😞 $\Theta(N^2P)$ running time in the worst case
- 😞 arbitrary shaped clusters
- 😞 very sensitive to the value of ϵ

Introduction

Hierarchical clustering

K-means and related methods

DBSCAN

Fuzzy and probabilistic models

Ambiguous points

- ▶ some points are difficult to associate to a given cluster
- ▶ numerous situations: very close clusters, “interpolation” points, etc.

Soft clusters

- ▶ \mathbf{X}_i belongs to C_k with “intensity”
- ▶ clustering through an assignment matrix: $(M_{ik})_{1 \leq i \leq N, 1 \leq k \leq K}$, with
 - ▶ $M_{ik} \in [0, 1]$: intensity between 0 and 1
 - ▶ $\sum_{k=1}^K M_{ik} = 1$: total unitary grade
 - ▶ crisp limit: $M_{ik} \in \{0, 1\}$
- ▶ interpretations:
 - ▶ M_{ik} as a membership grade: fuzzy sets
 - ▶ M_{ik} as a membership probability: mixture models

Vector quantization with membership

- ▶ prototypes: $\Gamma = (\gamma_1, \dots, \gamma_K)$
- ▶ assignment matrix: $(M_{ik})_{1 \leq i \leq N, 1 \leq k \leq K}$
- ▶ quality criterion

$$\mathcal{E}^b(\Gamma, \mathbf{z}) = \sum_{i=1}^N \sum_{k=1}^K M_{ik}^b d(\mathbf{x}_i, \gamma_k),$$

Fuzziness parameter

- ▶ b represents the non crispness of the assignment
- ▶ $b = 1$ corresponds to the standard quantization problem
- ▶ $b > 1$ generates fuzzy assignments

Fuzzy c-means

Principle

- ▶ $\mathcal{E}^b(\Gamma, \mathbf{z})$ is easily optimized with alternate optimization
- ▶ constrained optimization with respect to M

Algorithm

for $d(\mathbf{X}_i, \gamma_k) = \|\mathbf{X}_i - \gamma_k\|^2$

select Γ as a random subset of \mathcal{D}

repeat

compute $d_{ik} = \|\mathbf{X}_i - \gamma_k\|^2$

$$M_{ik} = \frac{(1/d_{ik})^{1/(b-1)}}{\sum_{j=1}^K (1/d_{ij})^{1/(b-1)}}$$

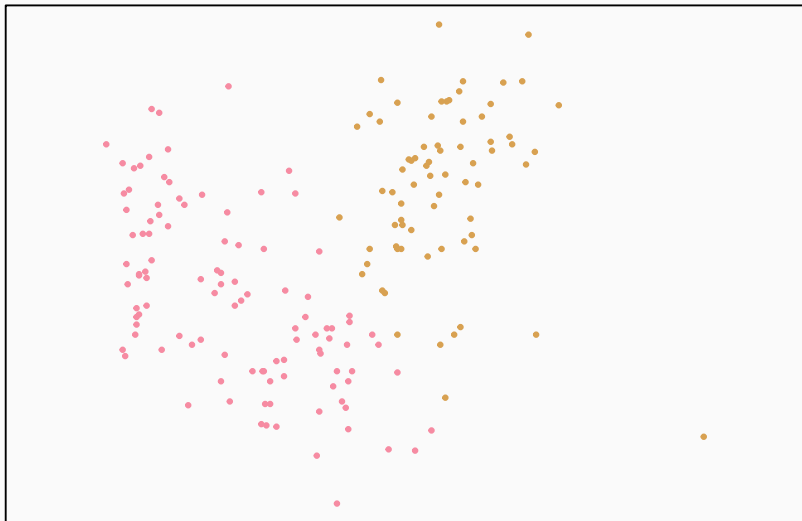
▶ assignment phase

$$\gamma_k \leftarrow \frac{1}{\sum_{j=1}^N M_{jk}^b} \sum_{i=1}^N M_{ik}^b \mathbf{X}_i$$

▶ representation phase

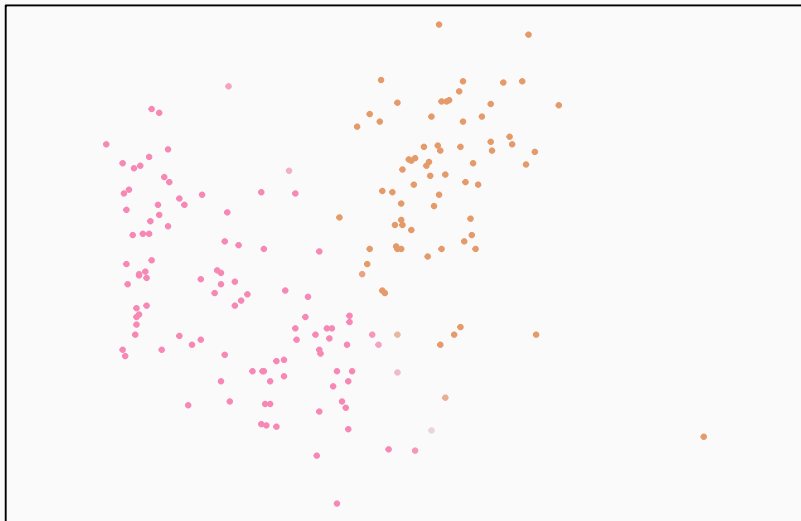
until convergence

Standard K-means



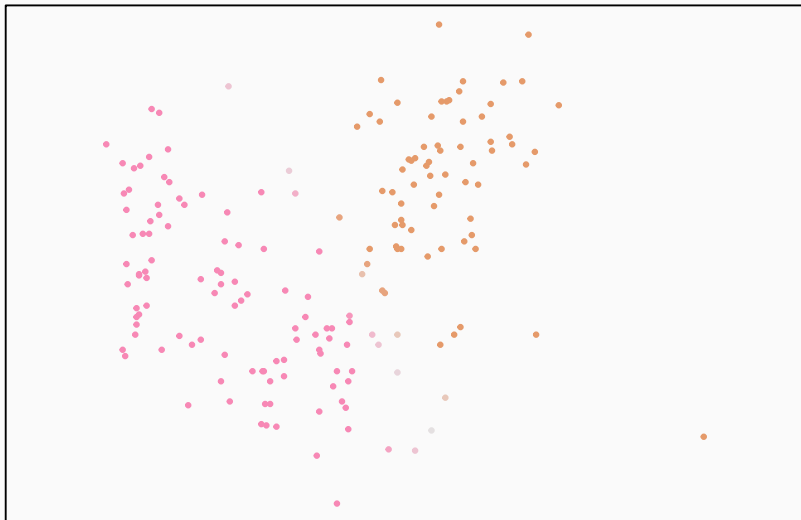
Examples

$b = 1.1$



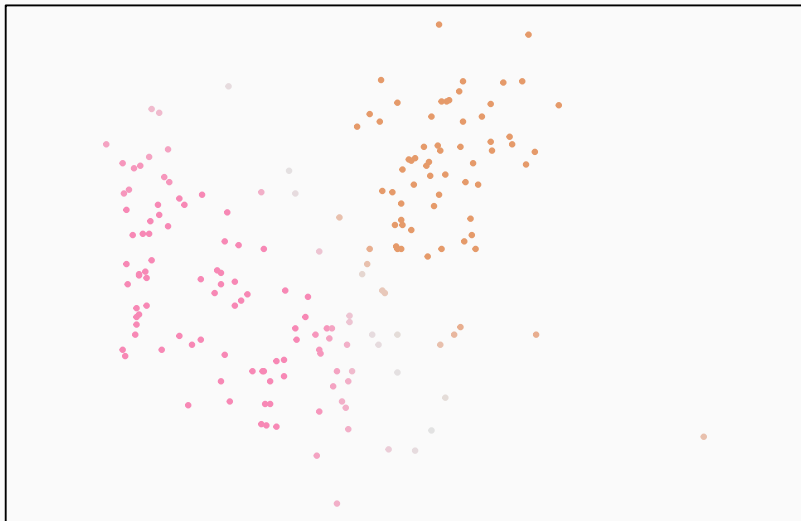
Examples

$b = 1.2$



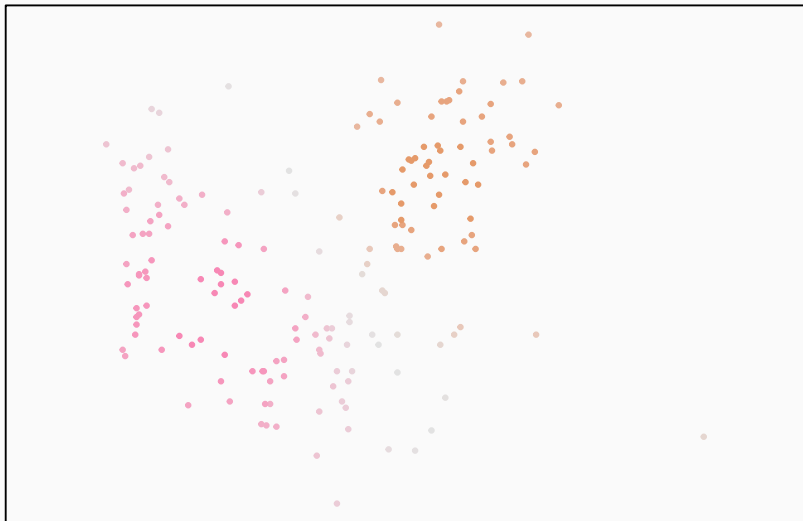
Examples

$b = 1.5$



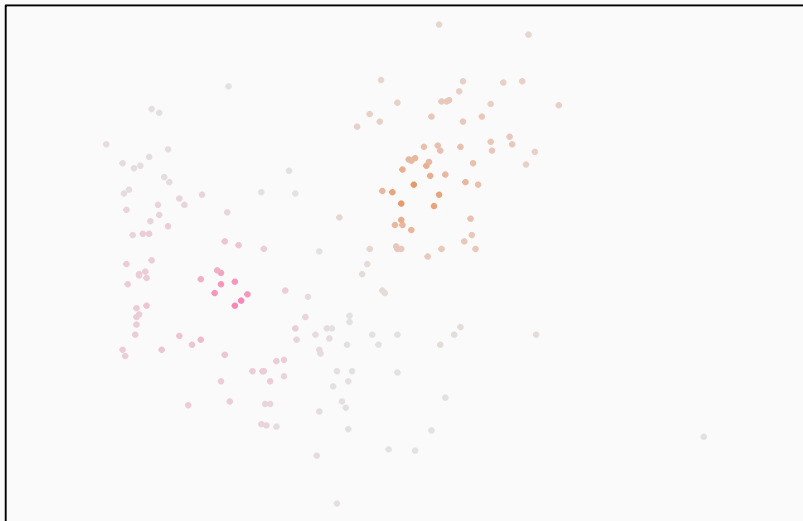
Examples

$b = 2$



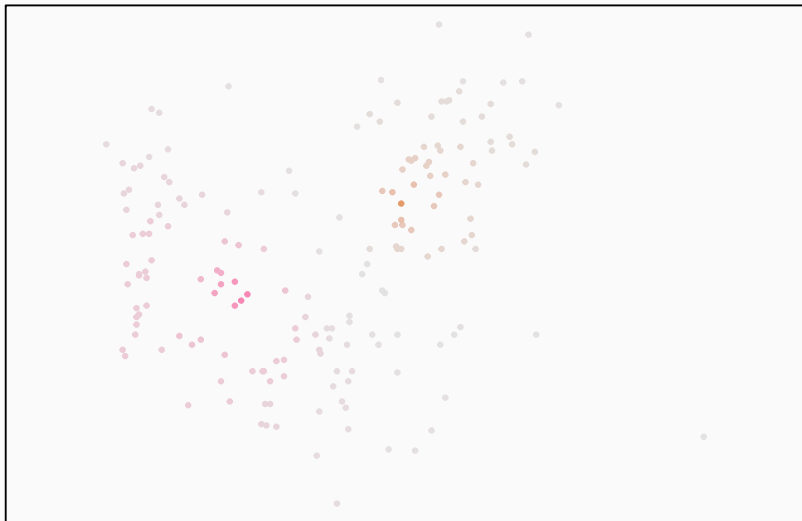
Examples

$b = 5$

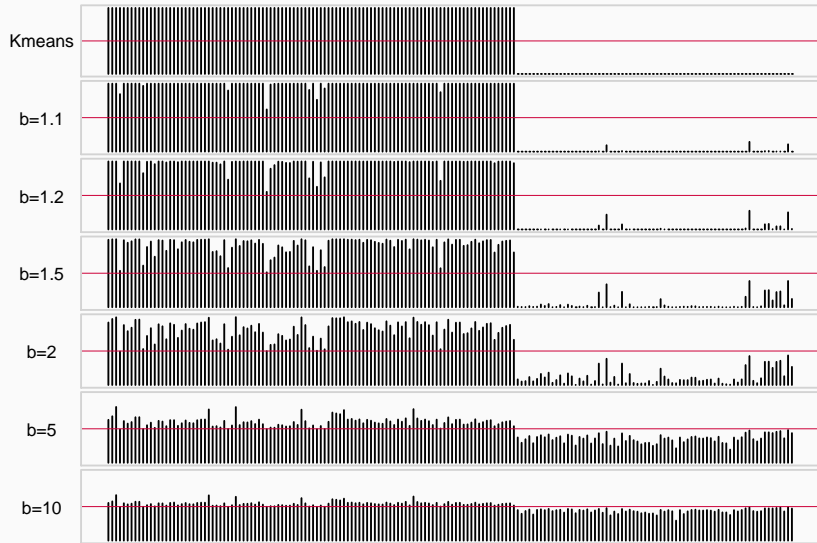


Examples

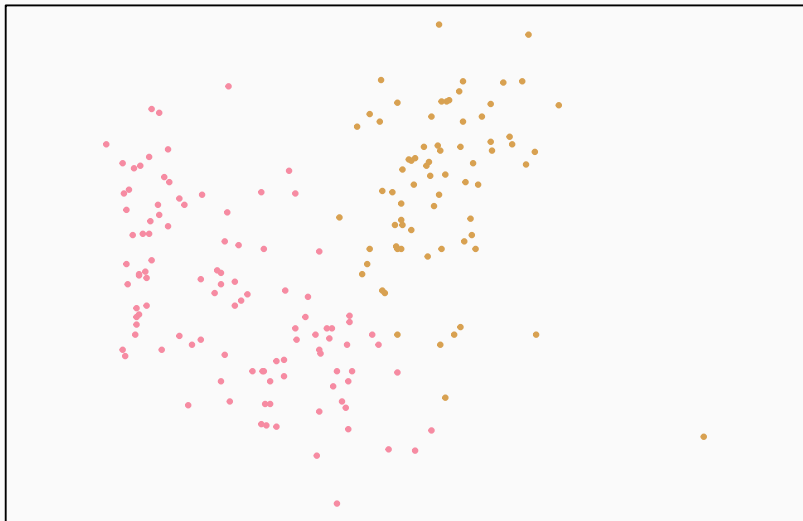
$b = 10$



Fuzziness evolution

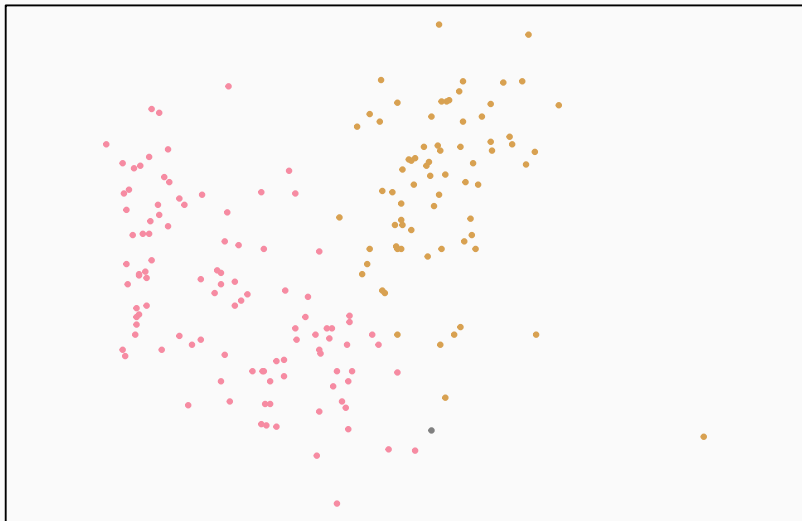


Standard K-means



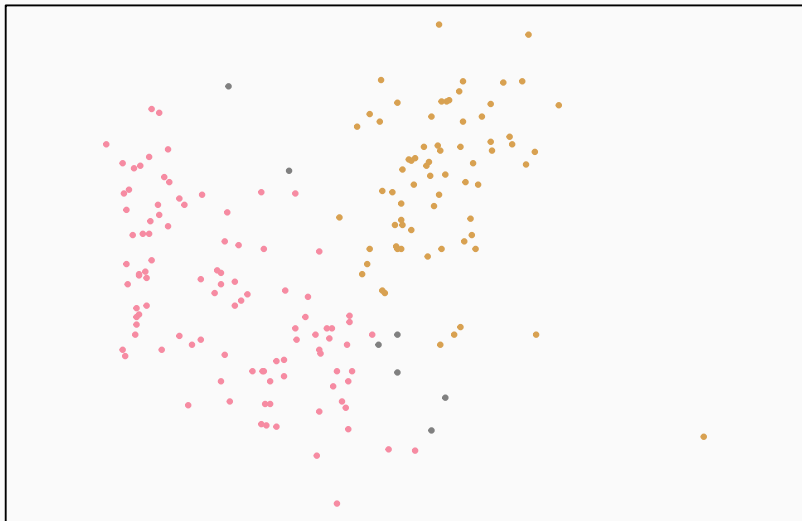
Defuzzified clustering

$b = 1.1$



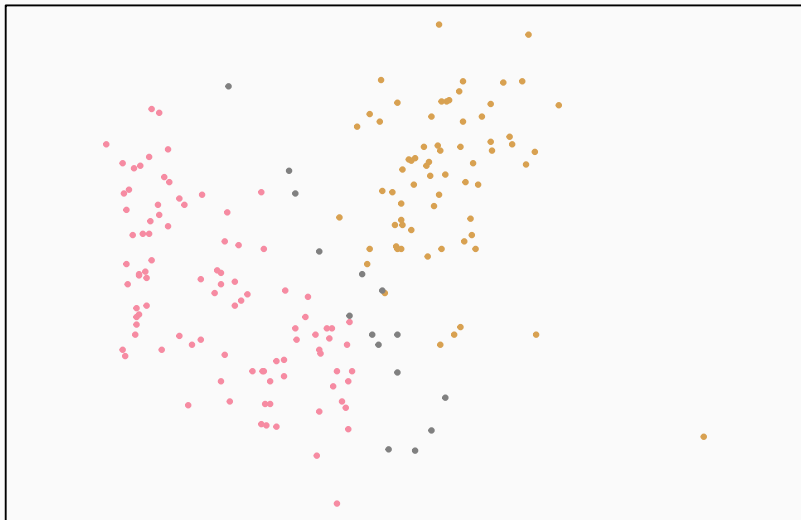
Defuzzified clustering

$b = 1.2$



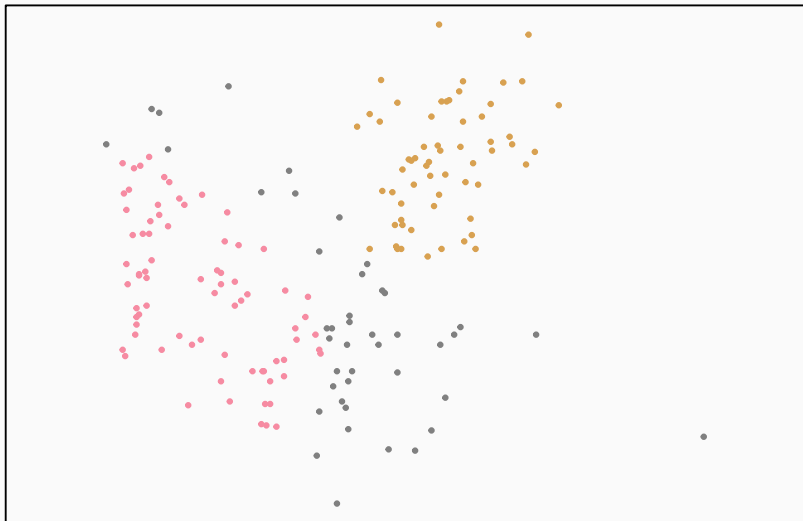
Defuzzified clustering

$b = 1.5$



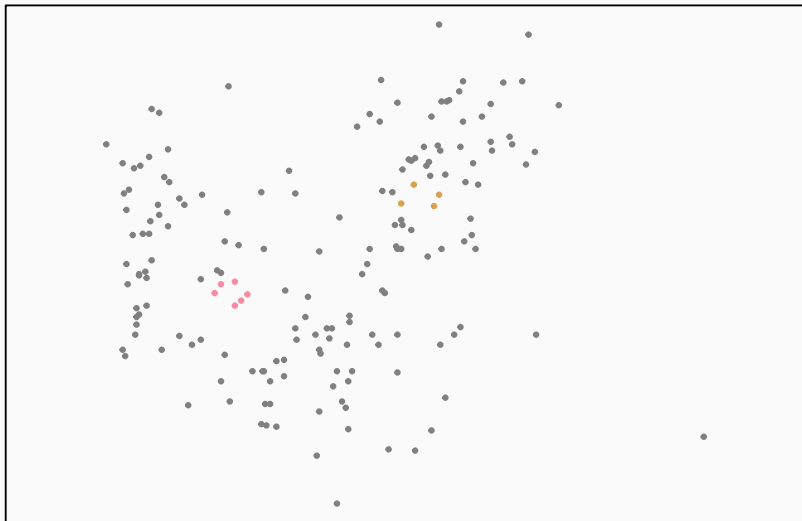
Defuzzified clustering

$b = 2$



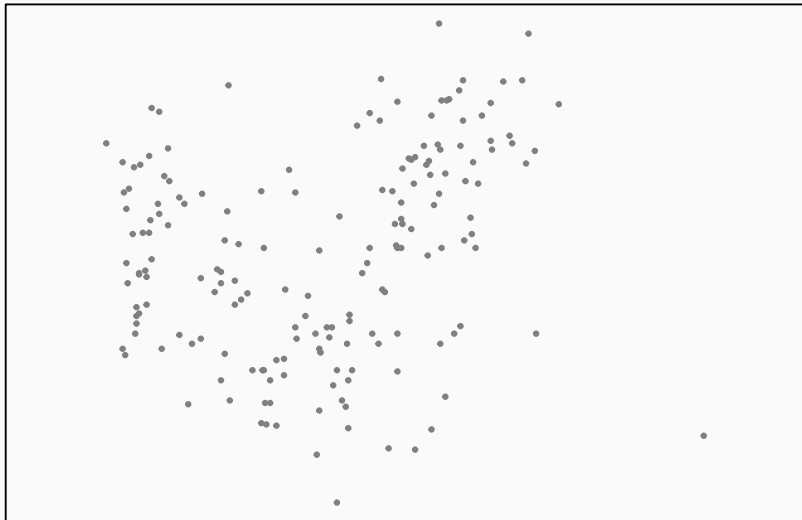
Defuzzified clustering

$b = 5$



Defuzzified clustering

$b = 10$



Fuzzy c-means

- 😊 soft clustering
- 😊 simple interpretation and implementation
- 😊 identify points that are complex to cluster
- 😞 slightly more expensive than the k-means
- 😞 quite sensitive to the additional parameter b

Strategy

- ▶ test several values of b
- ▶ use diagnostic plots
- ▶ can be used to identify core points

M_{ik} as a membership grade

- ▶ intrinsic fuzziness
- ▶ clusters are inherently ill defined
- ▶ no randomness

M_{ik} as a probability

- ▶ missing information
- ▶ belief
- ▶ clusters are perfectly defined but unknown to the analyst

Generative models

- ▶ parametric model for the distribution of $(\mathbf{X}_i)_{1 \leq i \leq N}$
- ▶ parameter estimation from a data set via maximum likelihood (or other techniques)

Clustering oriented models

- ▶ mixture models
- ▶ K parametric models with prior probabilities π_k ($\sum_{k=1}^K \pi_k = 1$)
- ▶ generative process for \mathbf{X}_i
 1. chose $z_i \in \{1, \dots, K\}$ with probability π_k for k
 2. generate \mathbf{X}_i according to the parametric model z_i

Hidden variable model

- ▶ each observation \mathbf{X}_i is associated to a hidden variable Z_i
- ▶ each Z_i takes values in $\{1, \dots, K\}$, with $\mathbb{P}(Z_i = k) = \pi_k$
- ▶ the Z_i are independent and given Z_1, \dots, Z_N , the \mathbf{X}_i are independent
- ▶ we are given K parametric distributions $(p_k)_{1 \leq k \leq K}$ on \mathcal{X} with parameters $\theta = (\theta_k)_{1 \leq k \leq K}$
- ▶ $\mathbf{X}_i \mid Z_i = k$ is distributed according to p_k
- ▶ then the log likelihood of the data set $\mathcal{D} = (\mathbf{x}_i)_{1 \leq i \leq N}$ is given by

$$\log p(\mathcal{D} \mid \pi, \theta) = \sum_{i=1}^N \log \left(\sum_{k=1}^K \pi_k p_k(\mathbf{x}_i \mid \theta_k) \right)$$

Notations

- ▶ integer notation $Z_i \in \{1, \dots, K\}$
- ▶ binary notation $Z_i \in \{0, 1\}^K$ with $\sum_{k=1}^K Z_{ik} = 1$
- ▶ $Z_i = k \Leftrightarrow Z_{ik} = \delta_{ik}$
- ▶ then $p(z_i | \pi) = \prod_{k=1}^K \pi_k^{z_{ik}}$

Complete likelihood

- ▶ the log likelihood of the full data set $\mathcal{D}_F = (\mathbf{x}_i, z_i)_{1 \leq i \leq N}$ is given by

$$\log p(\mathcal{D}_F | \pi, \theta) = \sum_{i=1}^N \sum_{k=1}^K z_{ik} (\log \pi_k + \log p_k(\mathbf{x}_i | \theta_k))$$

- ▶ easy to optimize compare to $\log p(\mathcal{D} | \pi, \theta)$

Reversing the model

- ▶ according to the Bayes rule

$$\mathbb{P}(Z_i = k \mid \mathbf{x}_i, \pi, \theta) = \frac{\rho_k(\mathbf{x}_i \mid \theta_k)\pi_k}{\sum_{l=1}^K \pi_l \rho_l(\mathbf{x}_i \mid \theta_l)}$$

- ▶ $\gamma_{ik} = \mathbb{P}(Z_i = k \mid \mathbf{x}_i, \pi, \theta)$ is the responsibility of component k for generating \mathbf{x}_i
- ▶ z_i can be “guessed” in a probabilistic sense given the true parameters of the model

Key ideas

- ▶ averaging $\sum_{i=1}^N \sum_{k=1}^K z_{ik} (\log \pi_k + \log p_k(\mathbf{x}_i | \theta_k))$ over the probabilistic guesses of the z_i
- ▶ using the best possible guess $\gamma_{ik} = \mathbb{P}(Z_i = k | \pi, \theta)$
- ▶ alternating between improving the estimates of the parameters π and θ and improving the estimates of the hidden variables γ_{ik}

EM algorithm

initialize $\pi^{(0)}$ and $\theta^{(0)}$

$t \leftarrow 1$

repeat

compute

▷ E Phase

$$\gamma_{ik}^{(t)} = \frac{p_k(\mathbf{x}_i | \theta_k^{(t-1)}) \pi_k^{(t-1)}}{\sum_{l=1}^K \pi_l^{(t-1)} p_l(\mathbf{x}_i | \theta_l^{(t-1)})}$$

compute

▷ M Phase

$$N_k^{(t)} = \sum_{i=1}^N \gamma_{ik}^{(t)}$$

$$\pi_k^{(t)} = \frac{N_k^{(t)}}{N}$$

$$\theta_k^{(t)} = \arg \max_{\theta_k} \sum_{i=1}^N \gamma_{ik}^{(t)} \log p_k(\mathbf{x}_i | \theta_k)$$

$t \leftarrow t + 1$

until convergence

Standard approach for $\mathcal{X} = \mathbb{R}^P$

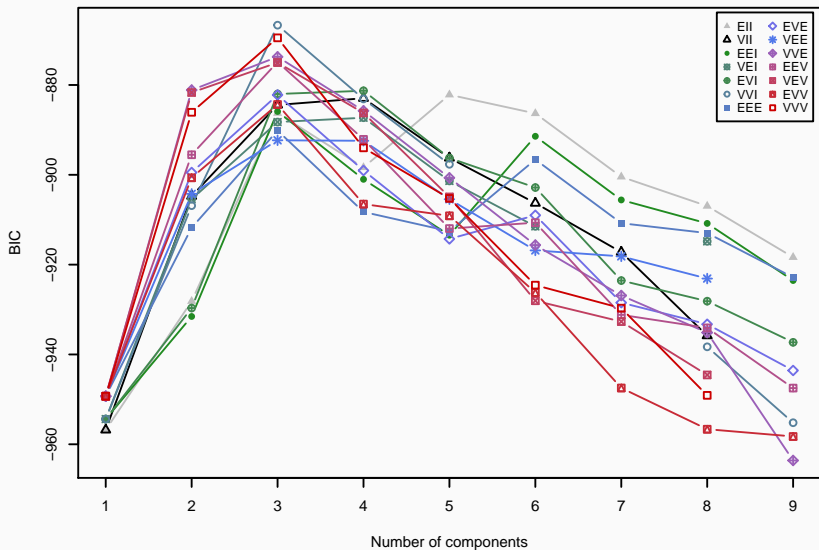
- ▶ each p_k is a multivariate Gaussian distribution
- ▶ $\theta_k = (\mu_k, \Sigma_k)$ and

$$p(\mathbf{x}_i | \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{P/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \mu_k)}$$

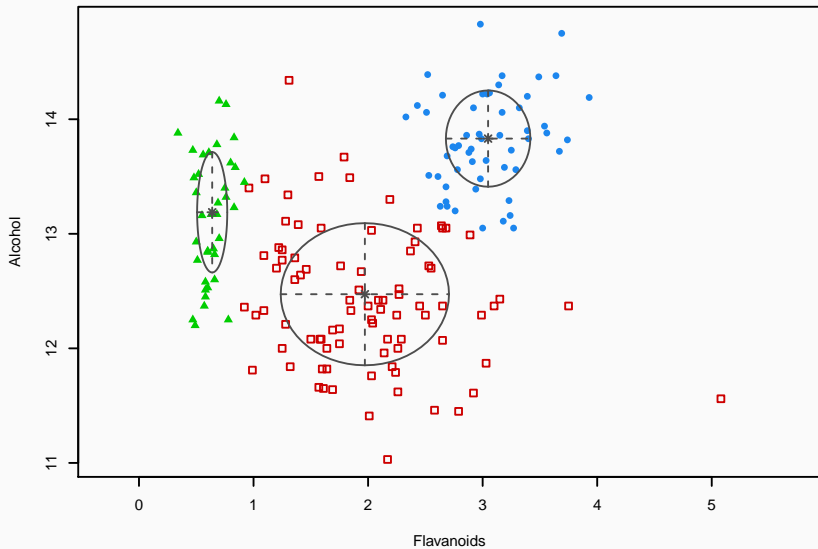
- ▶ then we have

$$\mu_k^{(t)} = \frac{1}{N_k^{(t)}} \sum_{i=1}^N \gamma_{ik}^{(t)} \mathbf{x}_i$$
$$\Sigma_k^{(t)} = \frac{1}{N_k^{(t)}} \sum_{i=1}^N \gamma_{ik}^{(t)} (\mathbf{x}_i - \mu_k^{(t)})^T (\mathbf{x}_i - \mu_k^{(t)})$$

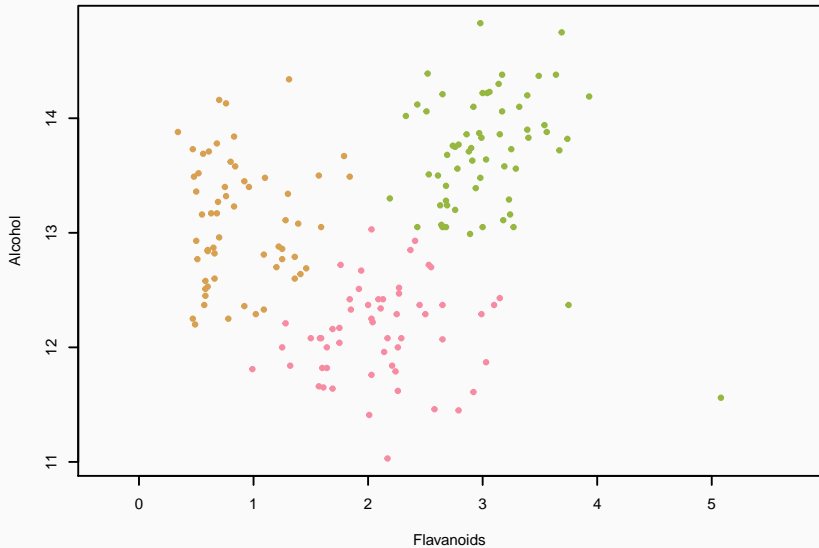
Example



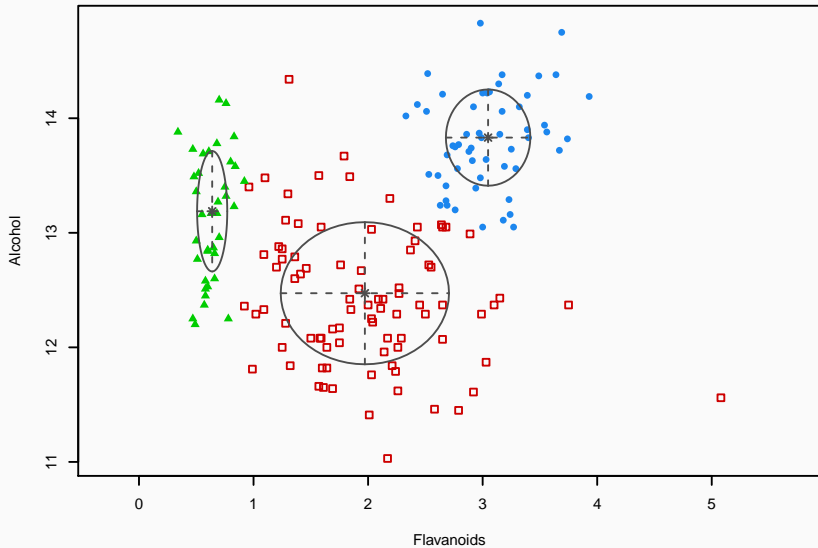
Example



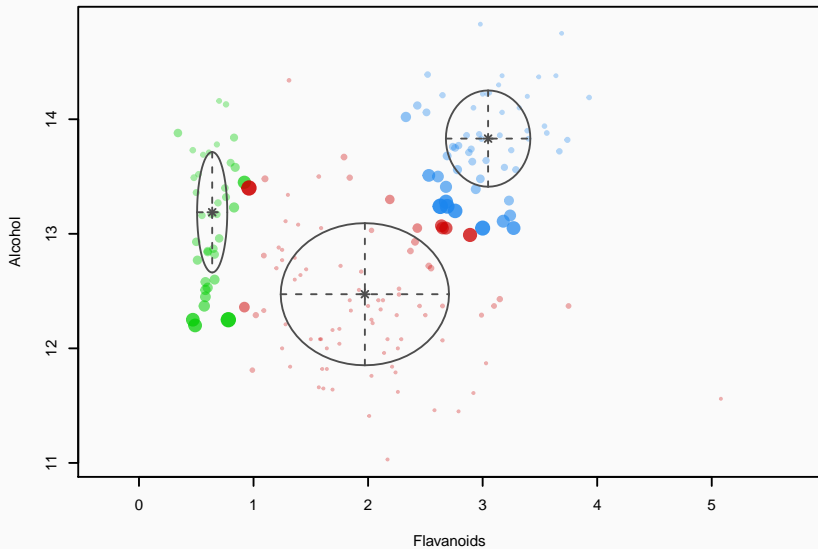
Example



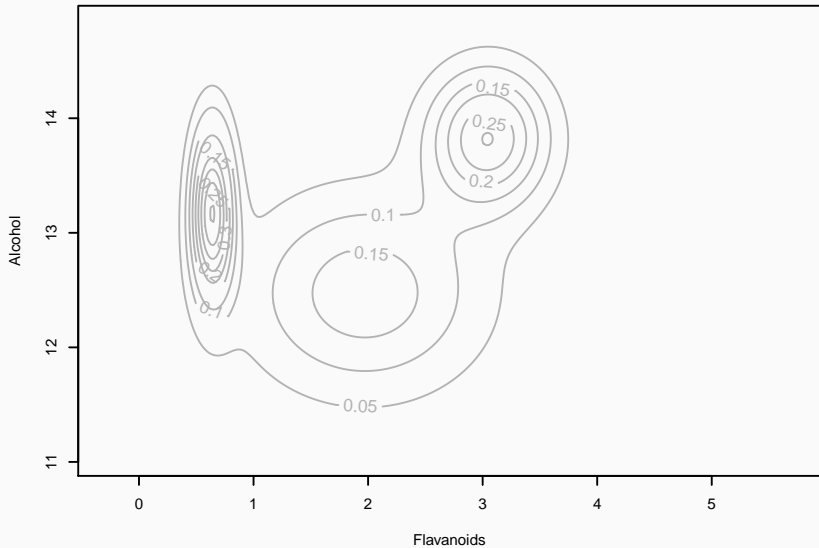
Example



Example



Example



Mixture models

- 😊 soft clustering
- 😊 rich outputs
- 😊 automatic model selection (via BIC)
- 😊 very flexible framework
- 😞 somewhat complex implementation
- 😞 high computational cost in some cases

Conclusion



D. Arthur and S. Vassilvitskii.

K-means++: The advantages of careful seeding.

In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07*, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.



M. Ester, H.-P. Kriegel, J. Sander, and X. Xu.

A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise.

In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96*, pages 226–231. AAAI Press, 1996.



J. M. Kleinberg.

An impossibility theorem for clustering.

In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 463–470. MIT Press, 2003.



U. von Luxburg, R. C. Williamson, and I. Guyon.

Clustering: Science or art?

In I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 65–79, Bellevue, Washington, USA, 02 Jul 2012. PMLR.



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

<http://creativecommons.org/licenses/by-sa/4.0/>

Last git commit: 2021-01-19

By: Fabrice Rossi (Fabrice.Rossi@apiacoa.org)

Git hash: 97cfd0a9975cf193f5790845c00e476c1572a327

- ▶ April 2019:
 - ▶ added k-means++
 - ▶ added DBSCAN
 - ▶ added fuzzy c-means
 - ▶ added mixture models
- ▶ March 2018: initial version