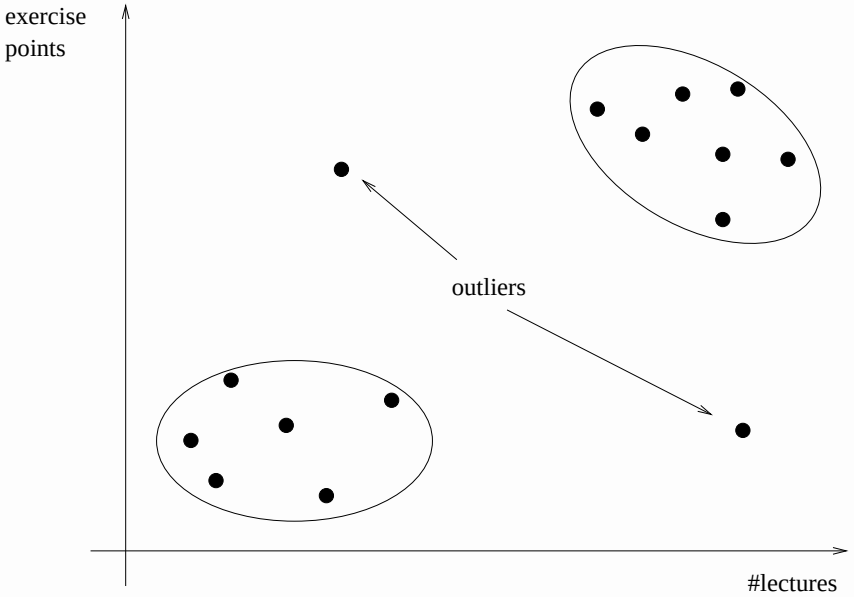
Clustering

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

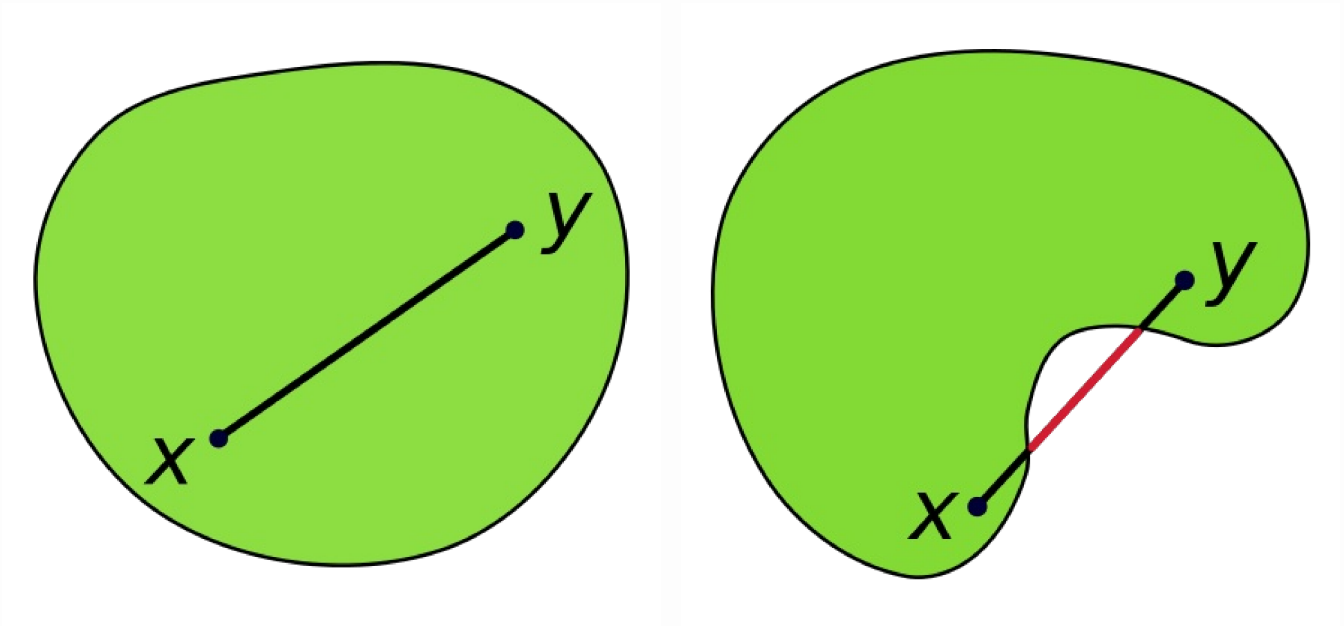
What is the objective of clustering?

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

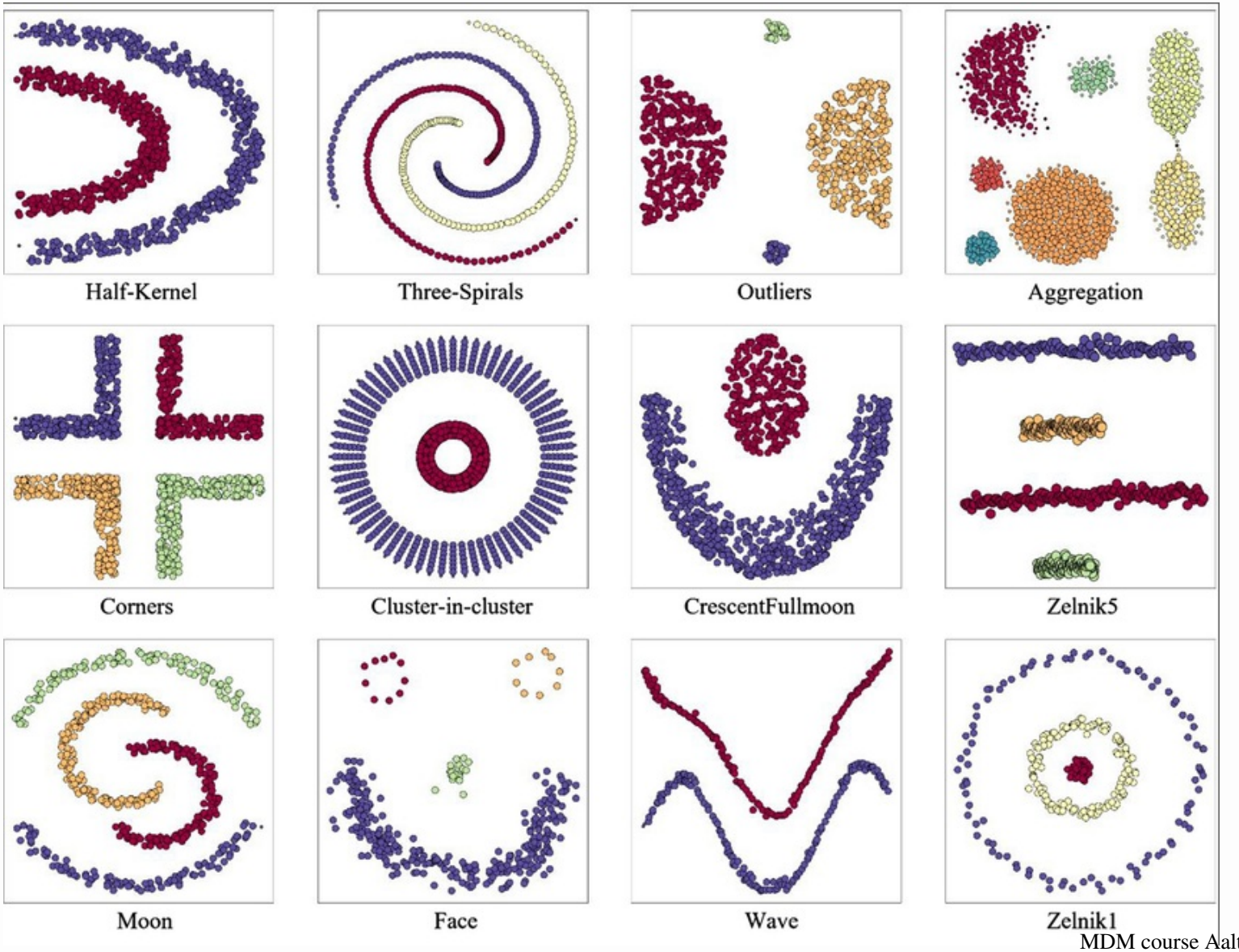
Shapes: convex or non-convex?



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:

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

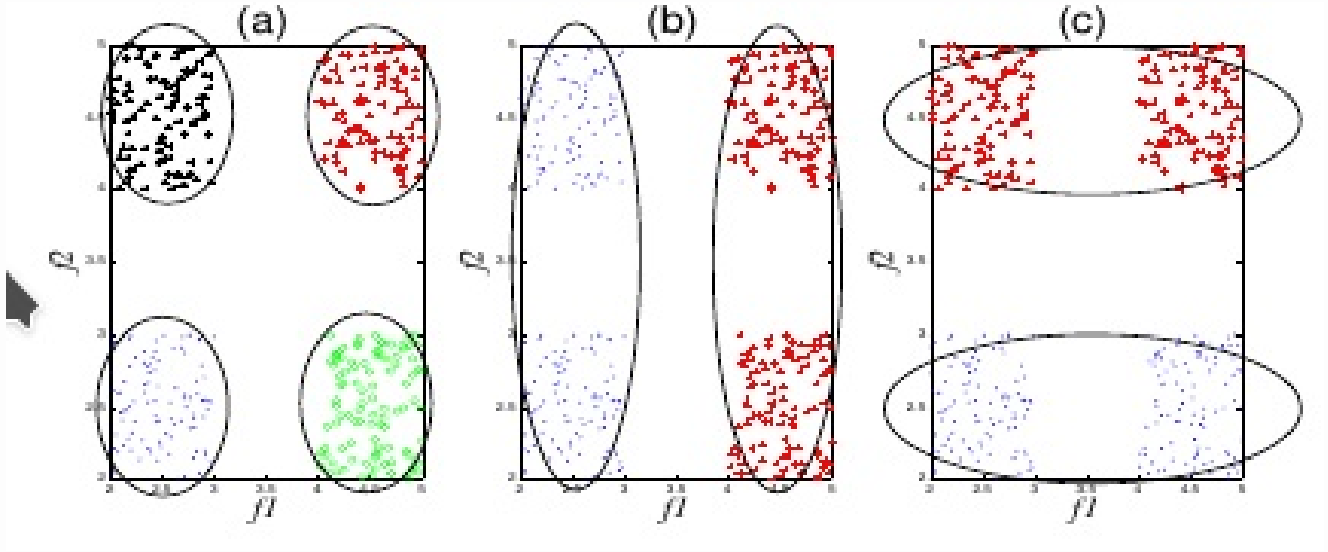
An example of an overall measure is **K-means criterion**:

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

(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

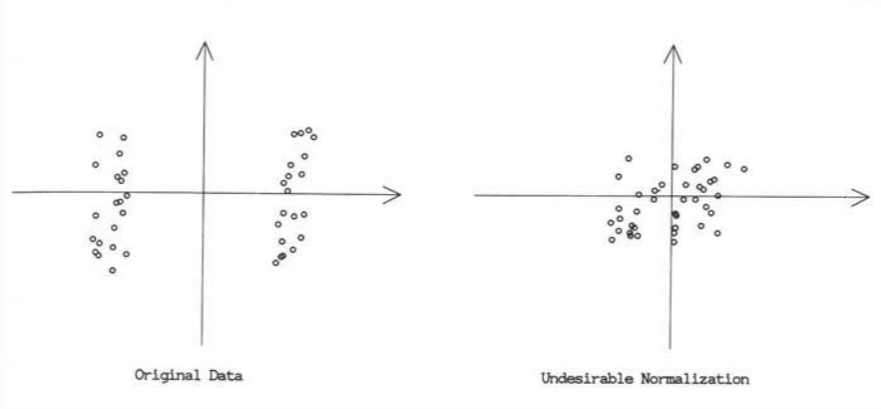


image source: Jain and Dubes: Algorithms for clustering data. 1988

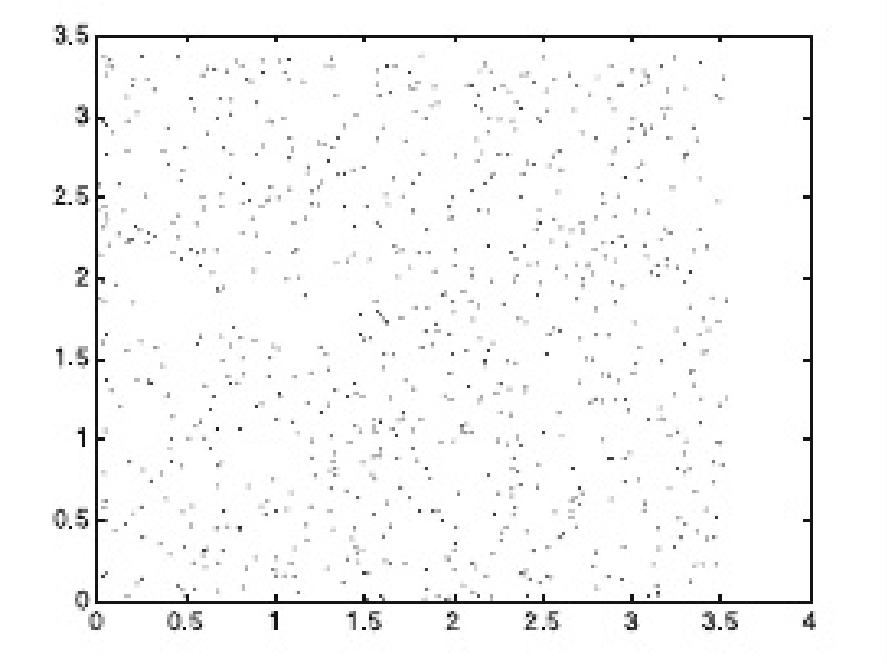
How to study clustering tendency and choose features?

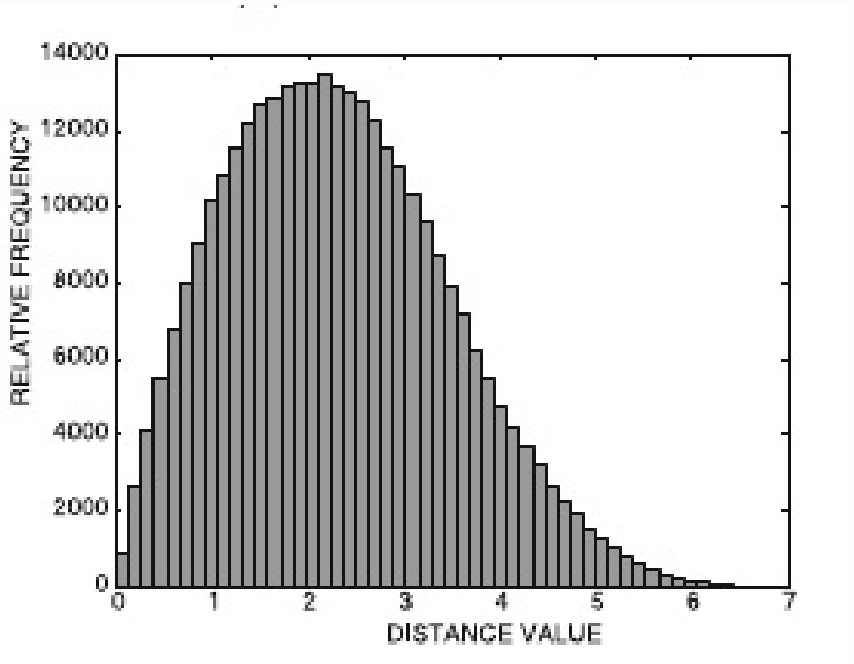
Approaches:

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?

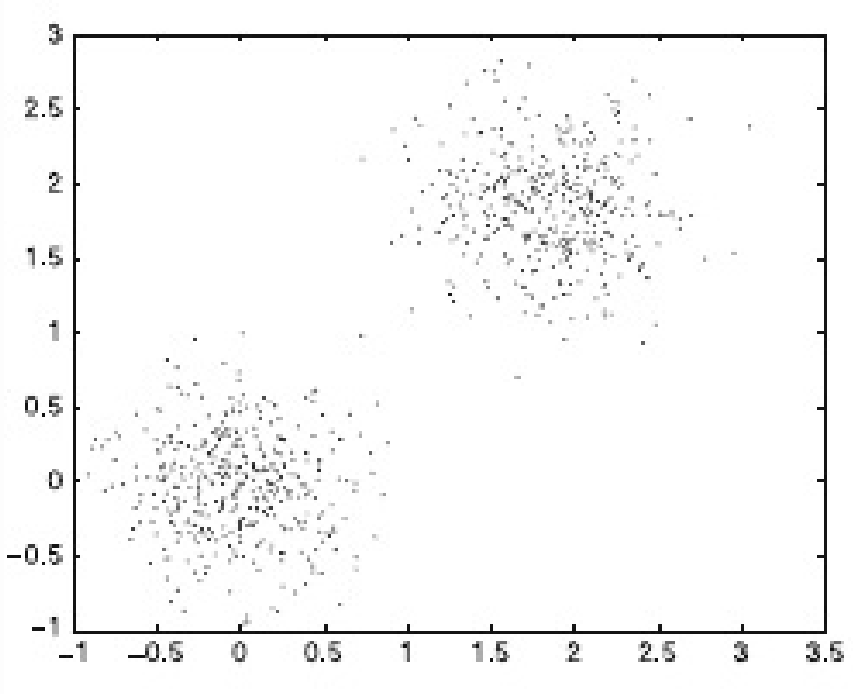


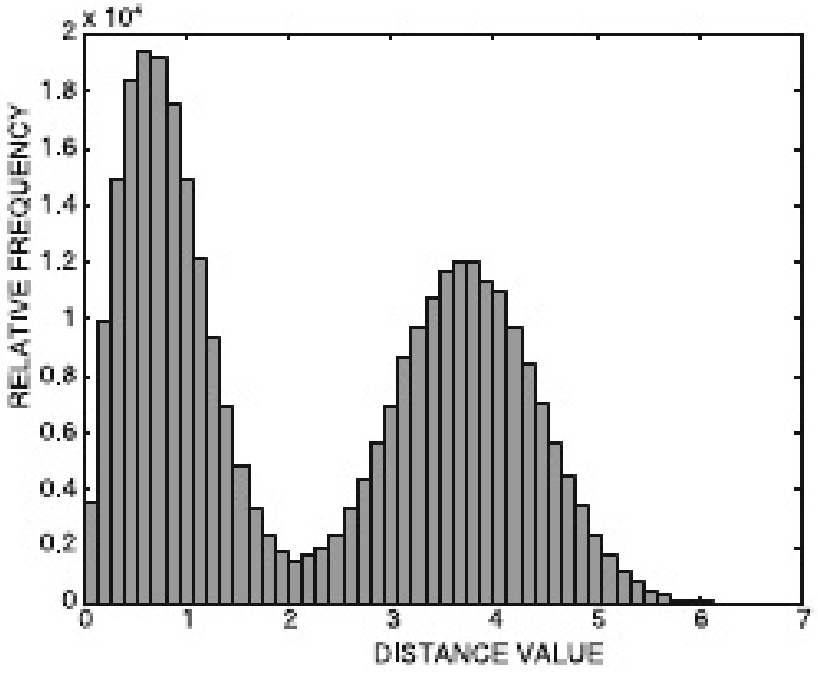


Source: Aggarwal Ch 6

Distance distributions (cont'd)

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

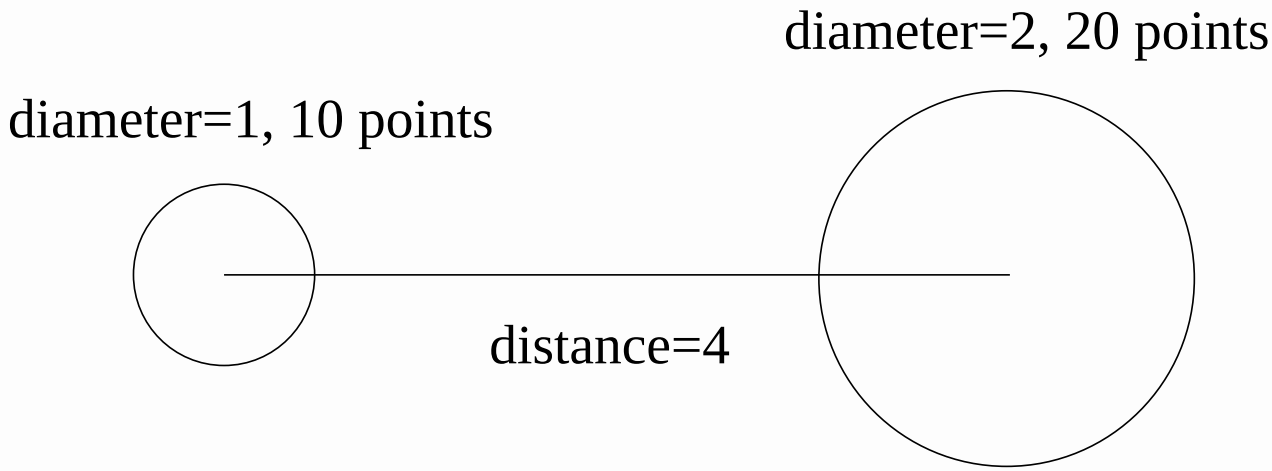




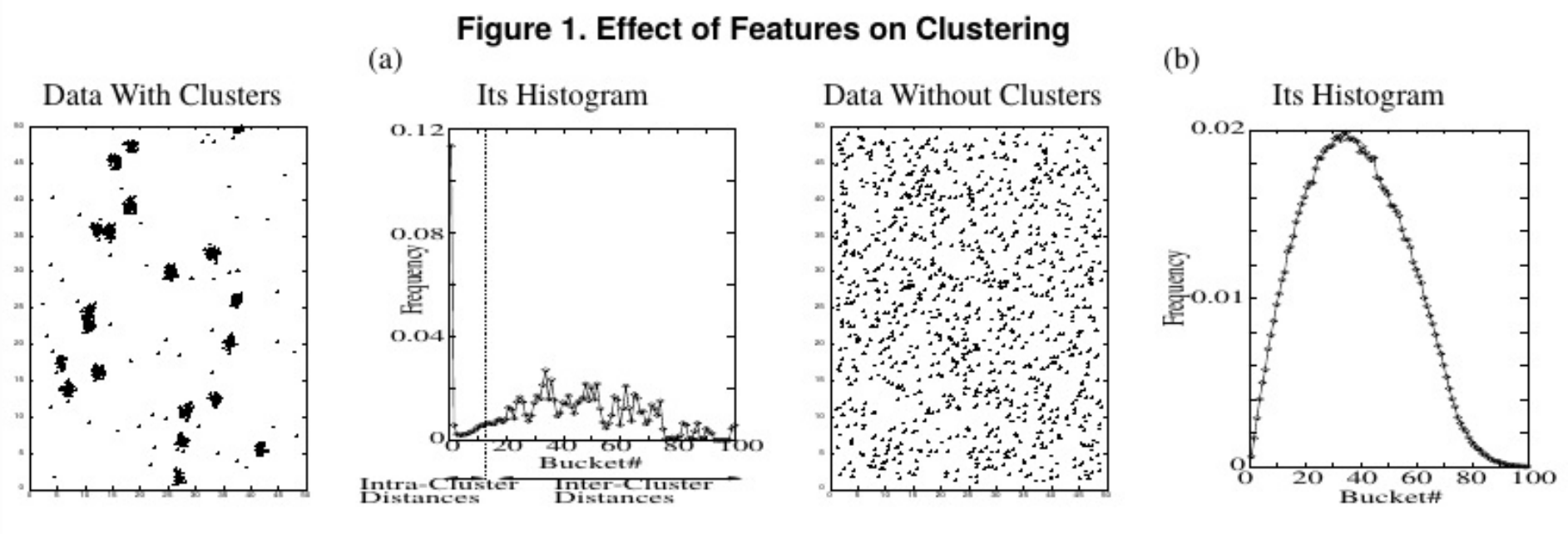
Source: Aggarwal Ch 6

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

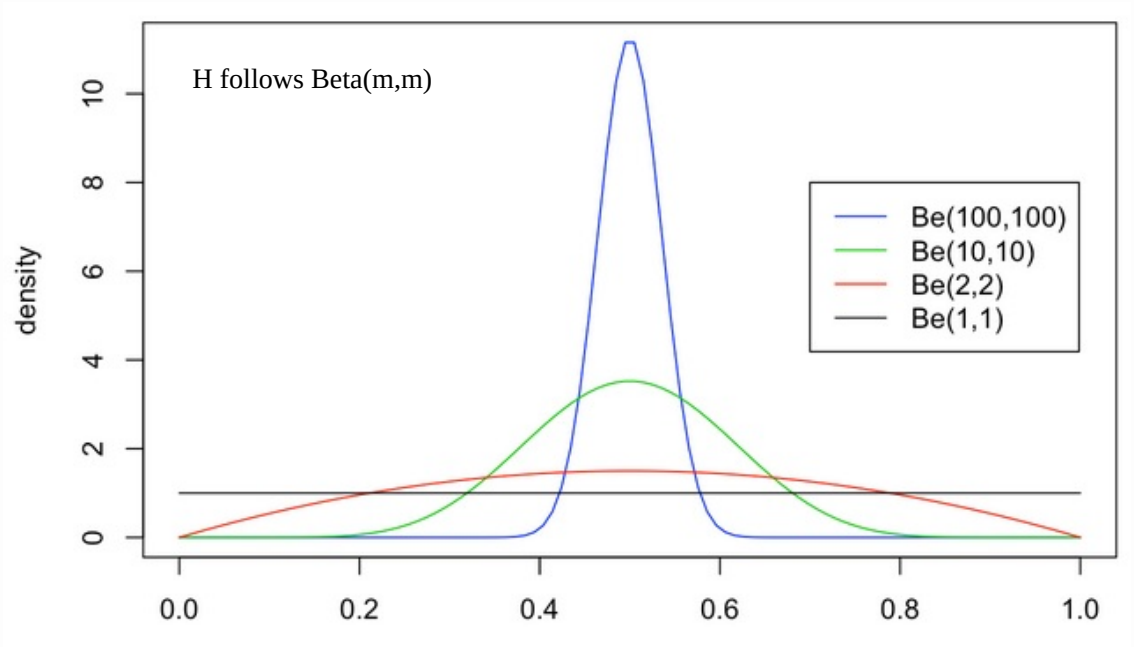
Idea: Compare nearest neighbour distances from the original data and random data points.

- 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 → 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.

Lecture programme

Lecture 3:

- Dimension reduction with PCA and SVD
- Clustering I (clustering tendency)

Book: Sec. 2.4.3, 6.1–6.2

- L4: Clustering II (K-representatives, hierarchical)
- L5: Clustering III (spectral, validation)

Why all eigen and singular stuff?

Goal: nice low-dimensional representation for data, when

- original data high-dimensional
- only pairwise distances/similarities known

Recall: Given $n \times n$ matrix $\mathbf{A}$, λ is eigenvalue and $\mathbf{v} \neq \mathbf{0}$ corresponding eigenvector, if $\mathbf{Av} = \lambda\mathbf{v}$

Good news: Everything will be real-valued!

- $k \times k$ covariance matrix and $n \times n$ Laplacian matrices positive semidefinite [a] $\Rightarrow$ real λs and $\mathbf{v}$s
- $n \times k$ data matrix real $\Rightarrow$ singular vectors real (singular values always real)

[a] $\mathbf{z}^T\mathbf{Az} \geq 0$ for all real $\mathbf{z} \neq \mathbf{0}$

Dimension reduction: motivation

1. **Curse of dimensionality**: hard to distinguish close and far neighbours in high dimensional data! $\rightarrow$ how to find clusters??
2. **Redundancy in data**: If features strongly correlated, the same information can be presented with a smaller number of features
   - intrinsic dimensionality r may be $r << d$

   $\Rightarrow$ Idea: reduce dimensionality by removing this redundancy!

Principal component analysis (PCA) assumptions

1. High **variance** reflects important structures of data.
2. Data can be presented well as a **linear** combination of suitable **orthogonal** basis vectors (PCs).

Idea: Given $n \times d$ data and suitable $d \times r$ ($r < d$) matrix $\mathbf{P}_r$, new data will be $n \times r$ matrix $\mathbf{DP}_r$

Remember: These assumptions may not always reflect reality!

PCA intuition

Rotate axes to match highest variance directions + choose the best new axes

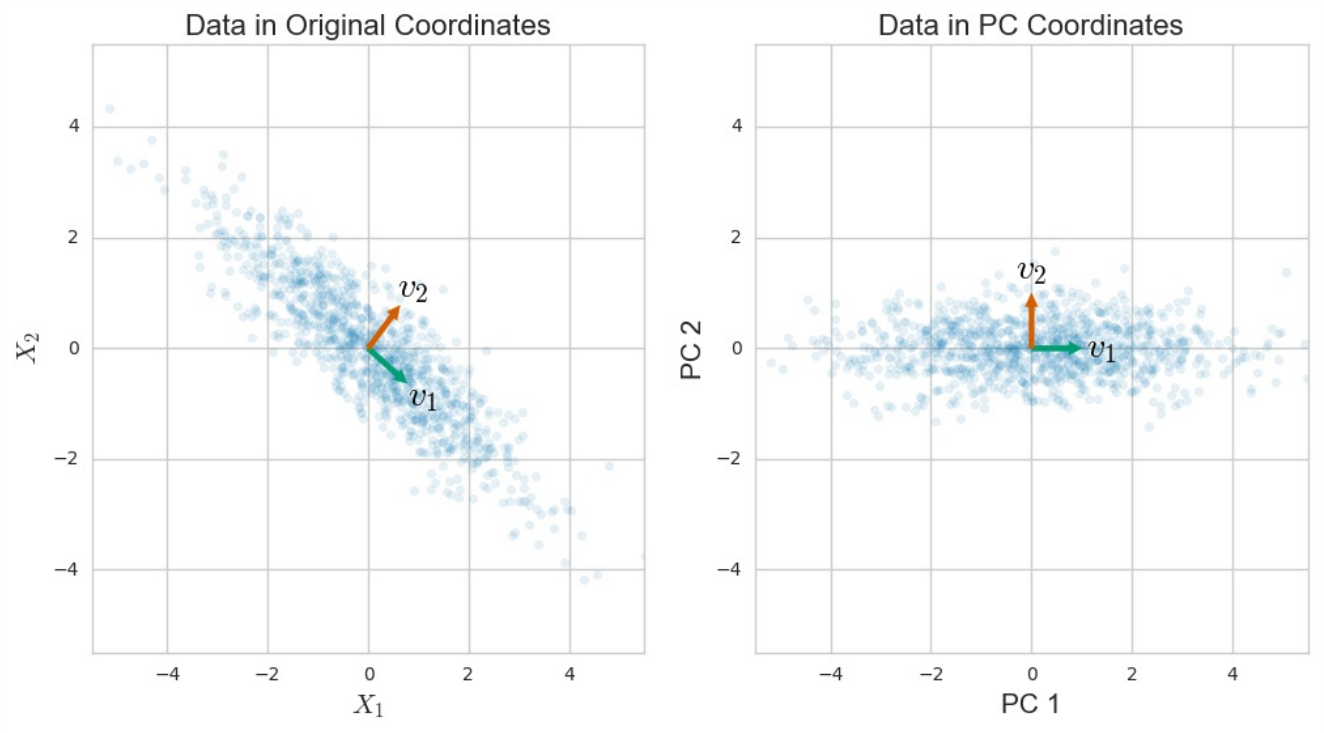


Image source: https://intoli.com/blog/pca-and-svd

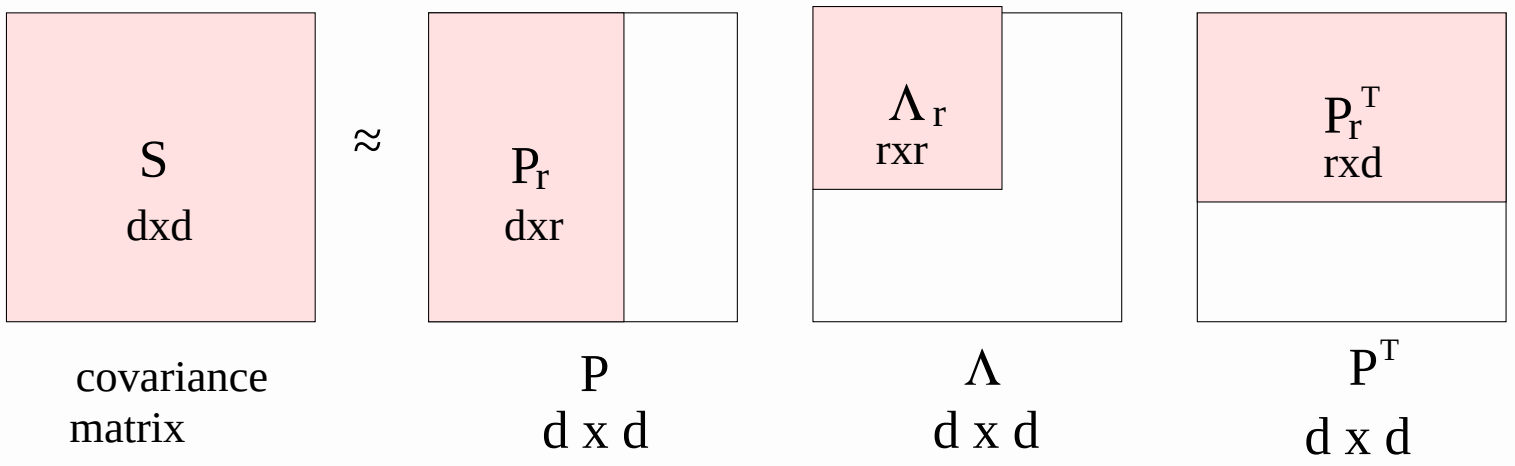
PCA: Use eigen-decomposition

- Assume $\mathbf{D}$ mean-centered. Covariance matrix $\mathbf{C} = \frac{1}{n-1}\mathbf{D}^T\mathbf{D}$ [a]
- since $\mathbf{C}$ positive semidefinite, it can be diagonalized:
  $$\mathbf{C} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T$$
  - $\mathbf{P}$'s columns=$\mathbf{C}$'s orthonormal eigenvectors
  - $\mathbf{\Lambda}$ diagonal, $\mathbf{\Lambda}_{ii}$=eigenvalues
- transformed data $\mathbf{D}' = \mathbf{DP}$
  - $\mathbf{\Lambda}$ new covariance matrix (diagonal, i.e., no correlations)

[a] unbiased estimate; for large n also $\frac{1}{n}\mathbf{D}^T\mathbf{D}$ ok

PCA: Dimension reduction

- Assume $\mathbf{\Lambda}$ ordered into decreasing order by $\lambda_i = \mathbf{\Lambda}_{ii}$
- Keep only r largest $(\lambda_1, \ldots, \lambda_r)$ + corresponding eigenvectors
- approximate data $\mathbf{D'} = \mathbf{DP}_r$



Example: Dimension reduction with PCA

Data $\mathbf{D}$

$$\begin{bmatrix} 2 & 2 & 1 & 2 & 0 & 0 \\ 2 & 3 & 3 & 3 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 3 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 & 1 & 2 \end{bmatrix}$$

Mean-centered data [a]

$$\begin{bmatrix} 0.83 & 0.67 & -0.17 & 0.00 & -0.50 & -0.67 \\ 0.83 & 1.67 & 1.83 & 1.00 & -0.50 & -0.67 \\ -0.17 & -0.33 & -0.17 & -1.00 & -0.50 & -0.67 \\ 0.83 & 0.67 & 0.83 & 1.00 & 0.50 & 0.33 \\ -1.17 & -1.33 & -1.17 & -1.00 & 0.50 & 0.33 \\ -1.17 & -1.33 & -1.17 & 0.00 & 0.50 & 1.33 \end{bmatrix}$$

[a] rounded for the slide presentation only!

Mean vector $\approx$ [1.167, 1.333, 1.167, 2, 0.5, 0.667]

Example (continued)

Covariance matrix [a]:

$$\mathbf{C} \approx \begin{bmatrix} 0.97 & 1.13 & 0.97 & 0.6 & -0.3 & -0.53 \\ 1.13 & 1.47 & 1.33 & 0.8 & -0.4 & -0.67 \\ 0.97 & 1.33 & 1.37 & 0.80 & -0.3 & -0.53 \\ 0.60 & 0.80 & 0.80 & 0.80 & 0.00 & 0.00 \\ -0.30 & -0.40 & -0.30 & 0.00 & 0.30 & 0.40 \\ -0.53 & -0.67 & -0.53 & 0.00 & 0.40 & 0.67 \end{bmatrix}$$

[a] rounded for the slide presentation only!

Example (continued)

Eigenvalues λ_i: 4.43, 0.89, 0.17, 0.066, 0.014, 1.2e-17
Eigenvectors $\mathbf{v}_i$ (column vectors) [a]:

$$\begin{bmatrix} 0.44 & 0.058 & 0.68 & -0.34 & -0.47 & 1.8e-15 \\ 0.57 & 0.027 & 0.092 & 0.18 & 0.54 & 0.58 \\ 0.53 & -0.14 & -0.71 & -0.26 & -0.35 & 1.5e-15 \\ 0.32 & -0.62 & 0.14 & 0.37 & 0.16 & -0.58 \\ -0.15 & -0.42 & 0.041 & -0.78 & 0.43 & 8.4e-17 \\ -0.26 & -0.65 & 0.052 & 0.19 & -0.38 & 0.58 \end{bmatrix}$$

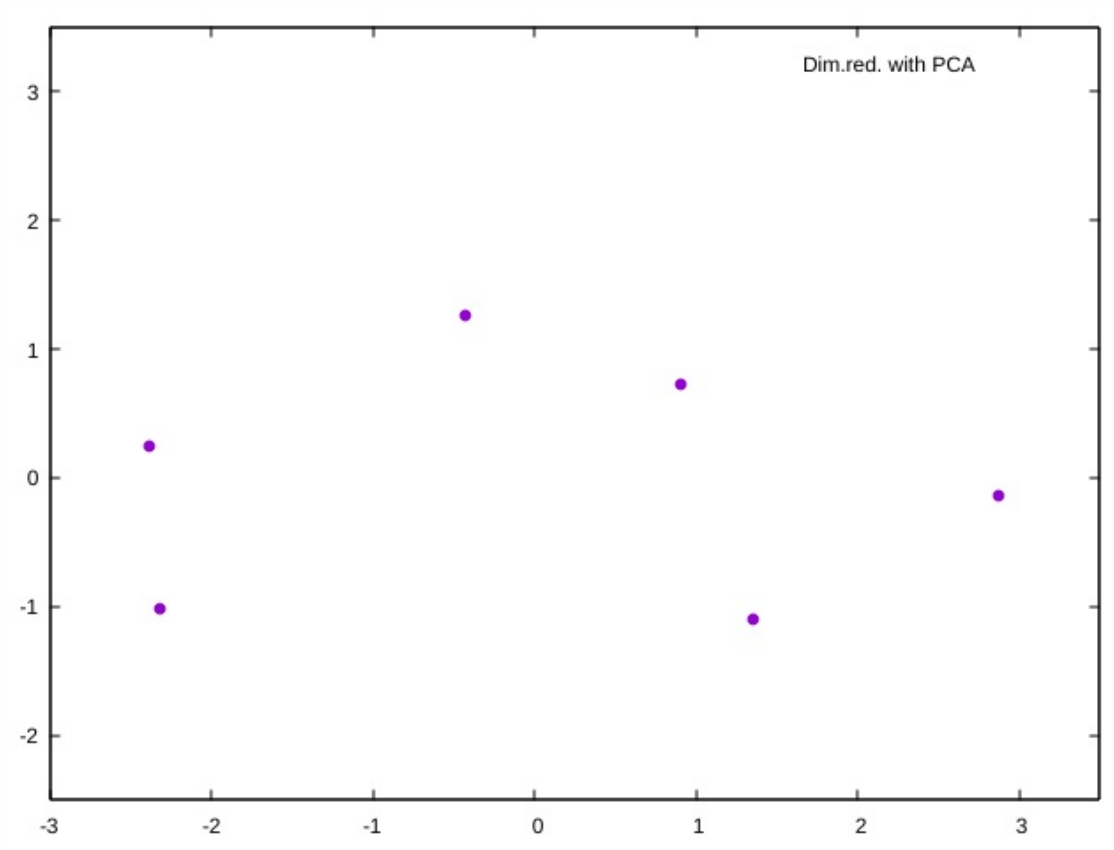
What shall we do if we want a 2-dimensional representation of data?

[a] All rounded for the presentation

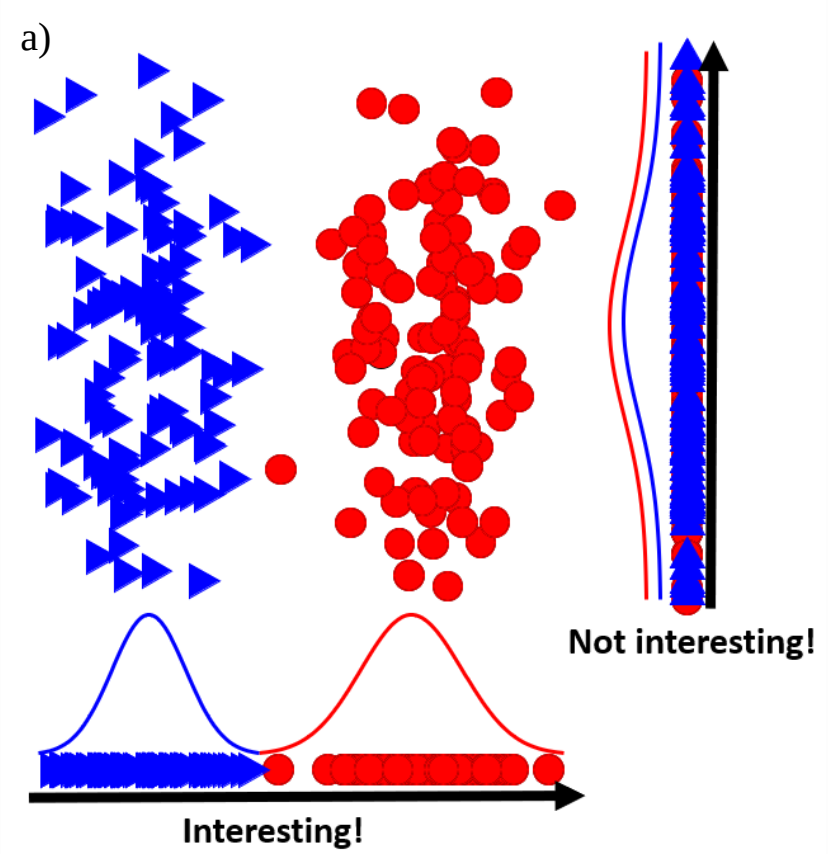
Example (continued)

Since λ_1 and λ_2 largest, set $\mathbf{P}_2 = [\mathbf{v}_1, \mathbf{v}_2]$.

$$\mathbf{D}' = \mathbf{D}\mathbf{P}_2 \approx \begin{bmatrix} 0.91 & 0.73 \\ 2.87 & -0.14 \\ -0.43 & 1.26 \\ 1.35 & -1.09 \\ -2.38 & 0.25 \\ -2.32 & -1.01 \end{bmatrix}$$



When PCA can fail?



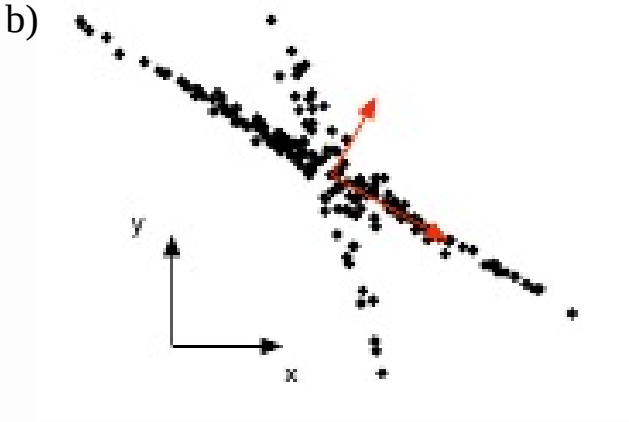


Image sources: https://towardsdatascience.com/interesting-projections-where-pca-fails-fe64ddca73e6 and Shlen's good tutorial: https://arxiv.org/abs/1404.1100

Singular value decompositions (SVD)

Factorize $\mathbf{D}$ as $\mathbf{D} = \mathbf{Q}\mathbf{\Sigma}\mathbf{P}^T$ [a]

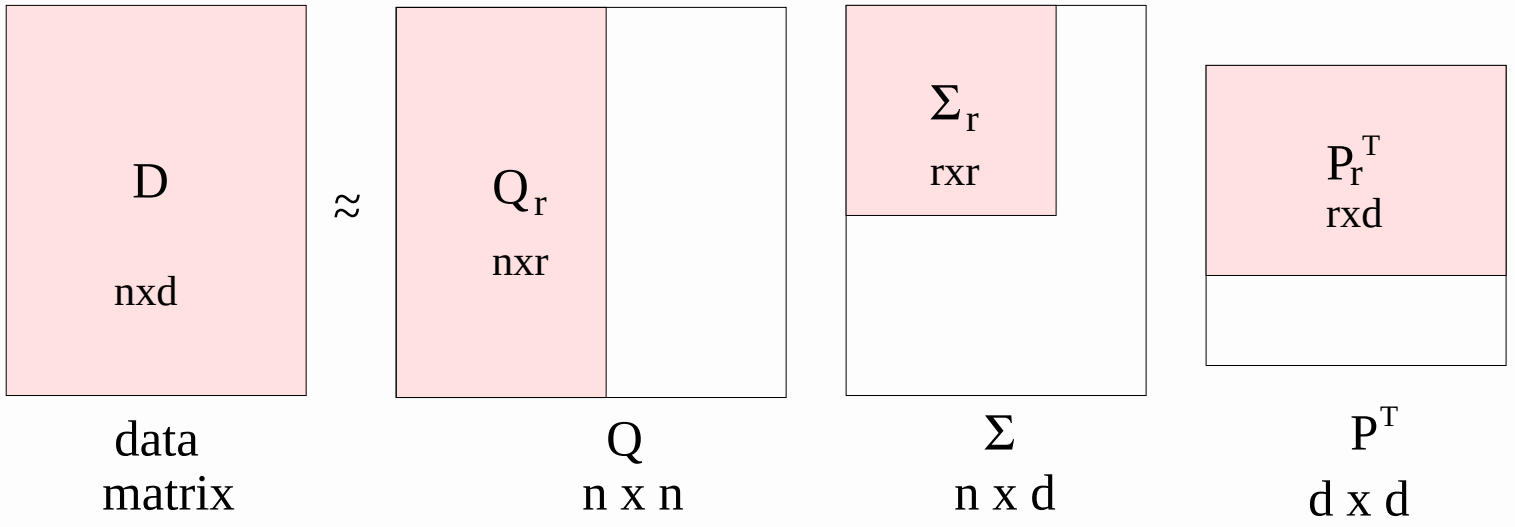
- $\mathbf{\Sigma}$ diagonal, $\mathbf{\Sigma}_{ii} = \sigma_i$ **singular values**
- $\mathbf{Q}$'s columns **left singular vectors**, (orthonormal eigenvectors of $\mathbf{D}\mathbf{D}^T$)
- $\mathbf{P}$'s columns **right singular vectors**, (orthonormal eigenvectors of $\mathbf{D}^T\mathbf{D}$)
- transformed data $\mathbf{D}' = \mathbf{D}\mathbf{P}$
  - if $\mathbf{D}$ mean-centered, same basis vectors as PCA [b]
  - mean-centering often skipped, if $\mathbf{D}$ sparse, non-negative (e.g., document-word matrices)

[a] always possible

[b] may be opposite direction

Truncated SVD: Dimension reduction

- Assume $\mathbf{\Sigma}$ ordered into decreasing order by $\sigma_i = \mathbf{\Sigma}_{ii}$.
- Keep only r largest $(\sigma_1, \ldots, \sigma_r)$ + corresponding singular vectors of $\mathbf{P}$
- approximate data $\mathbf{D}' = \mathbf{D}\mathbf{P}_r$



Previous example with SVD

Let's first test without mean-centering:

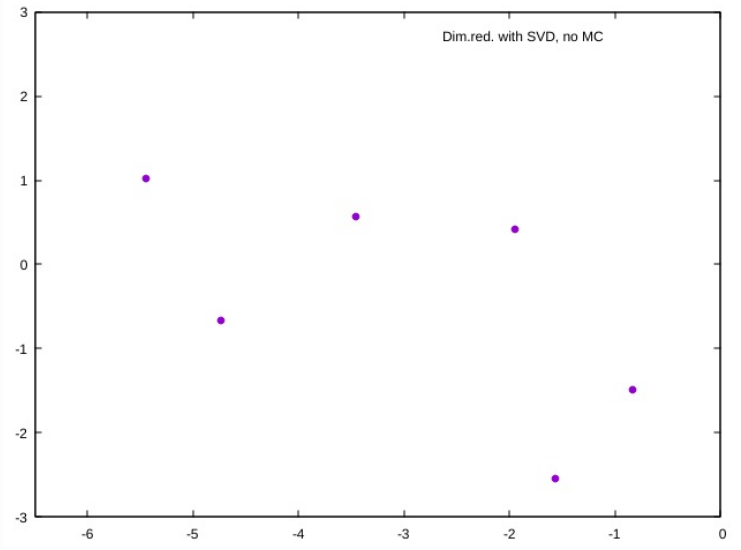
```
Q
[[-4.10936057e-01  1.74569815e-01  8.24528531e-01  2.52257347e-01 -2.39114763e-01  1.29454797e-16]
 [-6.45804322e-01  3.14417109e-01 -5.61570935e-01  3.01160896e-01 -2.79318562e-01  1.25912076e-16]
 [-2.31559546e-01  1.26698028e-01  3.39347806e-02 -9.93766673e-02  5.02626927e-01  8.16496581e-01]
 [-5.62143219e-01 -2.03086484e-01  4.38362617e-02 -6.02554461e-01  3.33302743e-01 -4.08248290e-01]
 [-9.90241264e-02 -4.56482541e-01 -2.40332995e-02 -4.03801126e-01 -6.71951112e-01  4.08248290e-01]
 [-1.86112160e-01 -7.77813886e-01 -3.37638835e-02  5.56475872e-01  2.22626206e-01 -3.01457954e-16]]

Sigma
[8.42523943e+00 3.26119142e+00 9.87979172e-01 5.74286469e-01 2.72145612e-01 2.01264667e-17]

P^T
[[-4.11777822e-01 -4.88428975e-01 -4.39654568e-01 -6.11083254e-01 -1.00564442e-01 -1.22654279e-01]
 [ 2.14185191e-01  3.10596922e-01  2.57067462e-01 -3.68663051e-01 -4.40753922e-01 -6.79259973e-01]
 [ 6.55400957e-01  8.69973391e-02 -7.47563302e-01  3.86918624e-02 -1.41307851e-02 -4.83054767e-02]
 [-3.44164653e-01  1.80244178e-01 -2.59009332e-01  3.65859132e-01 -7.83371609e-01  1.85614954e-01]
 [ 4.86378460e-01 -5.39978569e-01  3.38649460e-01 -1.48261638e-01 -4.26323839e-01  3.91716930e-01]
 [-0.00000000e+00 -5.77350269e-01  1.66533454e-16  5.77350269e-01  9.12464548e-16 -5.77350269e-01]]
```

σ_1 and σ_2 largest $\Rightarrow \mathbf{P}_2$ = first two columns of $\mathbf{P}$
+ calculate $\mathbf{D}' = \mathbf{D}\mathbf{P}_2$

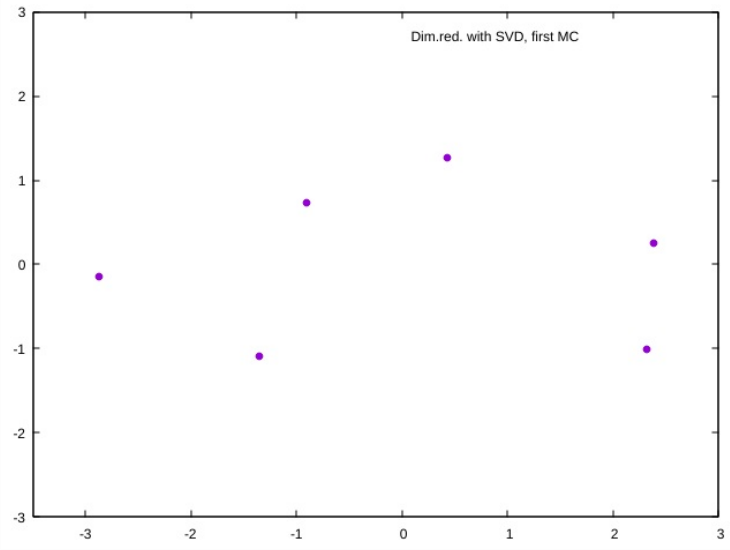
Example continued

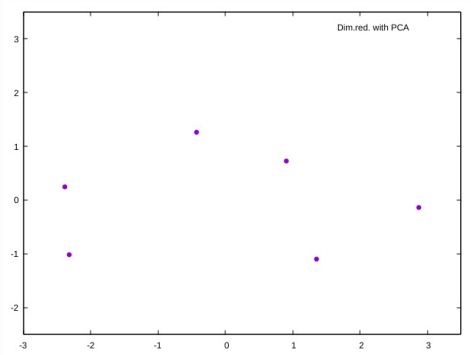
Note: Different $\mathbf{Q}$, $\mathbf{\Sigma}$, and $\mathbf{P}$, if mean-centered data

$\mathbf{D}'$ without mean-centering



$\mathbf{D}'$ with mean-centering





SVD applications

- dimension reduction: $\mathbf{D}' = \mathbf{D}\mathbf{P}_r$
- sometimes $\mathbf{Q}$ also useful, e.g., user-item rating matrix
  - $\mathbf{D}^T\mathbf{Q}_r$ describes items by r latent components
- **Latent semantic analysis (LSA)** applies SVD on document-term matrix (e.g., tf-idf matrix)
  - often drastic dimension reduction!
  - helps with noise due to synonymous words
  - e.g., {(car), (truck), (flower)} $\rightarrow$ {(1.3452 $\cdot$ car + 0.2828 $\cdot$ truck), (flower)}
- noise reduction: truncated SVD tends to correct inconsistencies

Example of truncated SVD (Aggarwal p. 45)

$$D = \begin{pmatrix} 2 & 2 & 1 & 2 & 0 & 0 \\ 2 & 3 & 3 & 3 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 3 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 & 1 & 2 \end{pmatrix} \approx Q_2 \Sigma_2 P_2^T$$

$$\approx \begin{pmatrix} -0.41 & 0.17 \\ -0.65 & 0.31 \\ -0.23 & 0.13 \\ -0.56 & -0.20 \\ -0.10 & -0.46 \\ -0.19 & -0.78 \end{pmatrix} \begin{pmatrix} 8.4 & 0 \\ 0 & 3.3 \end{pmatrix} \begin{pmatrix} -0.41 & -0.49 & -0.44 & -0.61 & -0.10 & -0.12 \\ 0.21 & 0.31 & 0.26 & -0.37 & -0.44 & -0.68 \end{pmatrix}$$

$$= \begin{pmatrix} 1.55 & 1.87 & \underline{1.67} & 1.91 & 0.10 & 0.04 \\ 2.46 & 2.98 & 2.66 & 2.95 & 0.10 & -0.03 \\ 0.89 & 1.08 & 0.96 & 1.04 & 0.01 & -0.04 \\ 1.81 & 2.11 & 1.91 & 3.14 & 0.77 & 1.03 \\ 0.02 & -0.05 & -0.02 & 1.06 & 0.74 & 1.11 \\ 0.10 & -0.02 & 0.04 & 1.89 & 1.28 & 1.92 \end{pmatrix}$$

Summary

1. Factorize $\mathbf{C} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T$ or $\mathbf{D} = \mathbf{Q}\mathbf{\Sigma}\mathbf{P}^T$
2. Use $\mathbf{P}_r$ (best components of $\mathbf{P}$).
   Reduced data $\mathbf{D}' = \mathbf{D}\mathbf{P}_r$

Only heuristics! Work well only if the underlying assumptions are true.