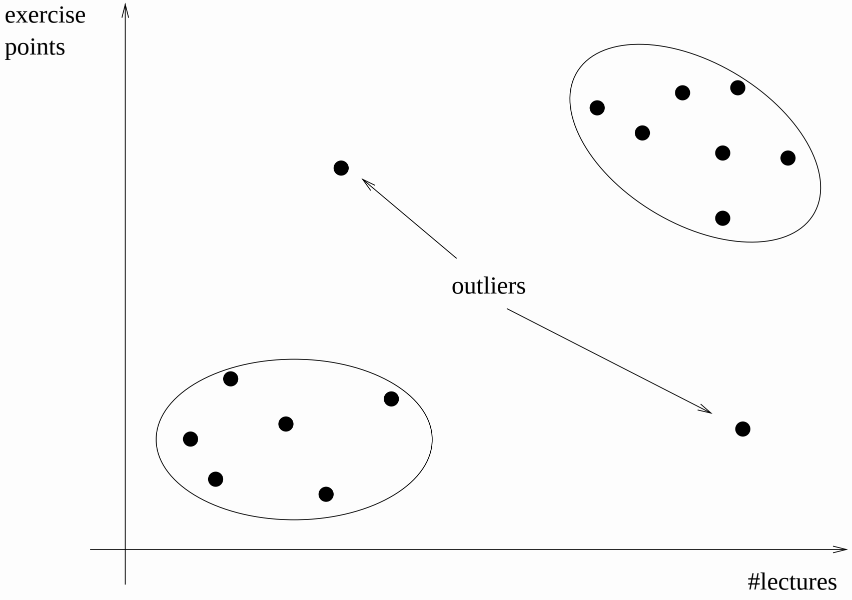
Clustering

Intuitively: Partition data $\mathcal{D}$ into K clusters $C_1, \ldots, C_K$ such that points in each cluster are similar to one another but dissimilar to points in other clusters.



- **hard clustering**: each point belongs to one cluster
- **soft clustering**: a point can belong to multiple clusters with different probabilities or weights

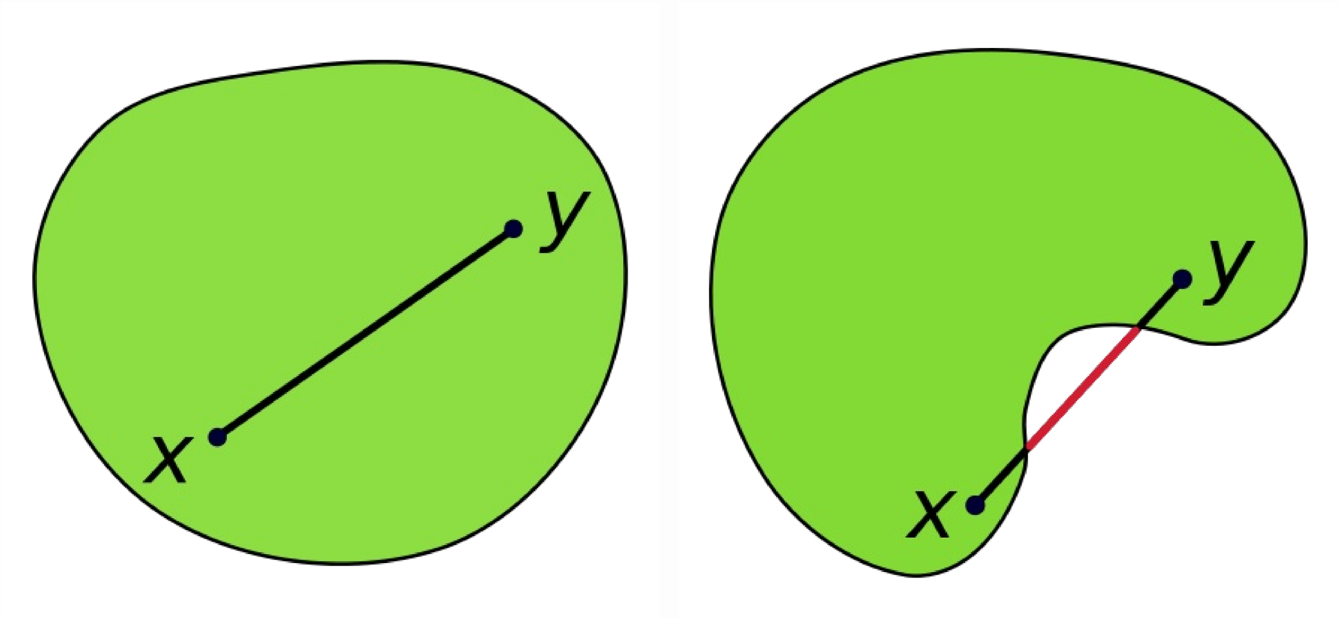
What is the objective of clustering?

What kind of clusters should be found?

- shape: is the shape of clusters fixed (e.g., hyperspherical) or arbitrary?
- size: balanced clusters or clusters of different sizes?
- density: equal or variable?
- overlapping or well-separated clusters?
- outliers?

$\Rightarrow$ different methods, objective functions and distance measures

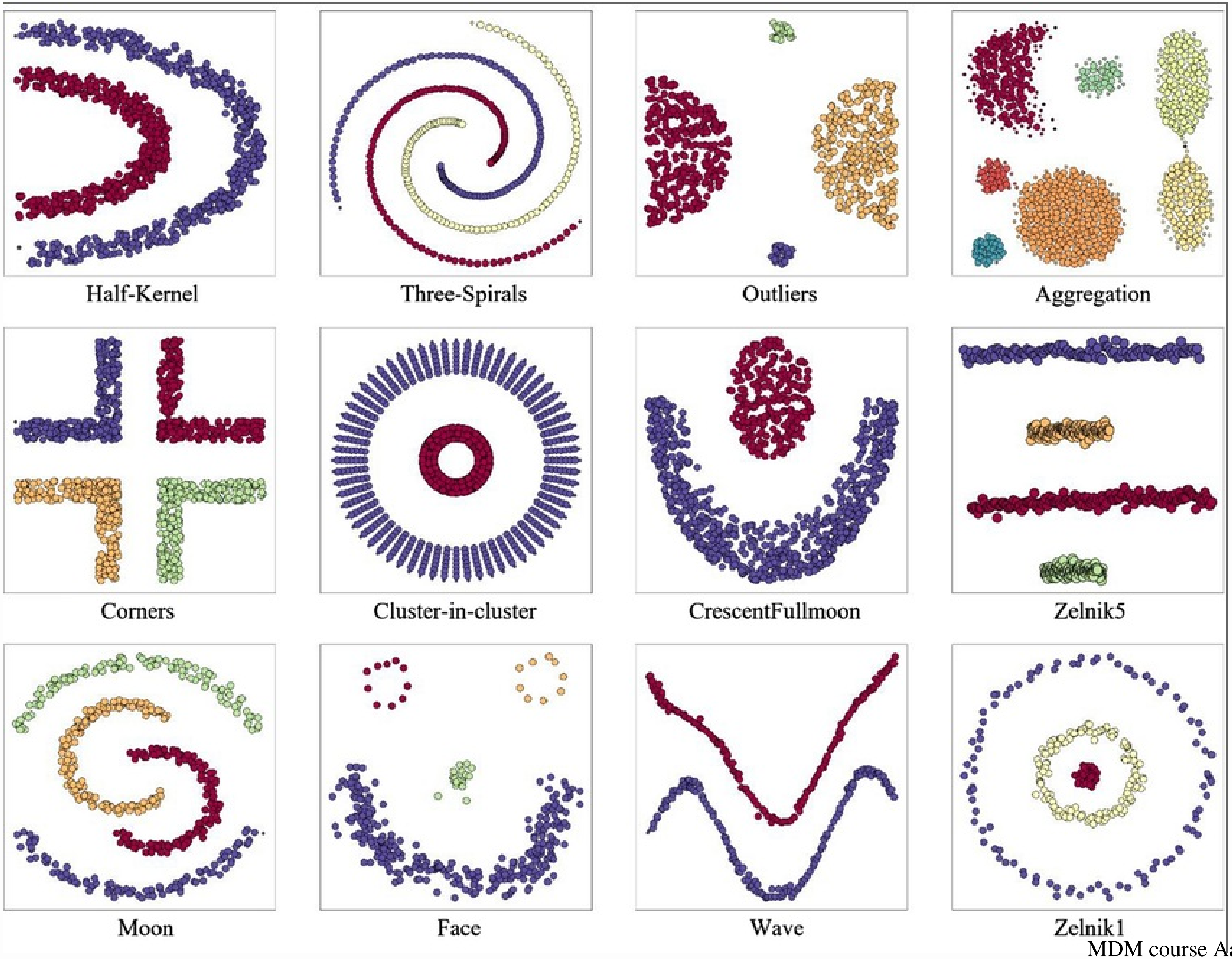
Shapes: convex or non-convex?



E.g., hyperspheres, hyperrectangles, and Voronoi cells are convex.

Image source: wikipedia,
https://en.wikipedia.org/wiki/Convex_set

Examples of tricky cluster structures (Senol 2023)



What is needed?

- distance measure d (or similarity measure)
- distance measure D for inter-cluster distances
  - sometimes needed
- vector space representation of $\mathcal{D}$?
  - sometimes needed
  - sometimes a similarity or distance graph suffices
- objective (score) function to evaluate clustering
  - algorithm tries to optimize this
  - not always explicit
- number of clusters K (often needed)

Examples of objective functions

Usually combine two objectives: **minimize within-cluster-variation** wc and **maximize between-cluster variation** bc

Let $\mathbf{C} = \{C_1, \ldots, C_K\}$ clusters, $\mathbf{c}_1, \ldots, \mathbf{c}_K$ their centroids and d distance function. Examples of wc:

$$wc(\mathbf{C}) = \sum_{i=1}^{K} wc(C_i) = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} d^2(\mathbf{x}, \mathbf{c}_i) \quad \rightarrow \text{hyperspherical clusters}$$

$$wc(C_p) = \max_i \overbrace{\min_{\mathbf{y} \in C_p} \{d(\mathbf{x}_i, \mathbf{y}) \mid \mathbf{x}_i \in C_p, \mathbf{x}_i \neq \mathbf{y}\}}^{\mathbf{x}_i\text{'s distance to its nearest neighbour in } C_p} \quad \rightarrow \text{elongated clusters}$$

Examples of objective functions

Let $\mathbf{C} = \{C_1, \ldots, C_K\}$ clusters, $\mathbf{c}_1, \ldots, \mathbf{c}_K$ their centroids and d distance function. Example of bc:

$$bc(\mathbf{C}) = \sum_{1 \leq i < j \leq K} d^2(\mathbf{c}_i, \mathbf{c}_j)$$

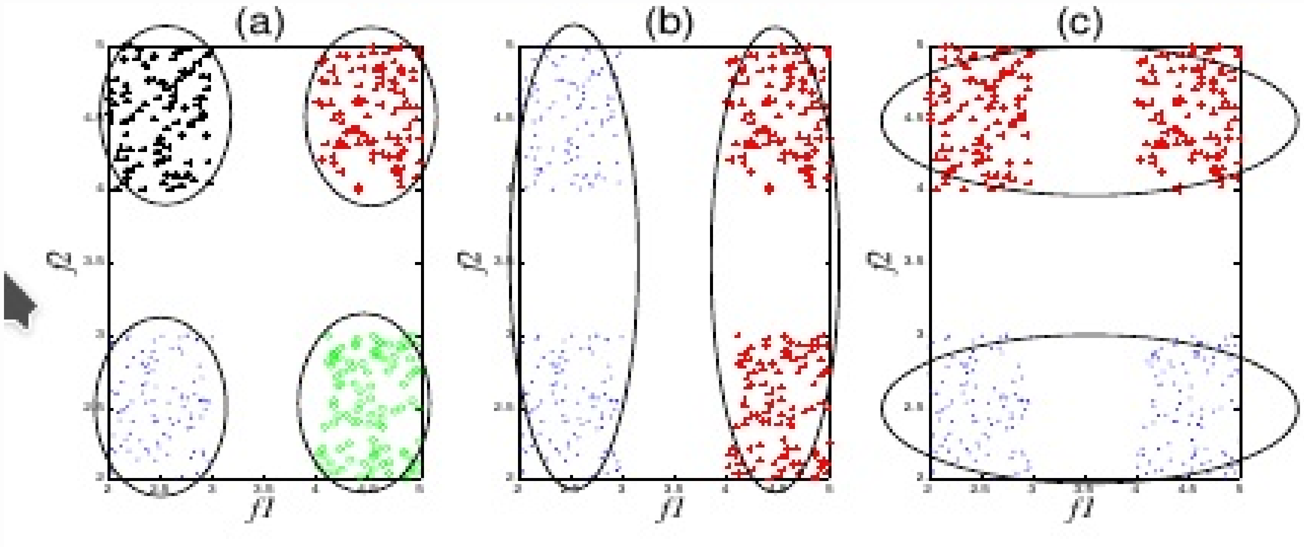
An example of an overall measure is K-**means criterion**:

$$SSE(\mathbf{C}) = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} L_2^2(\mathbf{x}, \mathbf{c}_i)$$

(minimizing SSE minimizes within-cluster variance and maximizes between-cluster variance)

Clusters depend on the features!

Example: Features $f1$ and $f2$ distinguish 4 clusters, while $f1$ alone or $f2$ alone distinguish 2 clusters:



$\Rightarrow$ Is there **clustering tendency** when the data is presented with the given features?

Source: Aleyani et al. (2018): Feature Selection for Clustering: A Review

Preprocessing has a crucial role in clustering!

- feature extraction
- feature selection and dimension reduction
- to scale or not to scale?
  - if features have very different scales, some scaling usually needed
  - **but** sometimes normalization distorts separation

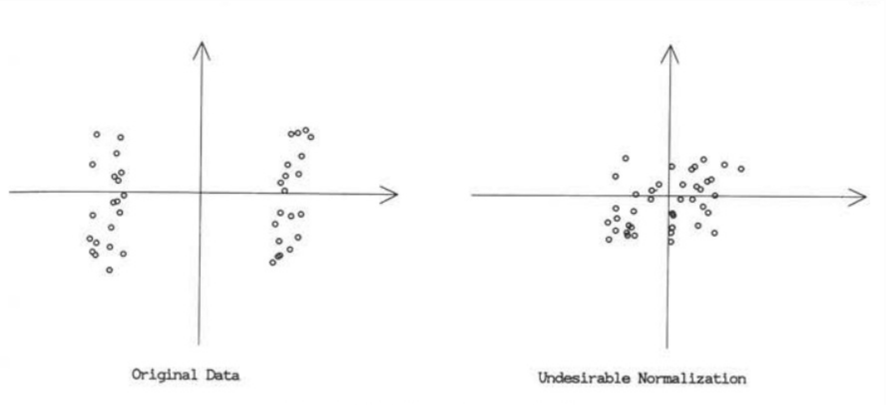


image source: Jain and Dubes: Algorithms for clustering data. 1988

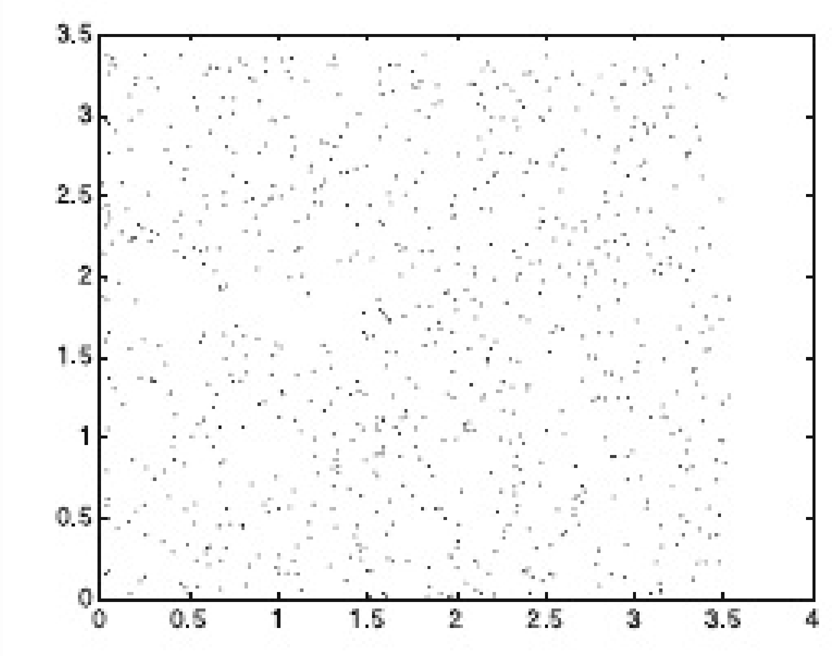
How to study clustering tendency and choose features?

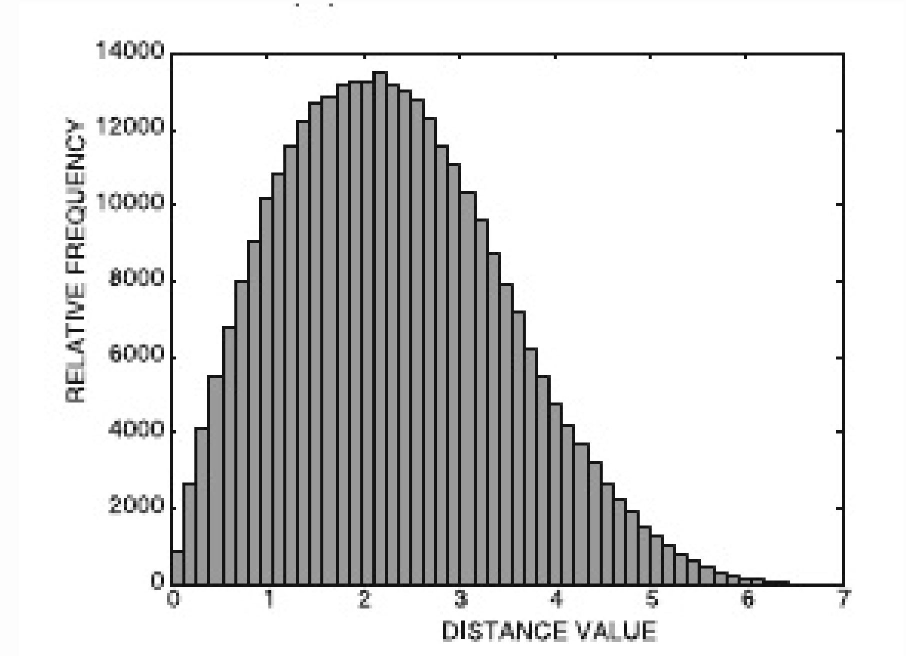
Approaches:

1. Visual inspection of pairwise distance distributions
   - only hints
2. Filtering methods, e.g.,
   - Entropy-based measures
   - Hopkins statistic
3. Wrapper models + cluster validation indices
   - e.g., average silhouette, Calinski-Harabasz, Davies-Bouldin, and external indices → next lecture

1. Visual inspection: Distance distributions

Plot a histogram of pairwise distances in data. What the distribution looks if there are no clusters?

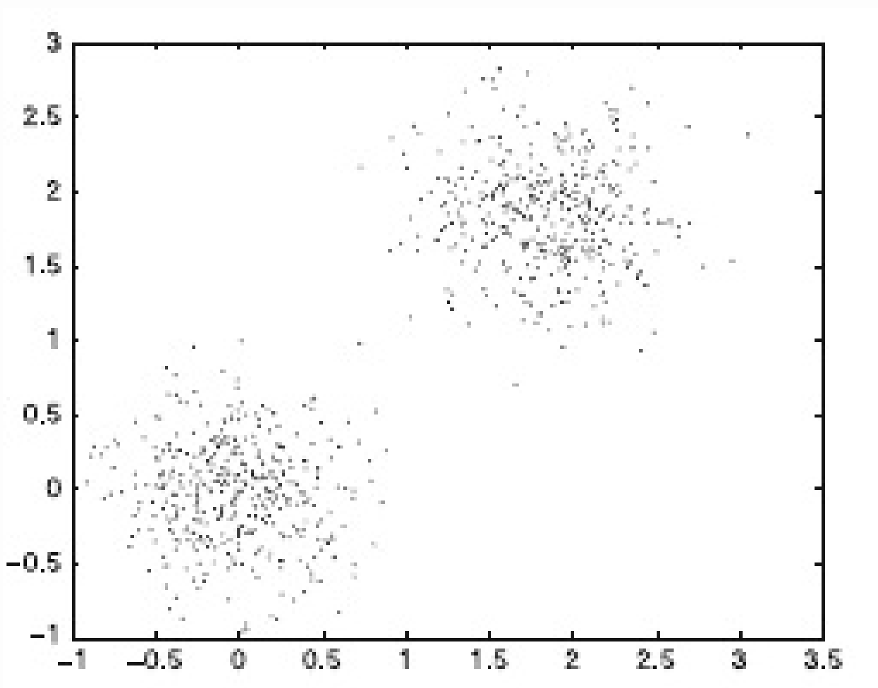


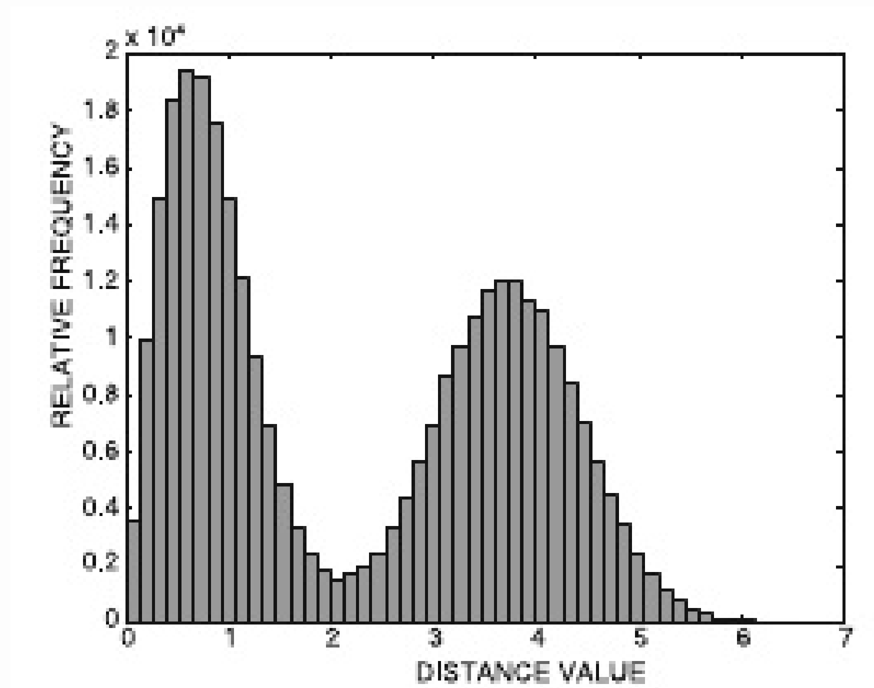


Source: Aggarwal Ch 6

Distance distributions (cont'd)

Distribution has more peaks if there are clear clusters!
Why?

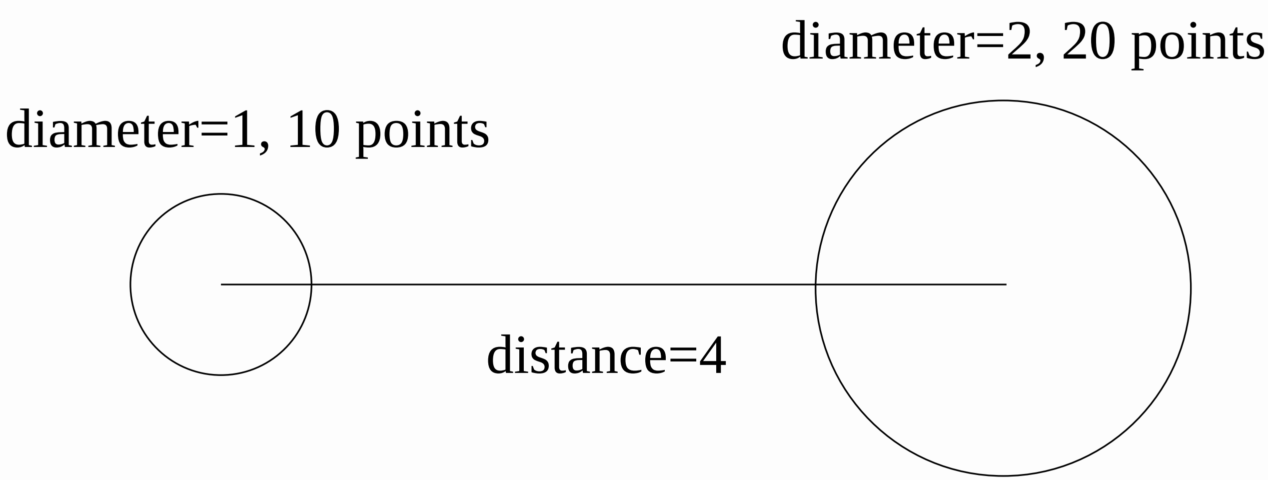




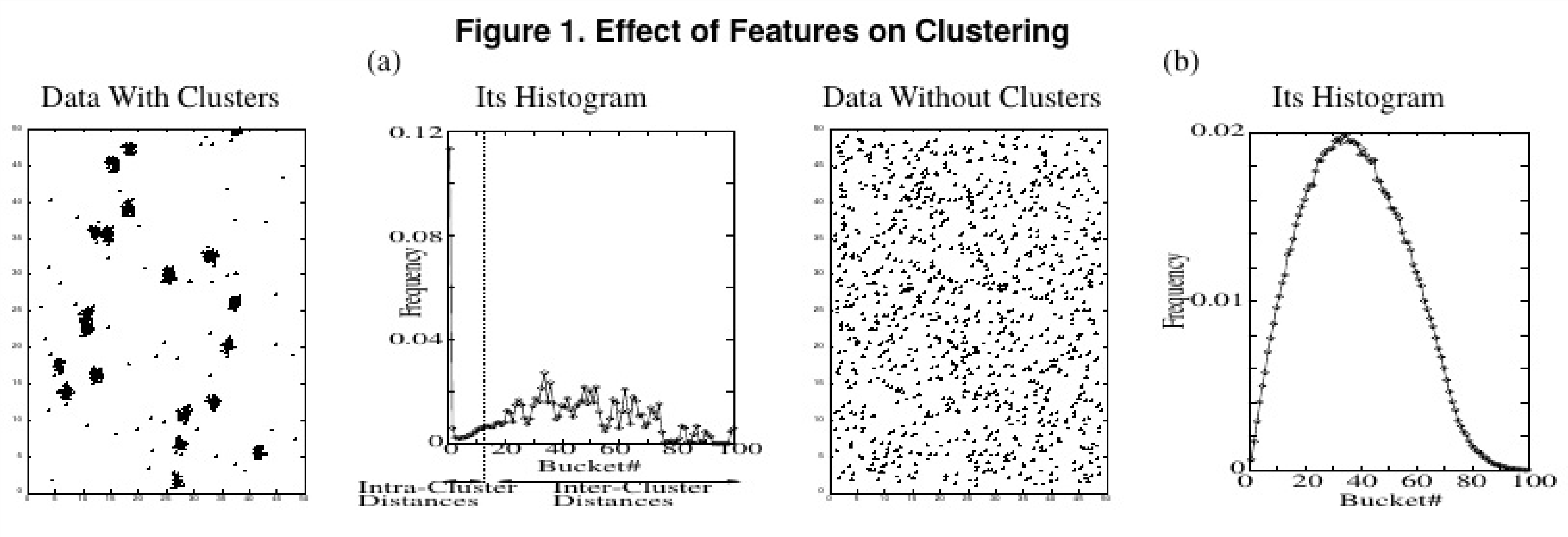
Source: Aggarwal Ch 6

Distance distributions: Task

Assume that points are distributed evenly inside each cluster (nothing outside).
What are the ranges of intra-cluster and inter-cluster distances? What does the pairwise distance distribution look like?



Distance distributions: Another example



Source: Dash et al.: Feature selection for clustering – a filter solution. ICDM, 2002.

2.1 Entropy-based measures

Idea: In random data (uniform distribution), the entropy is high, and in clustered data low.

Approach 1:

- Discretize data into m multidimensional grid regions
  p_i=fraction of data points in region i
- evaluate probability-based entropy

$$E = -\sum_{i=1}^{m}[p_i \log(p_i) + (1 - p_i)\log(1 - p_i)]$$

Note: Independent, binary region variabes (region is occupied with probability p_i or empty with $1 - p_i$).

Entropy-based measures (cont'd)

Problems:

- p_i can be hard to estimate accurately in large dimensionality
- how to choose m?
- m should be approximately the same for different feature subsets
  - e.g., for each of k dimensions, $\lceil m^{1/k} \rceil$ bins

Entropy-based measures (cont'd)

Approach 2:

- calculate pairwise distances between points
- discretize distances onto m bins
- p_i=fraction of distances in the ith bin
- calculate E

$$E = -\sum_{i=1}^{m} [p_i \log(p_i) + (1 - p_i) \log(1 - p_i)]$$

Entropy-based measures: choosing features

Often iterative, greedy search, either:

1. Forward selection: at each round add the best feature
   - largest decrease in entropy; **or**
2. Backward selection: at each round drop the worst feature
   - largest increase in entropy

More on entropy-based methods: Aggarwal 6.2.1.3 and Dash et al.: Feature Selection for Clustering – A Filter Solution. ICDM, 2002.

2.2 Hopkins statistic

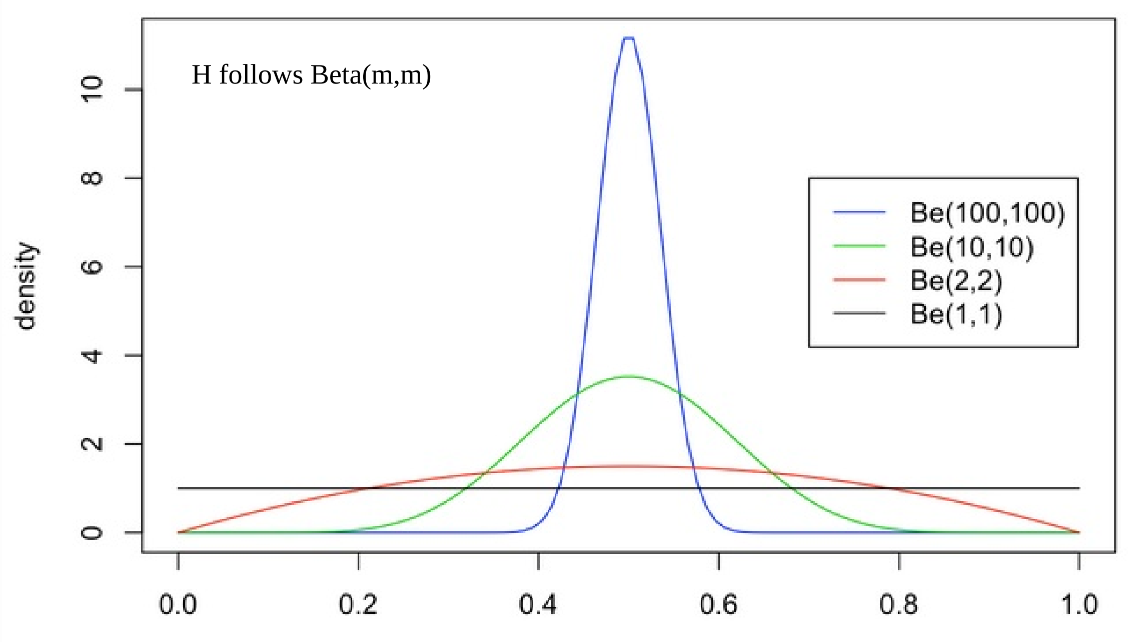
Idea: Compare nearest neighbour distances from the original data and random data points.

- Take a sample R of size r from original data $\mathcal{D}$
- Generate random data (from uniform distribution) and take a sample S of size r from it
- Calculate for all $\mathbf{x} \in R$ distances to their nearest neighbours (in $\mathcal{D}$). Let these be $\alpha_1, \ldots, \alpha_r$
- Calculate for all $\mathbf{x} \in S$ distances to their nearest neighbours (in $\mathcal{D}$). Let these be $\beta_1, \ldots, \beta_r$

Hopkins statistic H

$$H = \frac{\sum_{i=1}^{r} \beta_i}{\sum_{i=1}^{r} (\alpha_i + \beta_i)}$$

- if $\mathcal{D}$ has uniform distribution, $H \approx 0.5$
- if there are clusters, H approaches 1



source: betadistr.eps https://stephens999.github.io/fiveMinuteStats/beta.html

m= sample size (our r)

Hopkins statistic: Problems

1. distance distribution often very different in the center of data than on edges
   $\Rightarrow$ choose sample points inside a hypersphere centered at the mean of data and containing 50% of data points
2. results vary with different executions
   $\Rightarrow$ repeat multiple times and calculate average

3. Wrapper models and validation indices

Idea: Iteratively cluster data with different feature sets and use validity indexes to find good features.

First approach:

- Cluster data and calculate some internal cluster validity index
  - can't try all feature subsets $\rightarrow$ use e.g., greedy heuristic
  - results depend on the validity criterion (and clustering method)

Wrapper models and validation indices (cont'd)

Second approach:

- Create artificial class labels and identify discriminative features in a supervised manner
  - cluster data and use cluster identifies as class labels
  - evaluate each feature separately utilizing class labels (goodness measures for classification)
  - circular definition: features are good if the clustering is good, but good clustering requires good features

Summary

- Try to understand your clustering objective
- How to evaluate clustering tendency (given features)?

Further reading:

- Gan, Ma, Wu: Data clustering – theory, algorithms, and applications. SIAM 2007.
- Jain and Dubes: Algorithms for clustering data. Prentice-Hall 1988. (math properties, clustering tendency)

References

- Senol (2023): MCMSTClustering: defining non-spherical clusters by using minimum spanning tree over KD-tree-based micro-clusters. Neural Computing and Applications 35(1):1-21.