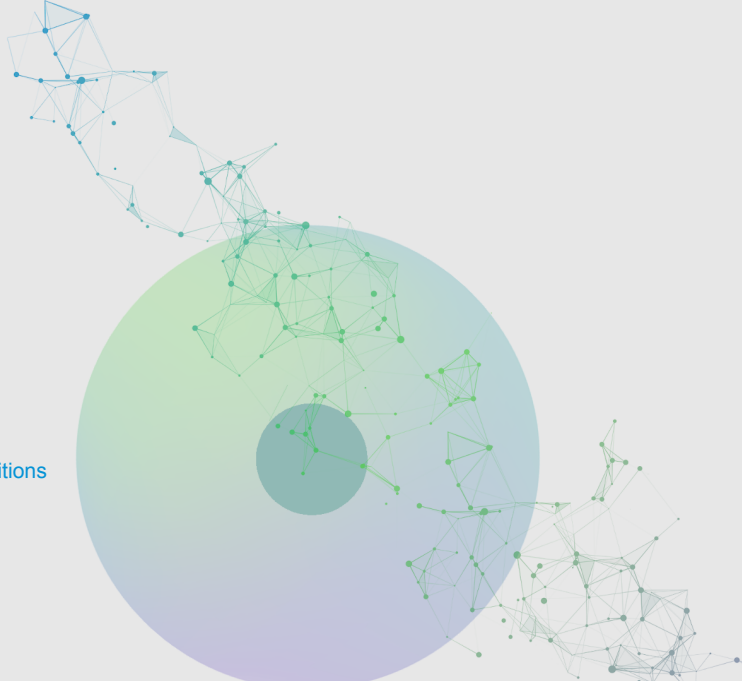
Clustering

Max Toller

CLARIAH-AT Summer School
Machine Learning for Digital Scholarly Editions
2025-09-10



Introduction: Central Tendency

- Datasets can be large
- We are typically not interested in every entry
- Statistics summarize data
- Central tendency $\rightarrow$ What is the “typical” entry
  - What is the typical salary of an ML engineer?
  - What are the typical topics of letters between politicians?

Naive Central Tendency: Mean, Median, Mode

- If our dataset ***x*** has *n* entries…
- Mean: $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$
- Median: $m = x_{(n+1)/2}$ if ***x*** is sorted
- Mode: $M_0 =$ Most frequent $x \in \boldsymbol{x}$

Naive Central Tendency: Mean, Median, Mode

- If our dataset $\boldsymbol{x}$ has n entries…
- Mean: $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$
- Median: $m = x_{(n+1)/2}$ if $\boldsymbol{x}$ is sorted
- Mode: $M_0 =$ Most frequent $x \in \boldsymbol{x}$

- Example: $\boldsymbol{x} = [2, 1, 11, 5, 4, 3, 2]$

Method	Explanation	Result
Mean	Sum of all divided by n	4
Median	Middle value after sorting	3
Mode	Most frequent value	2

Questions about central tendency

- Are there cases where naive central tendency is misleading?
- Can there be more than one central tendency?
- What is the central tendency of a sentence?

Questions about central tendency

- Are there cases where naive central tendency is misleading?
  Most people earn less than the average income (skewed distribution)
- Can there be more than one central tendency?
  Yes, example on next slide
- What is the central tendency of a sentence?
  Ambiguous, hard to define

Fictional example: Central tendencies

Alien Running Contest

Alien Name	Alien Type	Age (Eons)	Time
0038001a	A	7.0	5:44
0038001b	B	7.1	7:02
0038001c	A	6.3	6:01
0038001d	A	7.3	5:56
0038001e	B	7.9	6:51
0038001f	B	6.0	7:12
0038001g	A	7.5	5:49
0038001h	A	8.5	5:46
0038001i	B	7.3	7:23
0038001j	A	7.8	6:14

Fictional example: Central tendencies

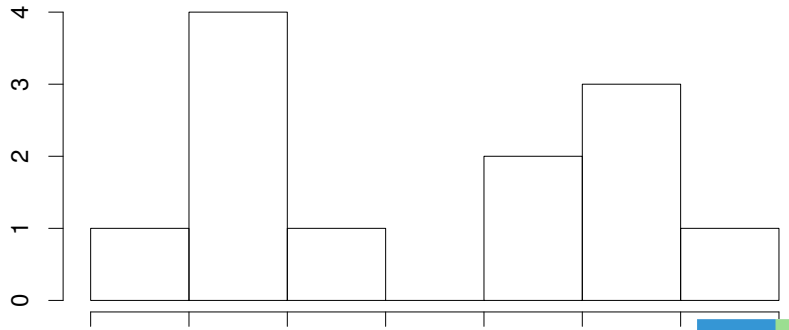
Alien Running Contest

Alien Name	Alien Type	Age (Eons)	Time
0038001a	A	7.0	5:44
0038001b	B	7.1	7:02
0038001c	A	6.3	6:01
0038001d	A	7.3	5:56
0038001e	B	7.9	6:51
0038001f	B	6.0	7:12
0038001g	A	7.5	5:49
0038001h	A	8.5	5:46
0038001i	B	7.3	7:23
0038001j	A	7.8	6:14

- $\mu =$ 6:29
- $m =$ NaN
- $M_0 =$ NaN

Fictional example: The Histogram…

Alien Running Contest



Fictional example: Conditional Mean

Alien Running Contest

Alien Name	Alien Type	Age (Eons)	Time
0038001a	A	7.0	5:44
0038001b	B	7.1	7:02
0038001c	A	6.3	6:01
0038001d	A	7.3	5:56
0038001e	B	7.9	6:51
0038001f	B	6.0	7:12
0038001g	A	7.5	5:49
0038001h	A	8.5	5:46
0038001i	B	7.3	7:23
0038001j	A	7.8	6:14

- $\mu_A = 5{:}55$
- $\mu_B = 7{:}02$
- Easy!…?

Fictional example: Without labels

Alien Running Contest #2

Time	Time Cont'd
21:12	25:16
20:05	21:23
28:49	26:14
26:18	20:39
22:23	20:04
24:01	21:02
22:02	22:21
25:14	25:56
26:02	26:51
21:45	21:42
28:55	26:05
22:24	22:21
28:49	26:14
26:18	21:39
22:23	20:04
24:01	21:02
22:02	22:21
25:14	25:56
26:02	26:51
21:45	21:42
28:55	26:05
22:24	22:21
27:29	20:04

Real example

Old Faithful

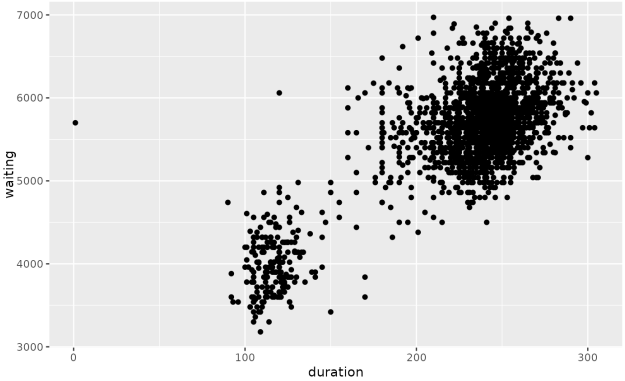
- Geyser eruption data
- Dataset has two features
  - Duration
  - Waiting time

Real example: Analysis

Old Faithful

Dataset has two main groups

1. Long waiting time long duration
2. Short waiting time short duration

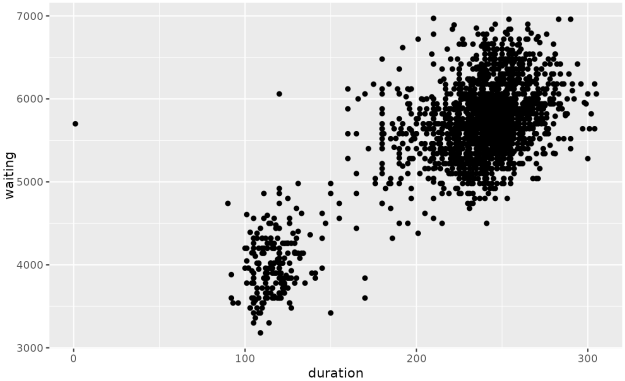


Real example: Analysis

Old Faithful

Dataset has two main groups

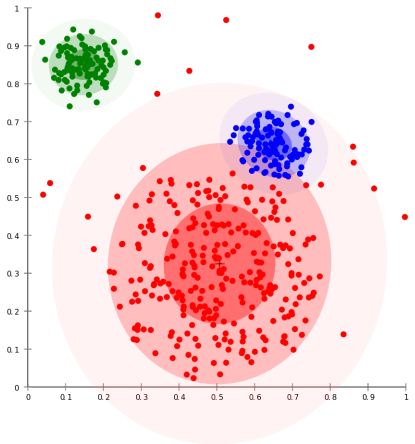
1. Long waiting time long duration
2. Short waiting time short duration



Question: How to find these groups with machine learning?

The Answer: Clustering

- Cluster ≈ group of similar objects
- Clustering: Find clusters in data
- Main questions
  - How to find clusters?
  - How many clusters?
  - What does “similar” mean?
  - Which “shapes” are permissible?



By Chire - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=17085713

Naive Approach to Clustering

- Direct approach to clustering:
  1. Partition data into groups
  2. Check if grouping is suitable
  3. If
     - Suitable→ Done
     - Not suitable → Go to 1.

- Some simple math:
  - Dataset: 100 objects
  - Clusters: 5 groups
  - Question: How many ways to arrange objects into groups?

Stirling Numbers of the Second Kind

- Solved by James Stirling in 18th century
- For n objects and k partitions: *Stirling numbers of the second kind*
- $\left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n$.
- Simple approximation: $\left\{ {n \atop k} \right\} \sim k^n$

Stirling Numbers of the Second Kind

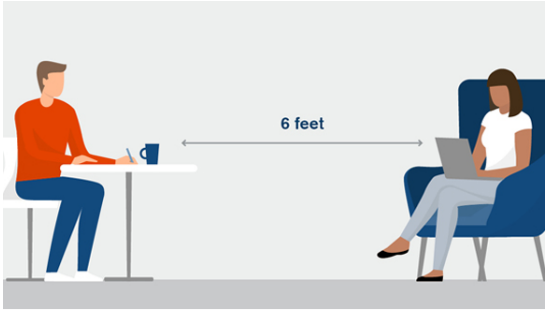
- Solved by James Stirling in 18th century
- For *n* objects and *k* partitions: *Stirling numbers of the second kind*
- $\left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n.$
- Simple approximation: $\left\{ {n \atop k} \right\} \sim k^n$
- $\left\{ {100 \atop 5} \right\} = 6573840870146189860689573375271143290269949536478824164584065977750 0$

Clustering cannot be “solved”

- Optimal clustering not possible (in general)
- We cannot find the “best” clustering of a dataset
- Need some tricks
  - Similarity as starting point
  - Not *best* solution, but *feasible* solution
  - Focus on simple cases
  - $\rightarrow$ *heuristics*

Distance functions

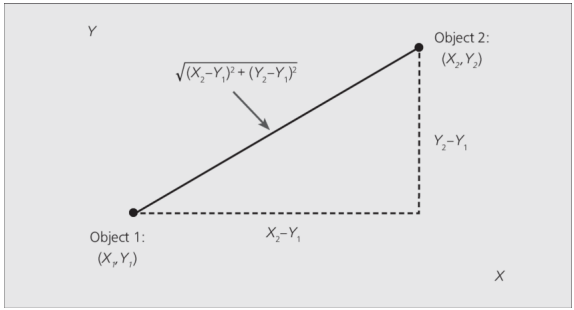
- similarity $\overset{\text{opposite}}{\leftrightarrow}$ distance
- distance often easier to measure
- distance function $d(x, y)$
- $d(x, y)$: how dissimilar are x and y
- important for many clustering algorithms



By Katie McCallum - https://www.houstonmethodist.org/-/media/Images/Contenthub/Article-Images/Coronavirus

Euclidean Distance

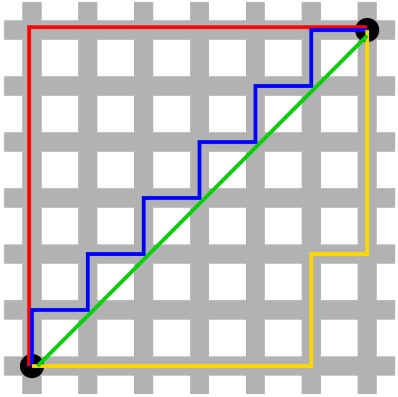
- Named after Euclid
- Natural, intuitive distance
- $d(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$
- “Airplane” distance
- Metric of Euclidean space
  - “Normal” space as humans think
  - Different dimensions as per dataset
  - Formula always the same



By Young-Sun Lee - Own work
https://www.researchgate.net/figure/An-example-of-Euclidean-distance-between-two-objects-on-variables-X-and-Y_fig1_263889770

Manhattan distance

- Taxicab geometry
- $d(x, y) = \sum_{i=1}^{d} |x_i - y_i|$
- “Taxi” distance
- Red, blue and orange line cover same Manhattan distance
- Green is Euclidean distance



Created by User:Psychonaut with XFig, Public Domain, https://commons.wikimedia.org/w/index.php?curid=731390

Other Relevant Distances

- Chebyshev distance
  - Chessboard distance $d(x, y) = \max_i |x_i - y_i|$
  - Widely used in logistics, e.g., crane movement, and in grid-based optimization
- Minkowski distance
  - General distance $d(x, y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{\frac{1}{p}}$
  - Flexible and very popular in ML. Need to select p.
- Jaccard distance
  - Set distance $d(x, y) = \frac{|x \cap y|}{|x \cup y|}$
  - Used for sets, categorical data, and clustering **documents** (bag-of-words, n-grams)
- Cosine distance
  - Angle distance $d(x, y) = 1 - \frac{x \cdot y}{||x|| \, ||y||}$
  - Cluster high dimensional data like **TF-IDF, word embeddings** etc.

Clustering Algorithms

- Algorithm $\approx$ instruction manual
- Clustering algorithm $\approx$ instruction manual for clustering data
- There are many different clustering algorithms
- Available in many programming libraries
- Clustering algorithms **do not** solve clustering!
- $\rightarrow$ Only solve specific clustering problems

k-Means: Problem

- Data have more than one “mean”
- Ingredients: Dataset, distance function and number of means
- Question: Where are those means? Can we use *distance* to find them?
- We know that **solving** this problem is very hard (NP-hard)
- We need some *heuristic* algorithm

k-Means: Lloyd's Algorithm

1. Start with some (random) *k* means

2. Assignment step:
   - 2.1 Compute distance between each point and each mean
   - 2.2 Assign each point to its nearest mean

3. Update step:
   - 3.1 Calculate new means as arithmetic average over all assigned points
   - 3.2 Check if new means are different from old means
     - If YES, GoTo assignment step
     - If NO, stop algorithm

k-Means: Demo

Live-Demo

k-Means: Pros & Cons

+ Super fast
+ Easy to understand and interpret
+ Geometrically partitions data
– Greedy algorithm - depends strongly on initial means
– Need to know *k*
– Cannot deal with noisy data or outliers

HAC: Problem

- **H**ierarchical **A**gglomerative **C**lustering
- Data should be represented by a hierarchy
- Ingredients: Dataset and distance function
- Question: How to construct the hierarchy?
- Problem is also hard, so heuristic needed

HAC: Single/Complete Linkage

1. Compute distances between all components
   - Single: Closest points between components
   - Complete: Farthest points between components
2. Merge most similar components
3. Update all distances for the merged components
4. Check number of components
   - If more than one component: GoTo 2
   - Else stop

HAC: Demo

Live-Demo

HAC: Pros & Cons

+ Don’t need to know number of clusters
+ Compute all HAC simultaneously
+ Dendrogram great for summarizing small datasets
– Very slow
– High memory requirements
– Dendrogram less helpful for large datasets

DBSCAN: Problem

- **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
- Dataset might be noisy, have outliers and have complex cluster structures
- Ingredients: dataset, distance function, distance threshold, minimum density
- Question: Can we cluster complex datasets where normal clustering algorithms don't work (non-linearly separable)?
- Problem seem hard, but there is a good solution based on density (number of points in specified region)

DBSCAN: Algorithm

1. Explore the neighborhood of each point
   - If many neighbors: core point
   - If few neighbors: non-core point
2. Find connected core points to form clusters
3. For each non-core point
   - If closer than distance threshold, add to cluster
   - Else add to noise cluster

DBSCAN: Demo

Live-Demo

DBSCAN: Pros & Cons

+ Doesn't need cluster number
+ Find clusters with complex shapes
+ Noise and outliers are no problem
– Struggles with high-dimensional data
– Equal density assumption does not always hold
– Sometimes, selecting distance threshold and minimum density can be difficult

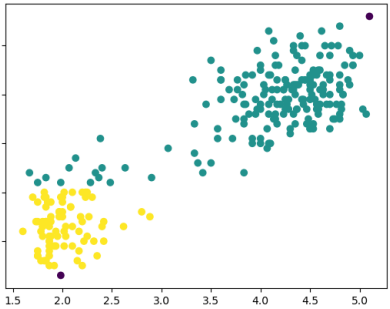
Beyond DBSCAN: HDBSCAN

- Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)
- Builds **hierarchical clustering** by varying distance threshold.
- Extracts the most stable clusters from the hierarchy.
- $\rightarrow$ Automated DBSCAN
  - Don’t need to specify distance threshold and minimum density
  - Need to minimum cluster size

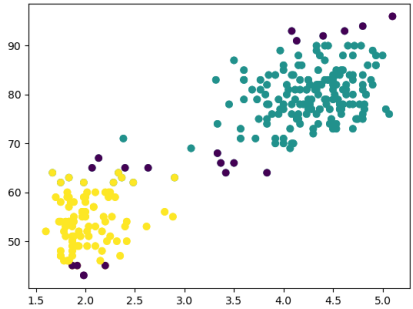
DBSCAN vs HDBSCAN

On Old Faithful (cleaned)

DBSCAN



HDBSCAN



HDBSCAN: Pros & Cons

+ Often works well with BERTopic (!)
+ Find clusters with different densities
+ Only needs minimum cluster sizes
– Automation implies heavy inductive bias
– Still struggles with high-dimensional data→ often needs dimensionality reduction
– Often, we don’t know minimum cluster size

Experimental Approaches for Clustering

Deep clustering

- Recently, many deep learning techniques explored for clustering
- Examples
  - Deep Embedded Clustering
  - DeepCluster (alternation approach)
  - Contrastive Clustering (e.g., SCAN, SimCLR+K-Means)
- Many novel techniques, but not established
- $\rightarrow$ Recommended to stick with established methods (especially for rich data like text documents)

Summary

Summary

- If your data have more than one central tendency $\rightarrow$ clustering

Summary

- If your data have more than one central tendency → clustering
- Optimal clustering is not possible

Summary

- If your data have more than one central tendency $\rightarrow$ clustering
- Optimal clustering is not possible
- Clustering algorithms are heuristics and often use distance functions.

Summary

- If your data have more than one central tendency $\rightarrow$ clustering
- Optimal clustering is not possible
- Clustering algorithms are heuristics and often use distance functions.
- Popular algorithms
  - *k*-means—Find *k* central tendencies according to distances
  - HAC—Build cluster hierarchy according to distances
  - DBSCAN—Find connected dense regions

Summary

- If your data have more than one central tendency $\rightarrow$ clustering
- Optimal clustering is not possible
- Clustering algorithms are heuristics and often use distance functions.
- Popular algorithms
  - *k*-means—Find *k* central tendencies according to distances
  - HAC—Build cluster hierarchy according to distances
  - DBSCAN—Find connected dense regions
- Hierarchical DBSCAN recommended approach for BERTopic

Thank you!

FROM DATA TO VALUE

Max Toller
mtoller@know-center.at

KNOW-CENTER GMBH
Research Center for
Trustworthy AI and Data
Sandgasse 34, 2nd floor, 8010 Graz, Austria
Commercial register court: LGZ Graz
FN 199 685 f
UID: ATU 50367703
www.know-center.at