M2 TIW – M2 BIO-INFO

DATA ANALYSIS

Other Data Types Transformations

DATA TRANSFORMATION

- Our data is provided in a given form
  - Tabular (vectors)
  - Network
  - Time series
  - Text
  - Images
  - ….

- To use the full potential of data mining, you might want to study it from multiple angles
  - How to convert from tabular to graph?
  - From Graph to Tabular?
  - From images/text to tabular (embedding)?

DIMENSIONALITY REDUCTION

Low dimensionality embedding

DIMENSIONALITY REDUCTION

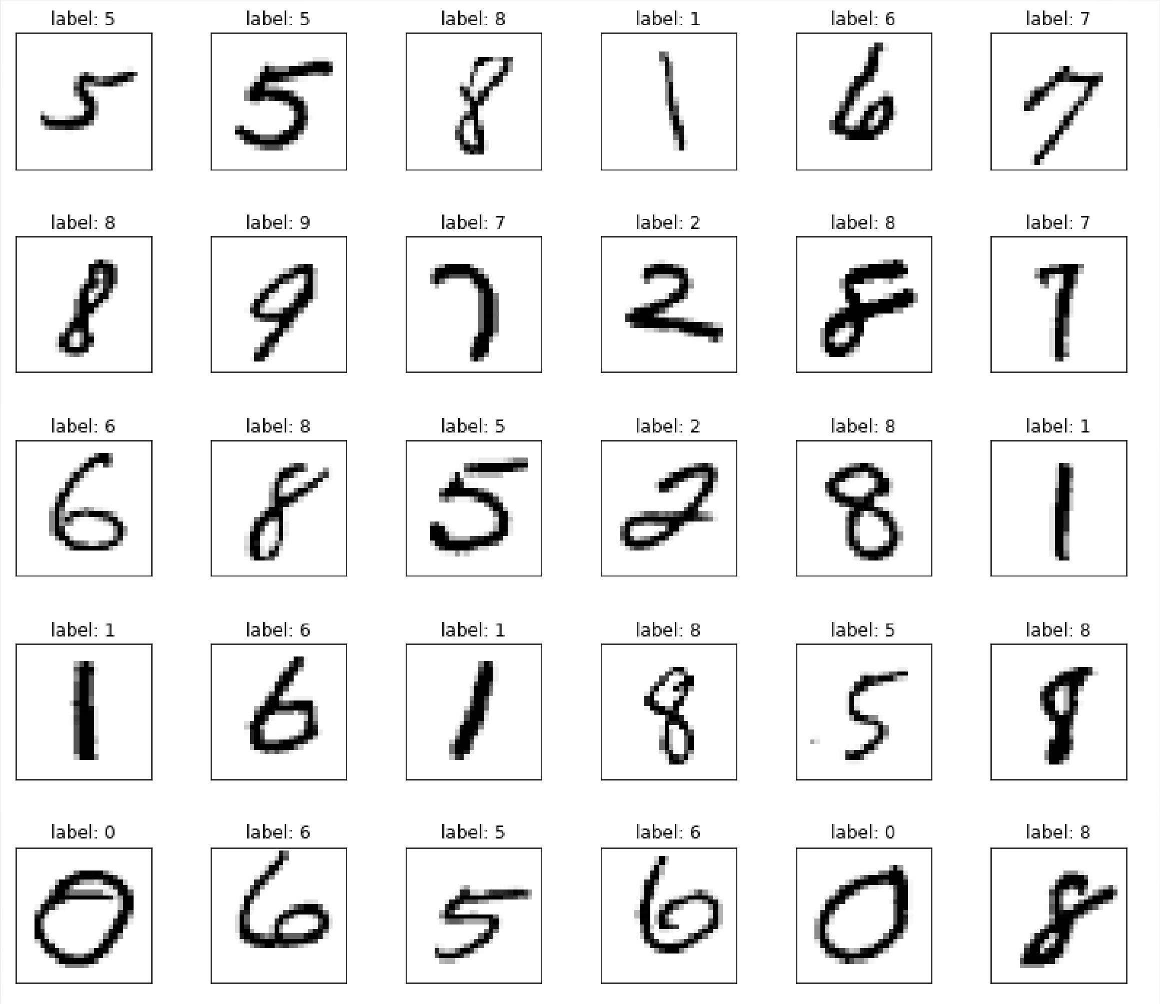
- Data Mining objective: understand our data
  - We get a dataset composed of many features
    - Or worst, complex object (image, sound, graph…)
  - How to understand the organization of our data?
  - How to perform clustering?

VISUALIZATION

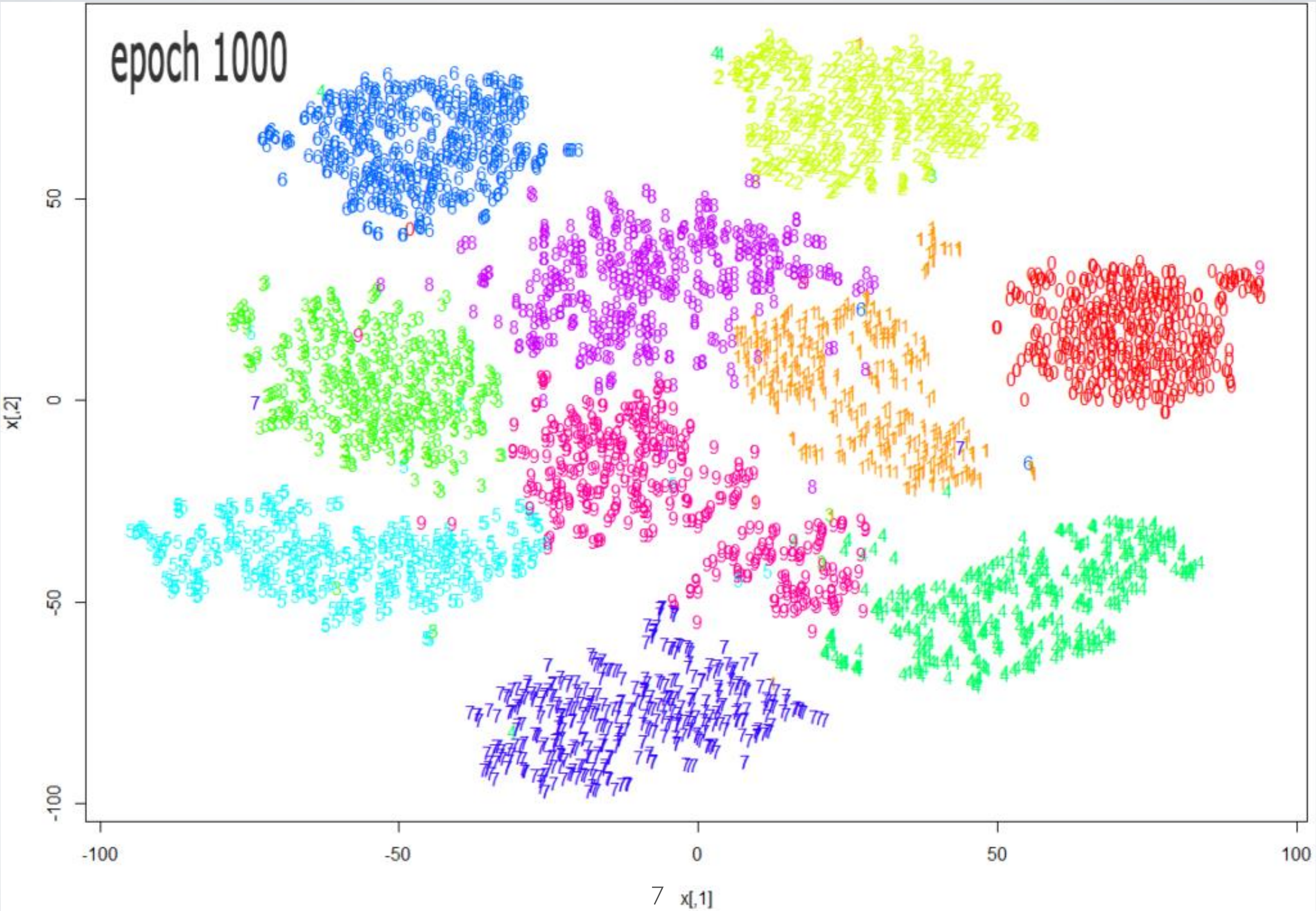
- Your data is perfectly fine, but you want to intuitively understand how it is organized
  - Are there groups of similar objects?
  - Are my clusters meaningful?
  - Is my classification/clustering on some types of elements and not others.

VISUALIZATION

Example: MNIST Dataset
Each pixel is a variable



t-SNE embedding



CURSE OF DIMENSIONALITY

- Having hundreds/thousands of attributes is a problem for data analysis.
  - e.g.: medicine: blood analysis, genomics….
  - e.g.: cooking recipes: each column an ingredient…
- We want to reduce the number of attributes while keeping most of the information
- Also helps with scalability

CORRELATION

- Assume that you have correlated features such as age, height and weight.
  - Redundancy ! Computational Inefficiency
    - e.g., Decision tree will spend a lot of time choosing between them for no reason
  - Risk of overfitting
    - noise between correlated variables used to distinguish individuals
  - Model interpretability
    - e.g., a model will say that y depends on x or w randomly, if x and w correlated

- Dimensionality reduction can create a single variable to capture what is common
  - The rest can be lost or captured by another feature,
    - Engine horsepower, Car weight, Fuel Consumption
    - =>Performance index (horsepower and weight)
    - =>Efficiency score (weight and fuel consumption)

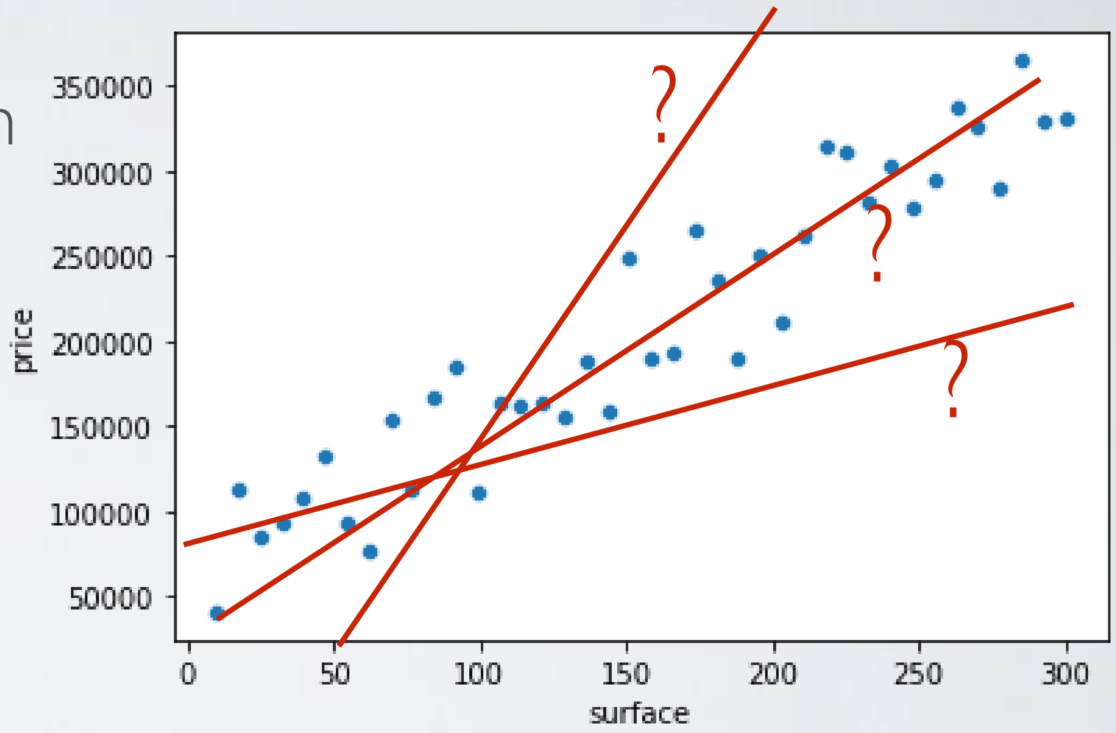
PCA

PCA

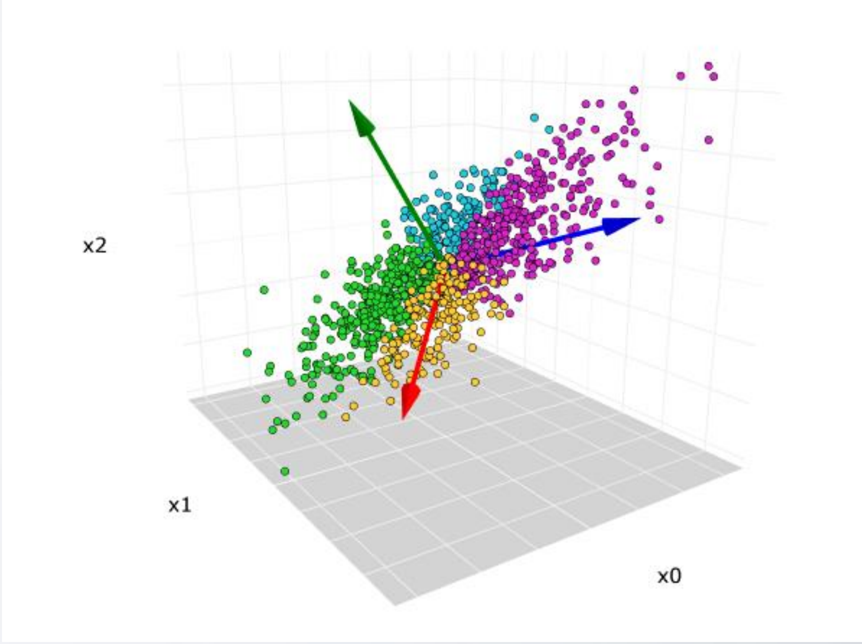
- PCA: Principal Component Analysis
- Defines new dimensions that are linear combinations of initial dimensions
  - Objective: concentrate the **variance** on some dimensions
    - So that we can keep only these ones.
    - Those we remove contain low variance, thus low information

PCA

- Algorithm:
  - 1)Find an "axis", a unit vector defining a line in the space
    - That minimizes the variance=>the squared distance from all points to that line

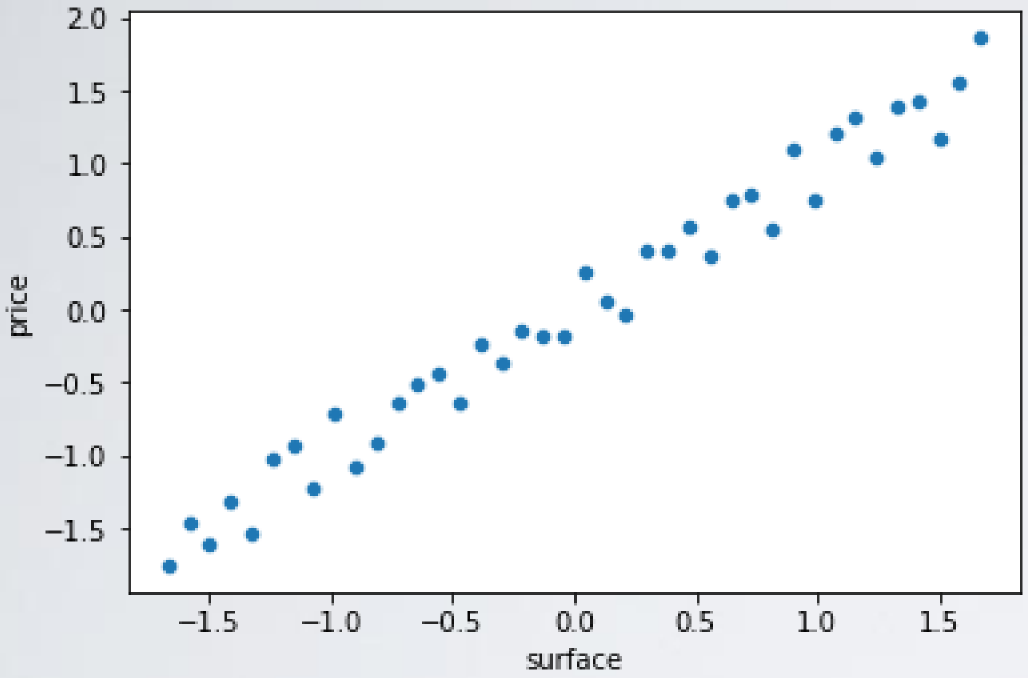


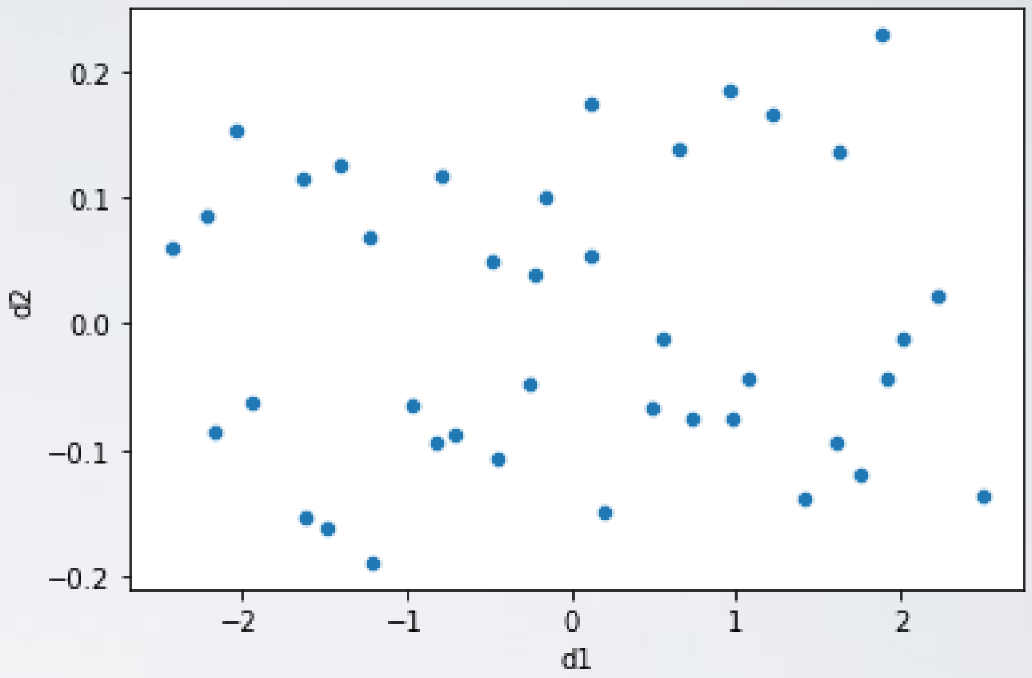
- 2)**For** d **in** [2:(initial_d)]
  - Find another axis, with two constraints:
    - Orthogonal to all previous axis
    - Among those, minimizing the variance



- 3)At the end, keep the first k dimensions
  - Some information is lost

EXAMPLE PCA 2D





Covariance matrix (original)

[1. , 0.98675899],
[0.98675899, 1.]

Sum of variance

2

Variance by dimension

1 1

Covariance matrix (pca)

[1.98675899e+00, 0],
[0, 1.32410092e-02]

Sum of variance

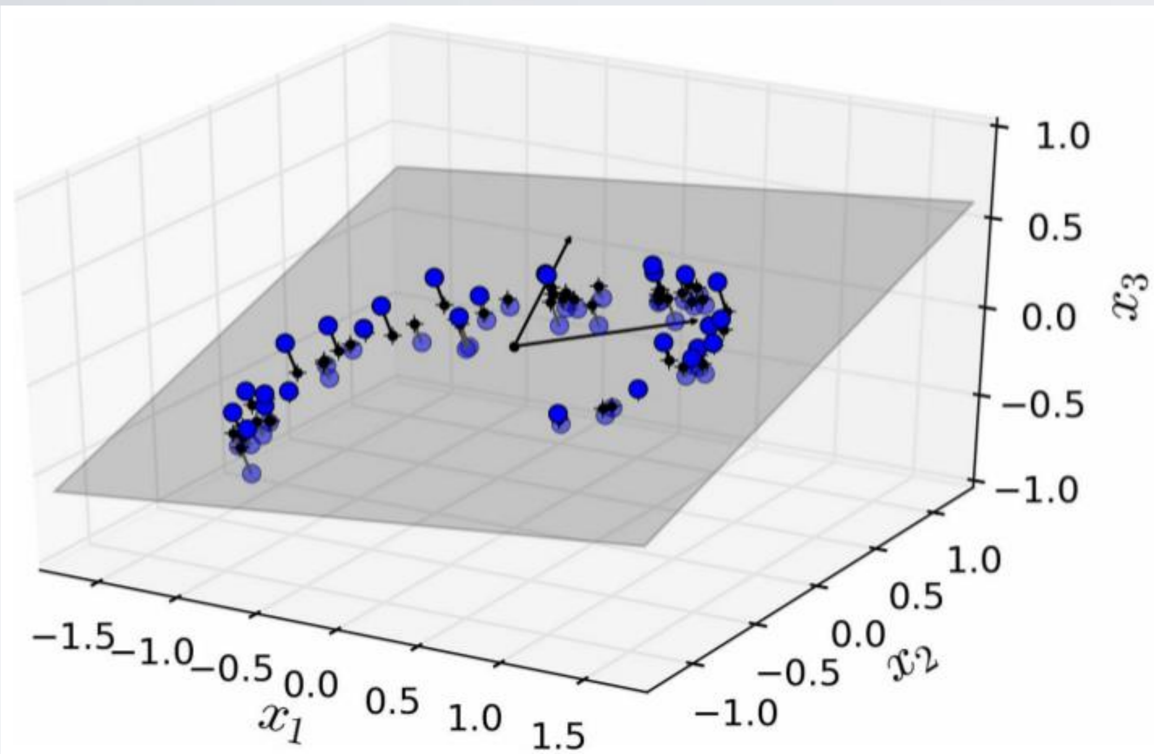
2

Variance by dimension

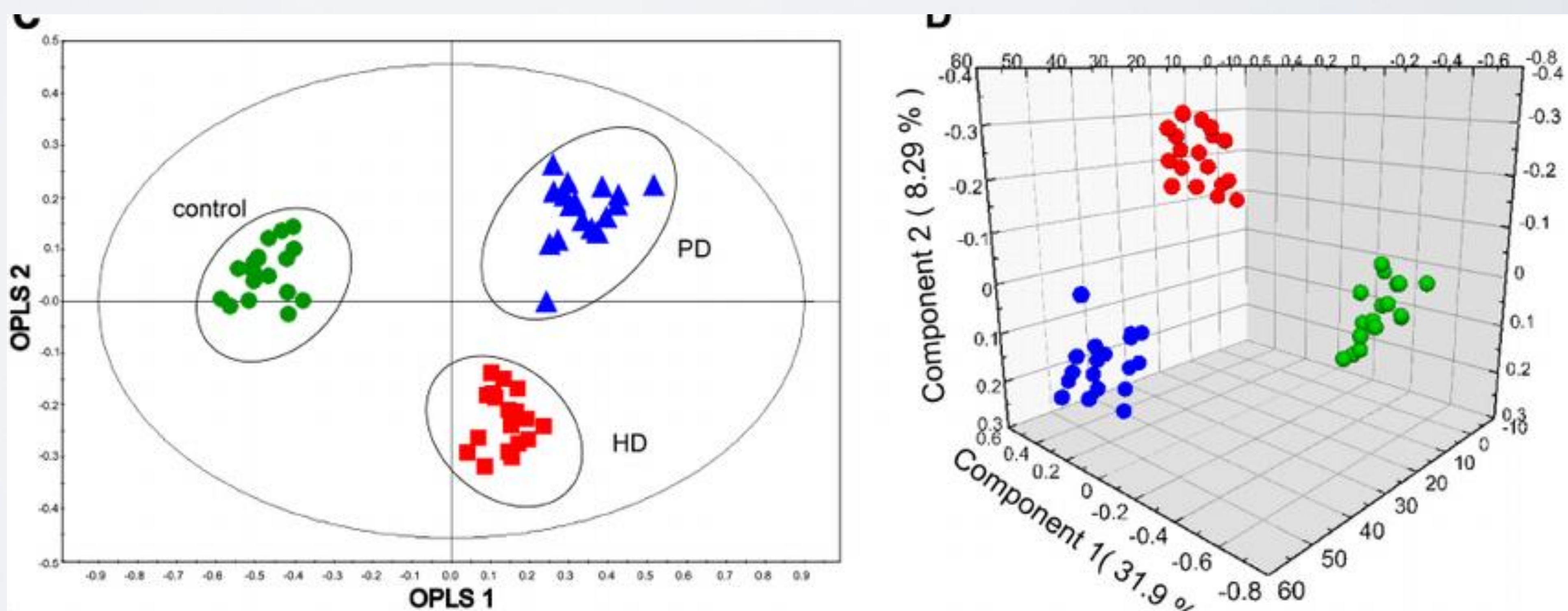
1.98675899 0.01324101

Explained variance(ratio) [0.9933795, 0.0066205]

3D=>2D



1.0
0.5
0.0
−0.5
−1.0
x_3
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
x_1
1.0
0.5
0.0
−0.5
−1.0
x_2

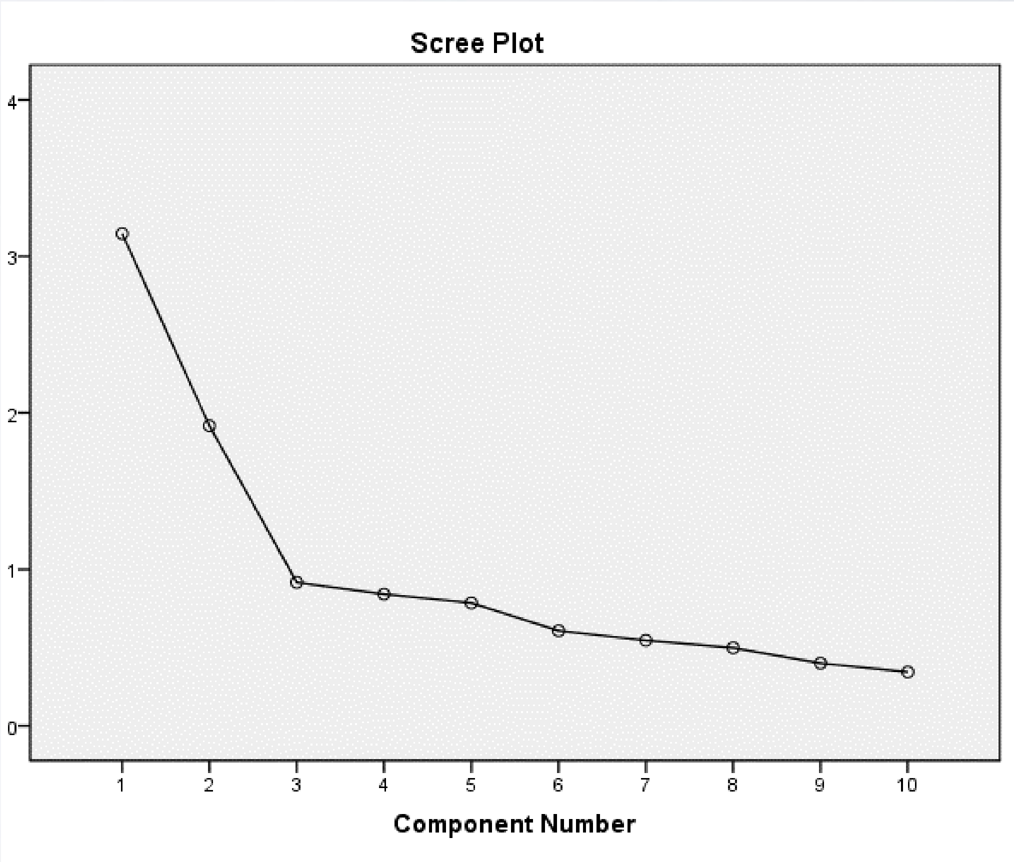


C
control
PD
HD
OPLS 1
OPLS 2
D
Component 2 (8.29 %)
Component 1(31.9 %)

CHOOSING COMPONENTS

- How to choose k?
  - Elbow method… BIC/AIC…
  - OR fix beforehand a min threshold of explained variance, e.g.: 80%
    - We are fine with losing 20% of information
  - If there is a downstream task, cross-validation

Explained variance



COMPUTATION IN PRACTICE

- From standardized dataset X

- Method 1:
  - 1)Compute the Covariance Matrix ($X^T X$)
    - => Linear Correlation Matrix
  - 2) Find the eigenvectors of this matrix
    - $X^T X = V \Lambda V^T$
    - V: eingenvectors = Pincipal components, Λ: Eigenvalues, = explained variance

- Method 2:
  - Apply SVD matrix decomposition
  - $X = U \Sigma V^T$
    - U: left singular vectors. Σ: diagonal matrix with the singular values, V^T:right singular vectors (the principal components)

COMPUTATION IN PRACTICE

- V are the principal components
- Computing the new positions for each observation:
  - XV

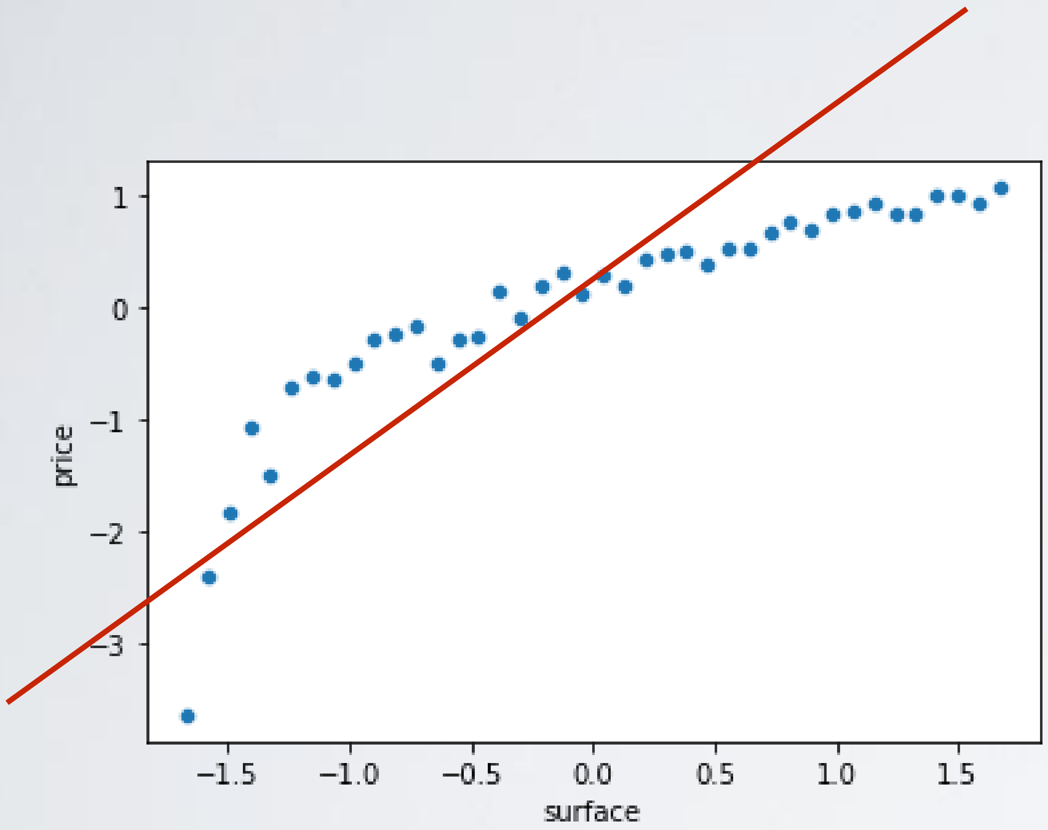
PCA POPULARITY

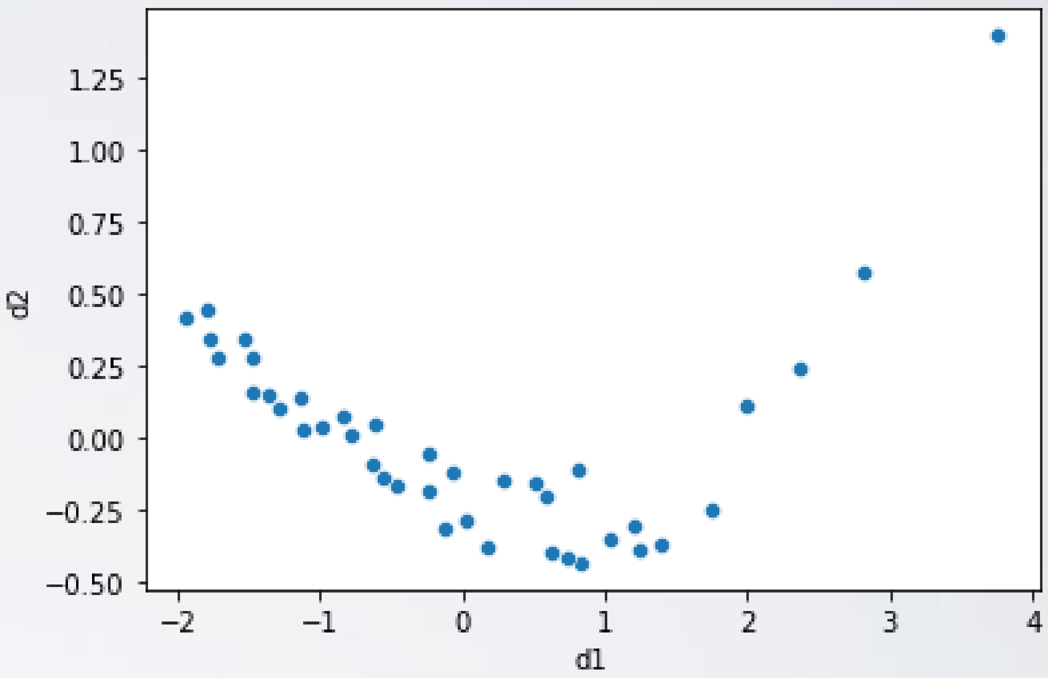
- Why is PCA popular?

- Similar reasons than linear regression:
  - Useful
    - Eliminate correlations
  - Analytical solutions
    - Guarantee to find the global minimum of the objective
    - Could be done before modern computers
  - Interpretable solution
  - Intuitively pleasant

- No reason to consider it “better” than other methods for demensionality reduction…

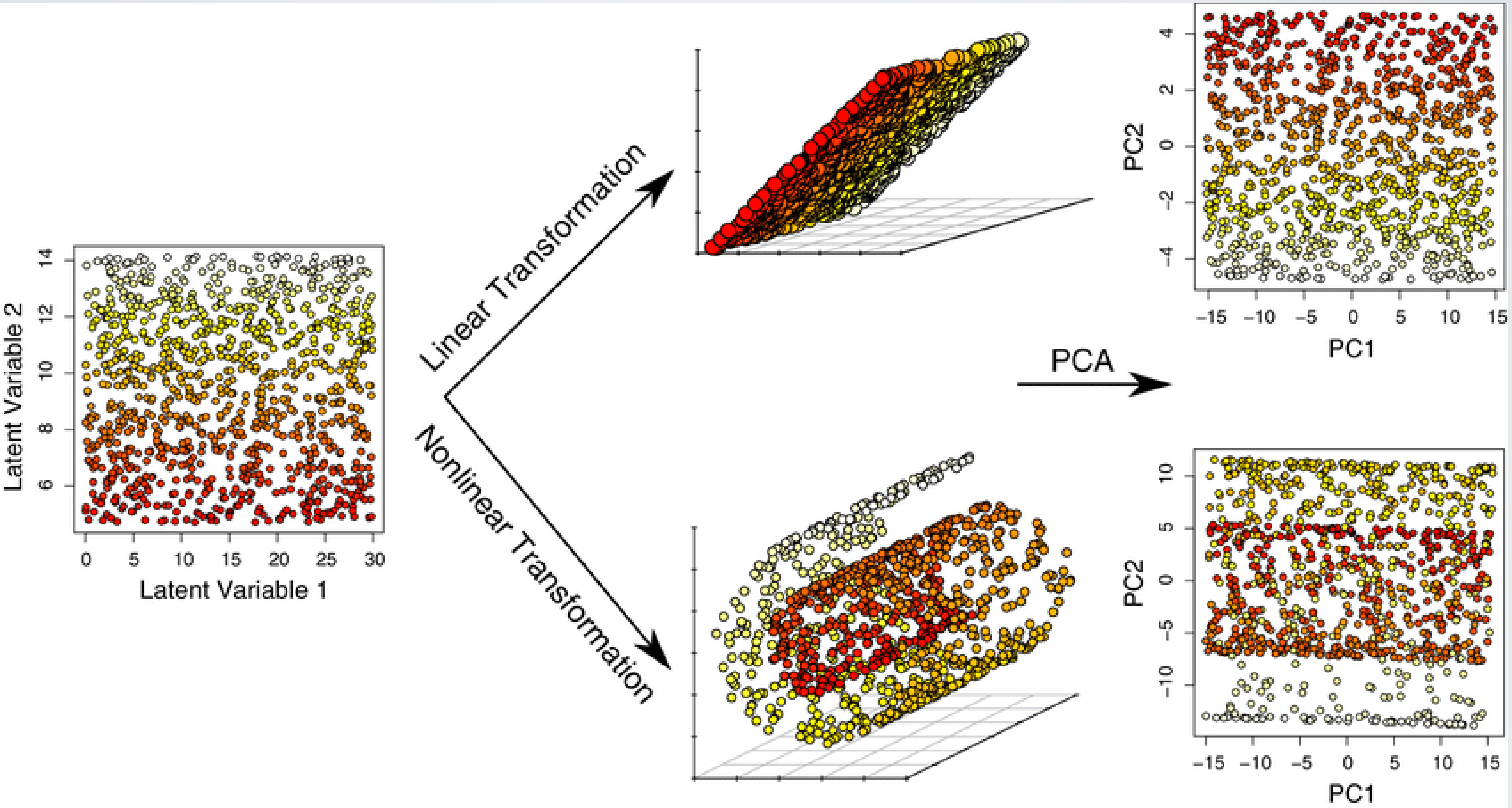
NON-LINEAR SITUATIONS





Pearson correlation(d1,d2): 0

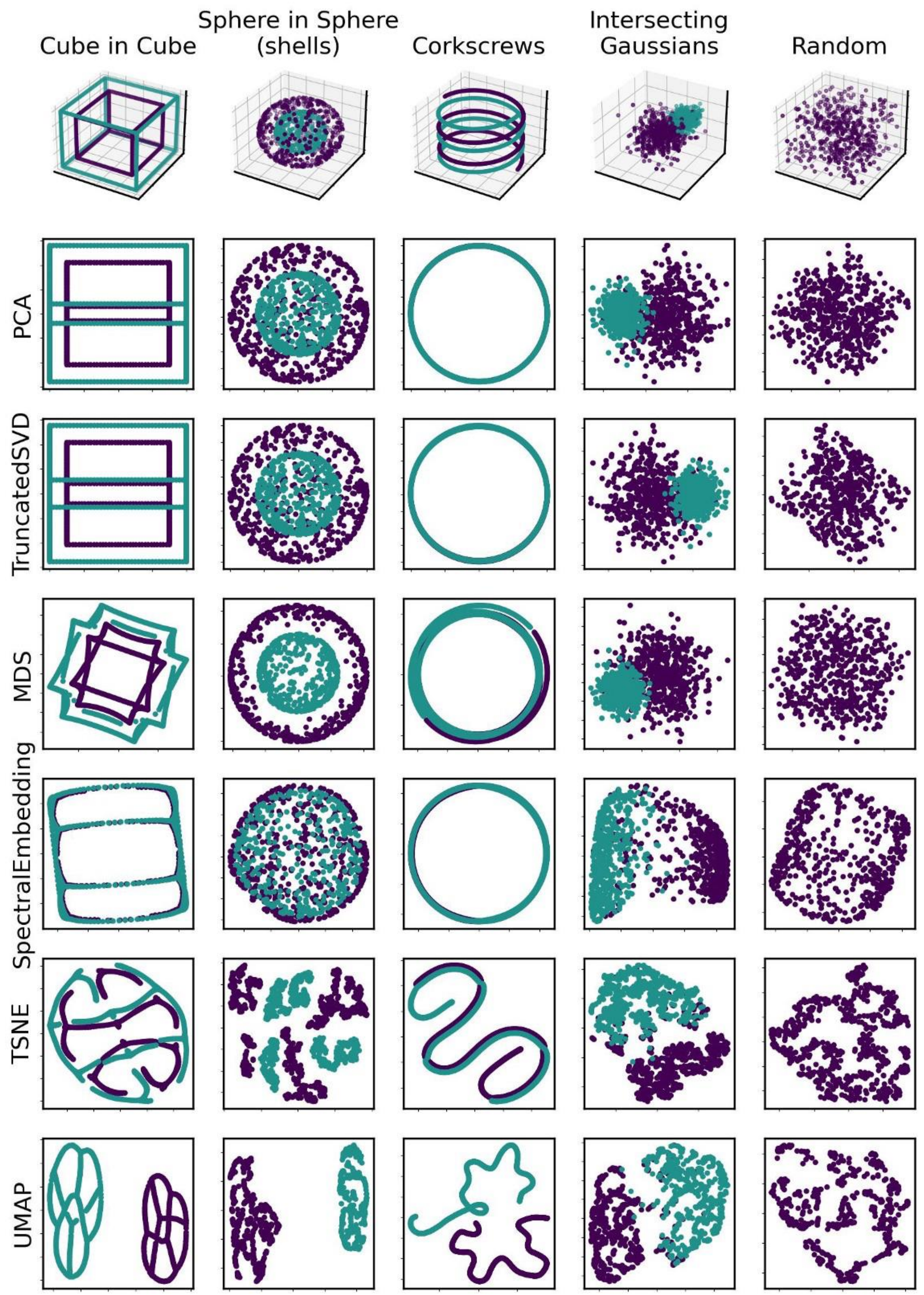
NONLINEAR DATA



MANIFOLDS

MANIFOLDS

- Manifolds are another approach to dimensionality reduction
- The general principle is to
  - 1)Define a notion of distance between elements in the original space
  - 2)Define a notion of distance between elements in a reduced, target space
  - 3)Minimize the difference between distances in original and target space
- In many cases, the process is nonlinear, i.e., we choose distances such as
  - We care more about preserving the distance for items “close” in space than for those “far” from each other


Cube in Cube
Sphere in Sphere (shells)
Corkscrews
Intersecting Gaussians
Random
PCA
TruncatedSVD
MDS
SpectralEmbedding
TSNE
UMAP

MDS

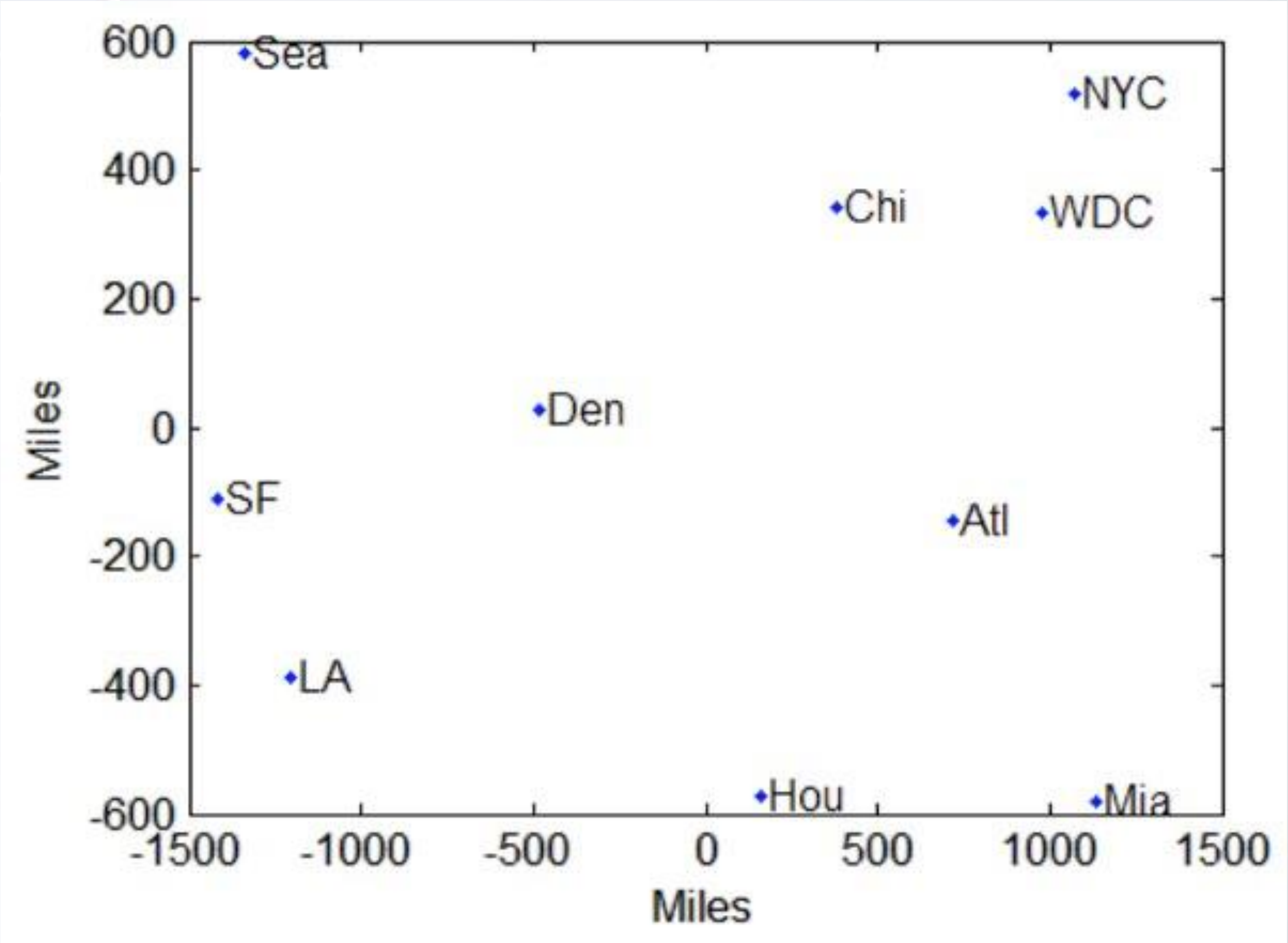
- MDS: Multi-dimensional Scaling:
  - Simply minimize distance between original space and target space
    - e.g., d-dimensional forced to 2-dimensional

- How to do it?
  - 1)Compute all (squared Euclidean) pairwise distances between items=>**Similarity matrix**
    - n x f matrix => n x n matrix
    - Apply double-centering (remove row and column means)
  - 2)Compute PCA on this similarity matrix

- Problems:
    - Very costly (nb features=nb elements), n^2
    - Try to preserve all distances, therefore extremely constrained

MDS

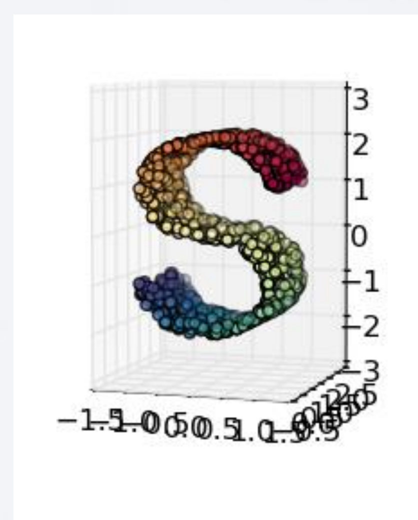
	Atl	Chi	Den	Hou	LA	Mia	NYC	SF	Sea	WDC
Atl	0	587	1212	701	1936	604	748	2139	2182	543
Chi	587	0	920	940	1745	1188	713	1858	1737	597
Den	1212	920	0	879	831	1726	1631	949	1021	1494
Hou	701	940	879	0	1374	968	1420	1645	1891	1220
LA	1936	1745	831	1374	0	2339	2451	347	959	2300
Mia	604	1188	1726	968	2339	0	1092	2594	2734	923
NYC	748	713	1631	1420	2451	1092	0	2571	2408	205
SF	2139	1858	949	1645	347	2594	2571	0	678	[illegible]
Sea	2182	1737	1021	1891	959	2734	2408	678	0	[illegible]
WDC	543	597	1494	1220	2300	923	205	2442	2329	

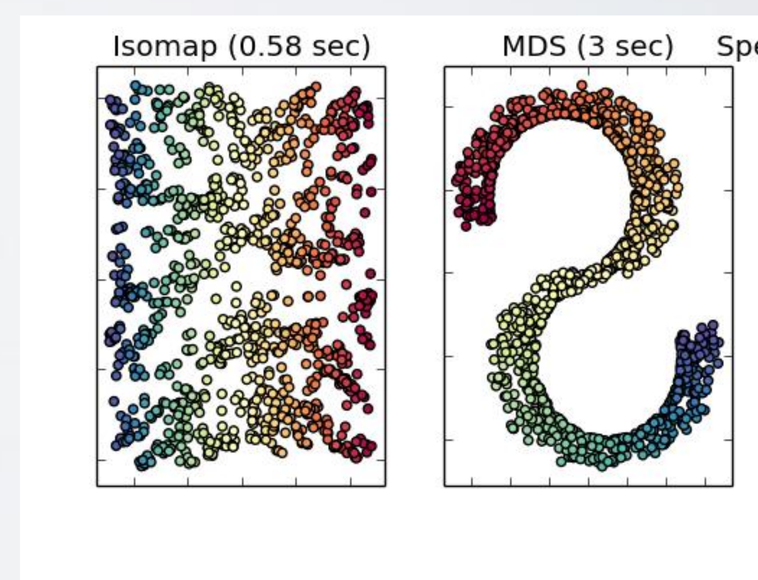


ISOMAP

- Variation of MDS
  - 1)We define a **graph** such as two elements are connected if they are at distance<threshold. (Alternative: fixed number of neighbors)
    - Put a weight on edges=euclidean distance
  - 2)Compute a **similarity** matrix, such as distance = weighted shortest path distance
  - 3)Apply MDS on it

- Non-linear distances





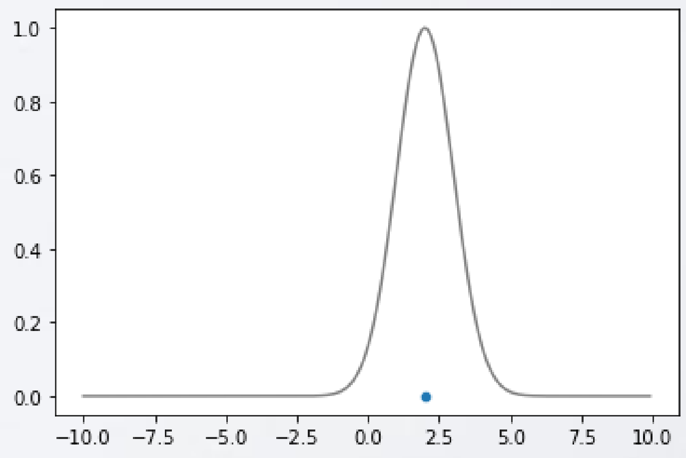
T-SNE

T-SNE

- t-SNE : t-distributed stochastic neighbor embedding
- Non-linear dimensionality reduction
- One of the most popular method for visualizing data in low dimensions
- Similar to MDS/Isomap, but:
  - Do not try to preserve long distance at all
    - Can preserve local structure but loose global one
  - Optimized via gradient descent
    - No way to guarantee global optimum

SNE

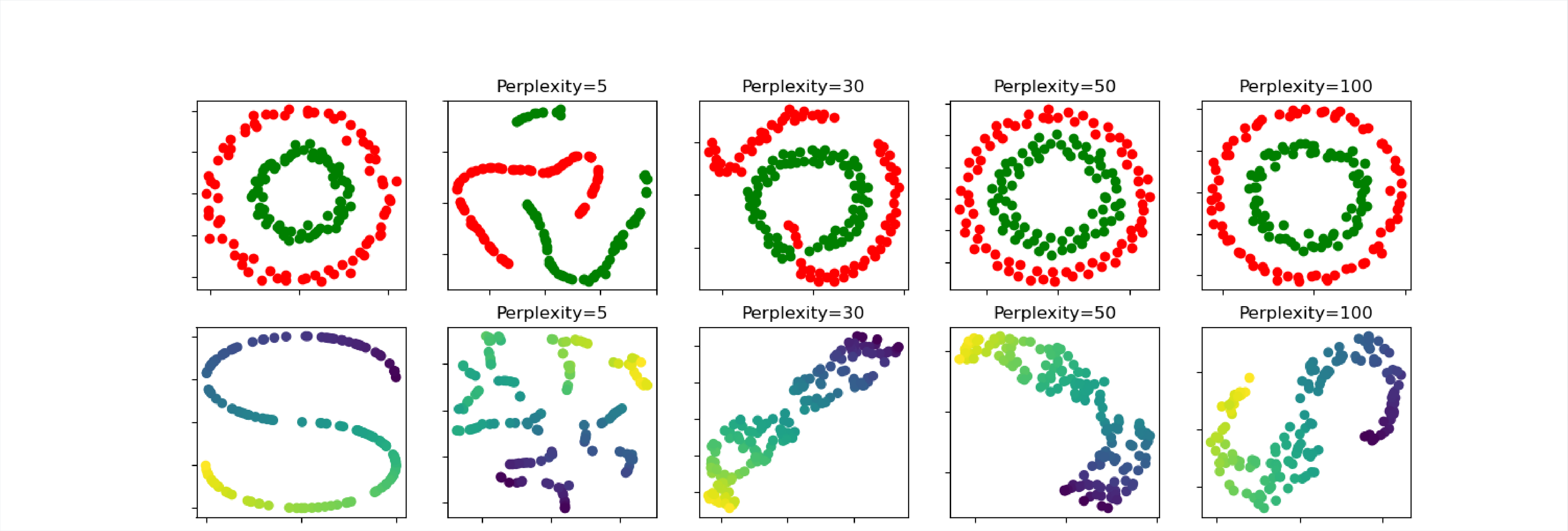
- General principle:
  - Define a notion of similarity $p_{j|i}$ in the high dimensional space P
    - Based on normal distribution
  - Define a notion of similarity $q_{j|i}$ in the low dimensional space Q
    - Based on student-t distribution, tends to "exaggerate" differences
  - For each point of initial coordinates x_i, find a new coordinate y_i in the lower dimensional space, such as to minimize the difference between P and Q
    - $\forall_{i,j} p_{j|i} \approx q_{j|i}$



T-SNE: PERPLEXITY

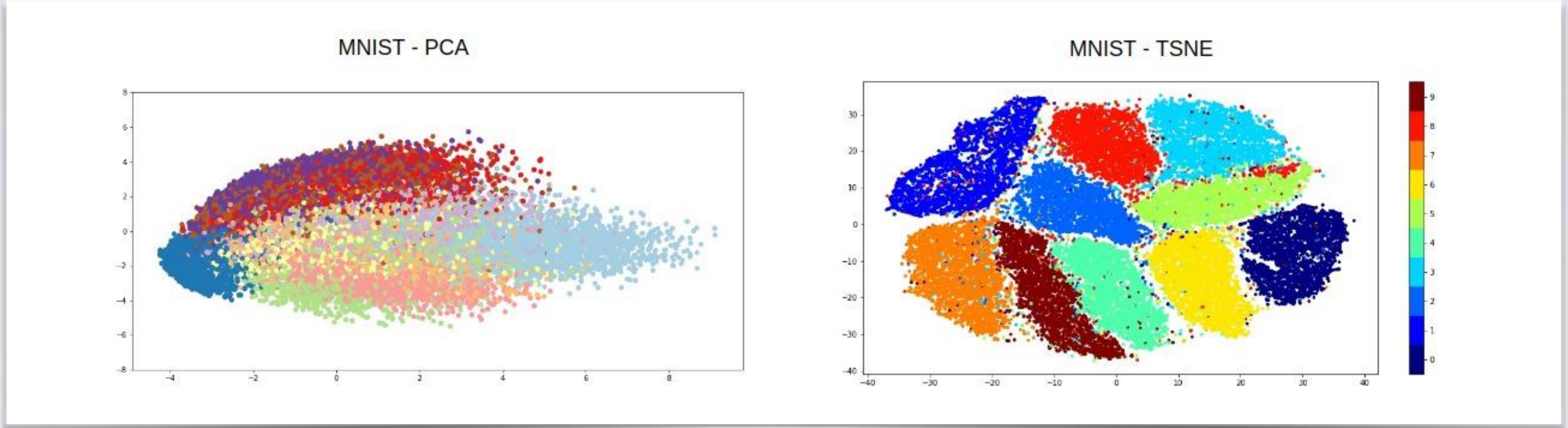
- There is a perplexity parameter σ: it controls how much each point cares more about close neighbors compared with farther neighbors
  - Low σ: Preserve mostly local distances
  - High σ: Give more importance to long-range distances
    - More expensive, more similar to MDS

INFLUENCE OF PERPLEXITY



More like Isomap

More like MDS


MNIST - PCA
MNIST - TSNE

LOW DIMENSIONAL EMBEDDINGS

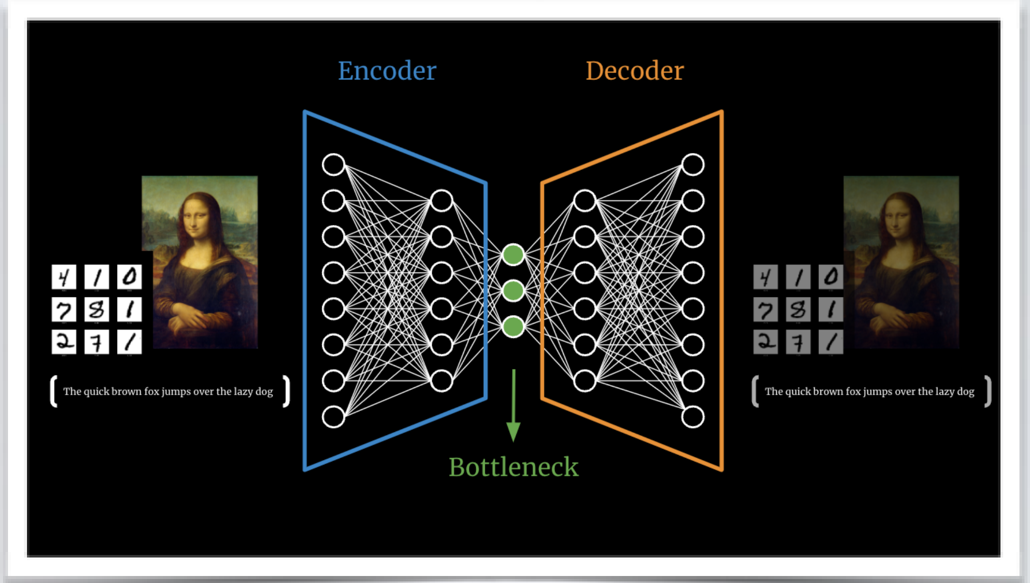
EMBEDDINGS

- A recent usage of low dimensional embeddings is to encode complex objects as vectors
  - Words as Vector => Word2Vec
  - Nodes (of graph) as Vectors => Node2Vec
  - Documents as Vectors => Doc2Vec
  - ….

NODE2VEC

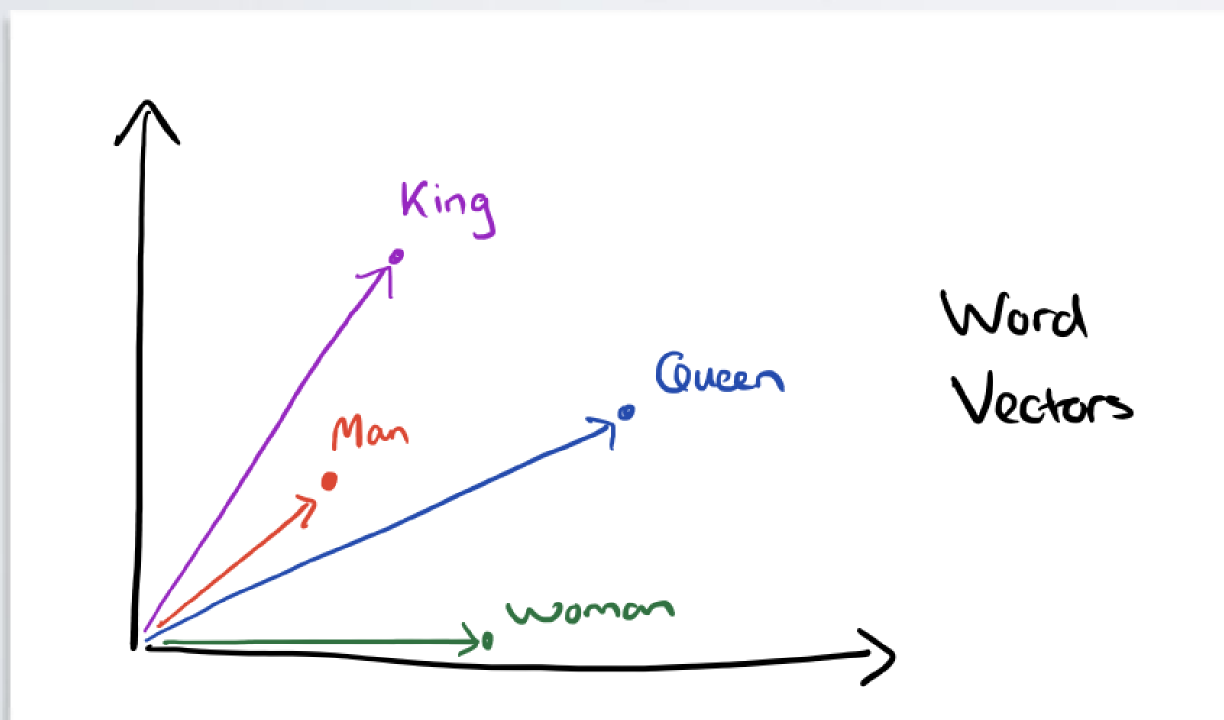
- Shallow neural network method
- Objective similar to MDS/tSNE:
  - Minimize the difference between **graph distance** and **embedding distance**
  - Parameter to tune local/long-distance graph distance
  - (Example implementation: https://github.com/eliorc/node2vec)

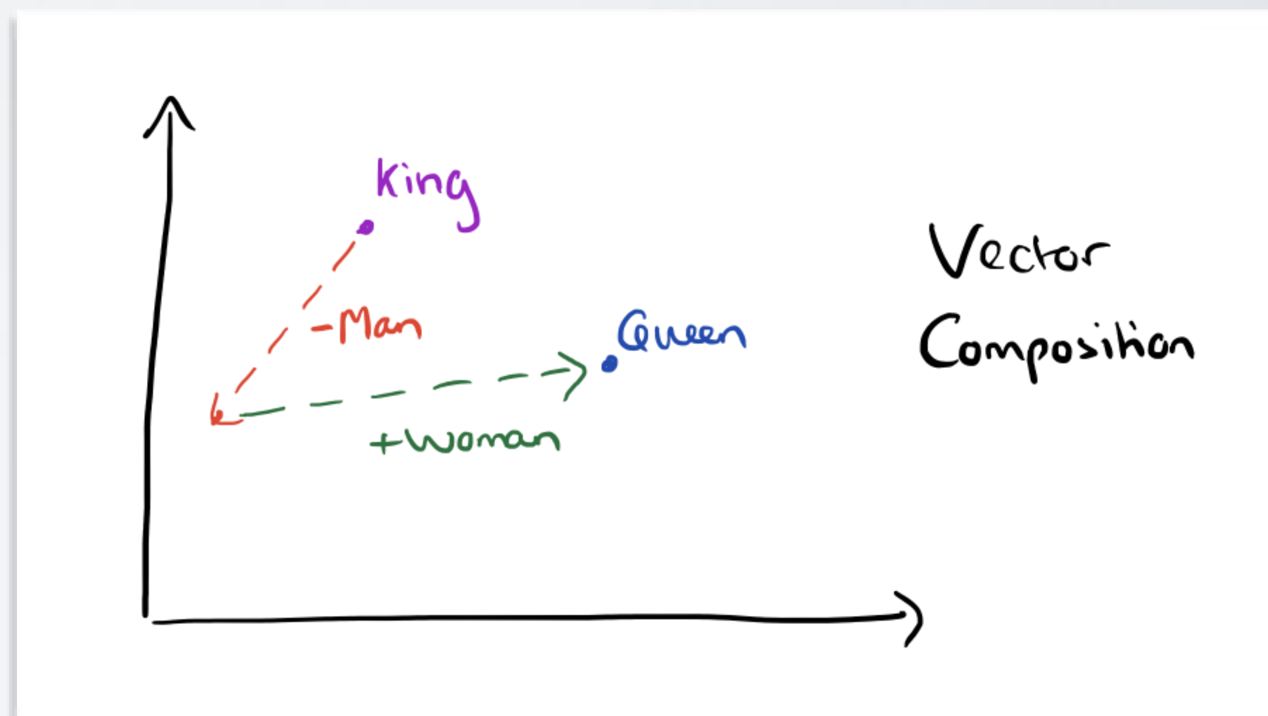
AUTO-ENCODER



- Deep learning approach
- Autoencoder:
  - DECODER neural network learns to reconstruct an input object from a small vector
  - ENCODER neural network learns to encode an input object into a small vector (to maximize reconstructability)
- Created for images, work similarly for texts or graphs
  - (Variational GAE)

EMBEDDINGS





[https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/]

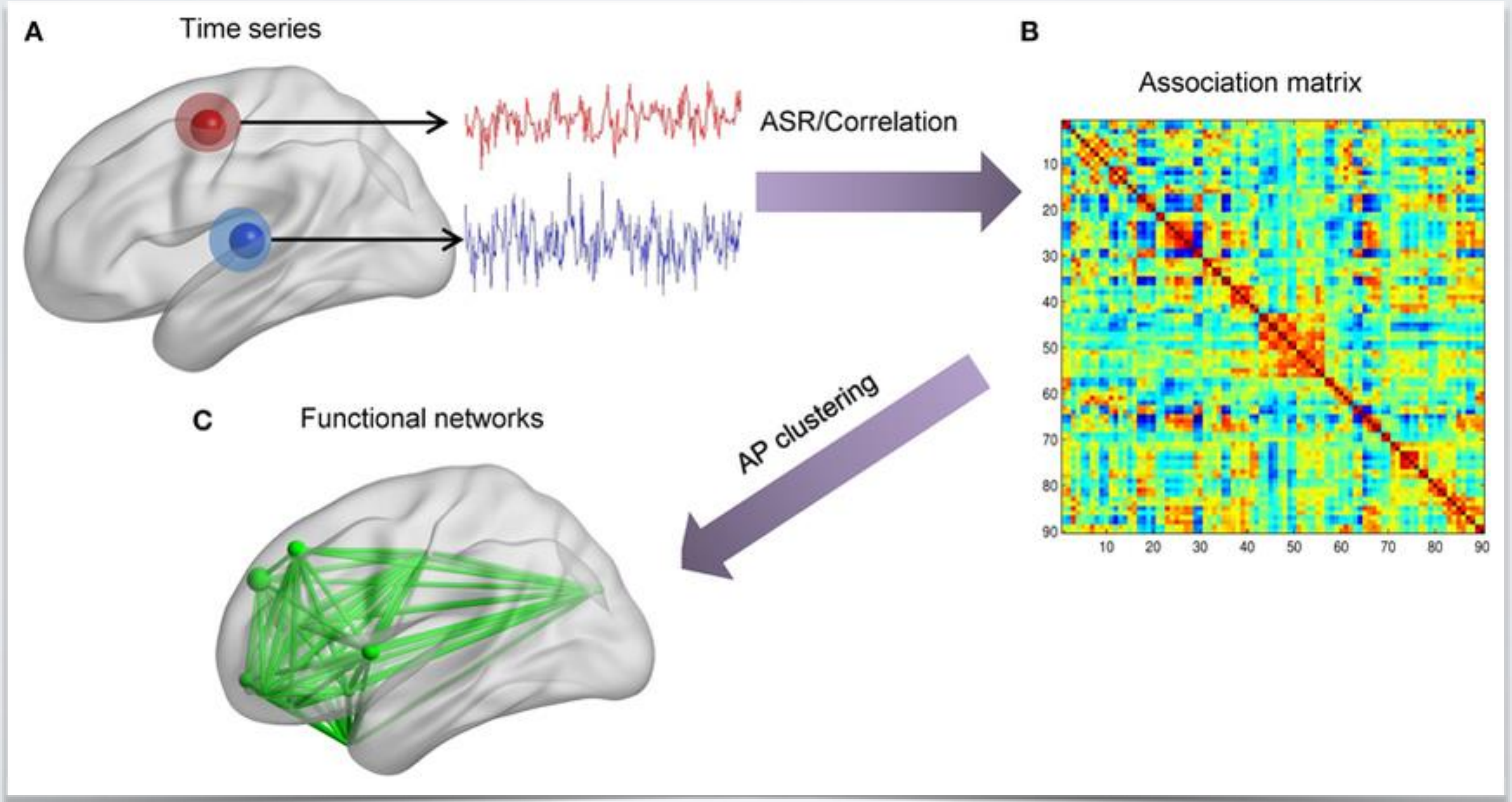
OBJECTS/VECTORS
TO
GRAPHS

GRAPH<->VECTORS

- Graph Embedding: Graph->Vectors
- What about Vectors->Graphs
  - Simple approach: Correlation matrix
  - =>Represent the relations between features in a dataset
    - 1)Compute the correlation between all variables(spearman/Pearson)
    - 2)Keep only correlations above a threshold (alternative: x% strongest)
    - 3)Correlation values can be represented as weights

ITEM-ITEM GRAPH

- Typical application case: Brain signal analysis
  - Distance is computed as signal correlation on fMRI, i.e., regional brain activity
  - => Time series to graph

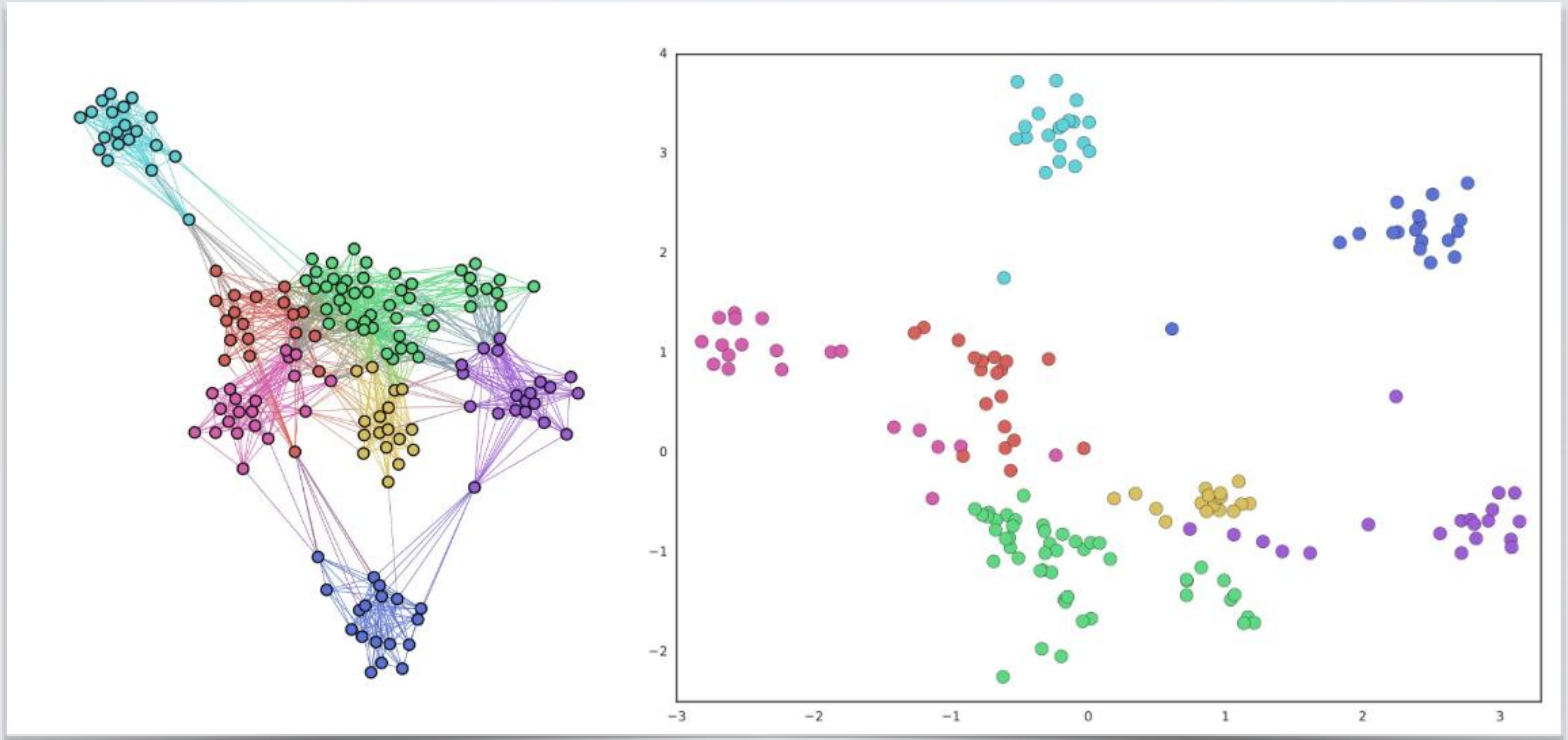


ITEM-ITEM GRAPH

- We can use graphs as an alternative to dimensionality reduction for visualization
  - PCA / tSNE: project items in 2D, close items are similar
    - Some impossibilities, e.g., multiple semantics for words (“palm”: part of the hand, tree)
  - Networks can also be viewed in 2D and preserve the similarity information

- Approach:
  - 1)Compute the distance between elements
    - Euclidean
    - Cosine
  - 2)Keep as an edge values above a threshold

ITEM-ITEM GRAPH



Comparison PCA-graph representation

FEATURE-FEATURE GRAPH

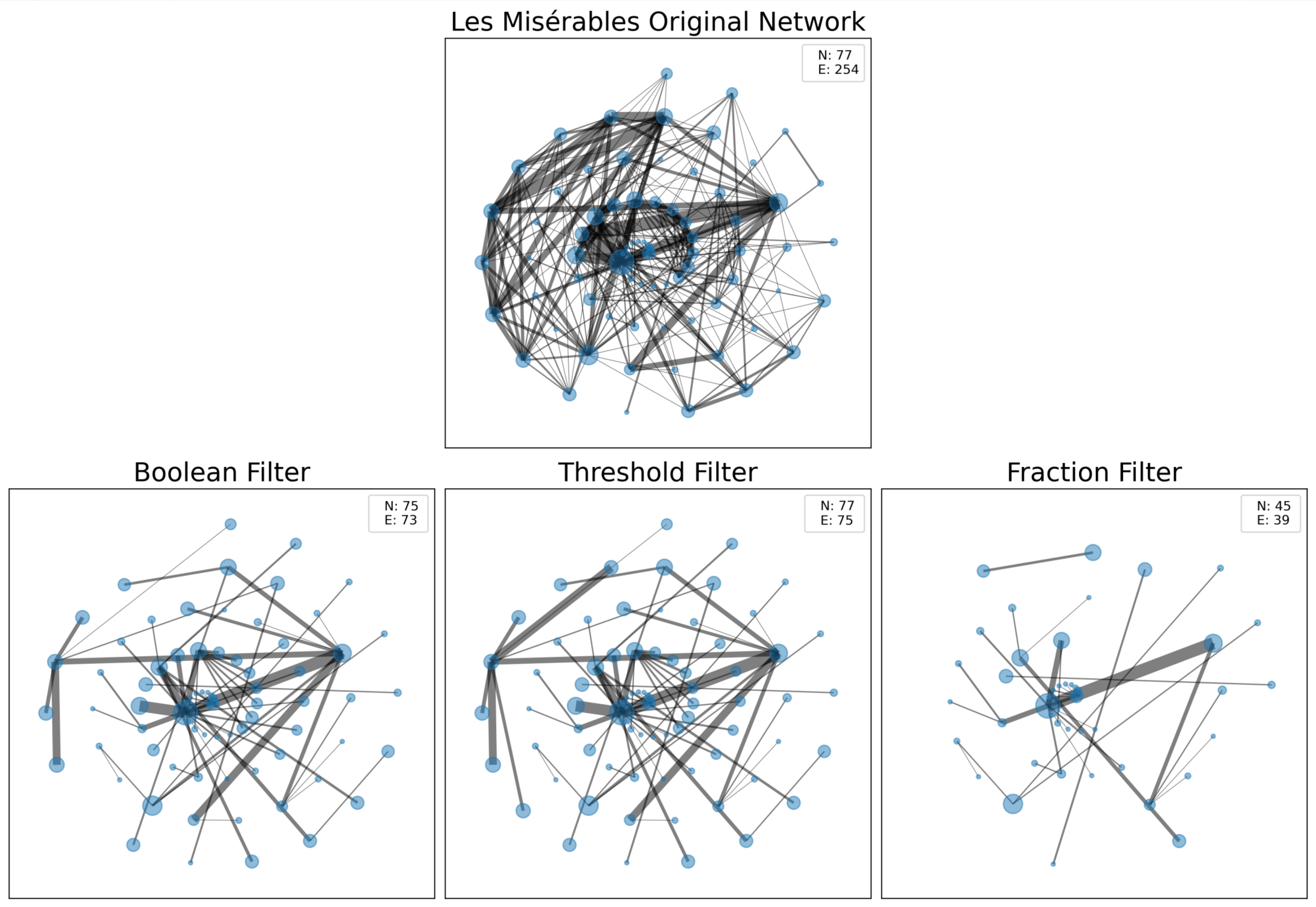
- Imagine an apartment dataset with variables surface, # rooms, etc.
  - Item-tem: apartment as nodes, links represent similar apartments
  - Feature-feature: each feature is a node, edges represent relations/correlation
- Useful in particular when many variables
  - Recommendation
  - Biological data
  - etc.

BACKBONE EXTRACTION

- In some cases, the network created might be too dense to be analyzed properly
  - Too low threshold: everything is connected
  - Too high: disconnected graph, most elements removed

- A solution is to use Backbone extraction
  - Methods that retain only the most important edges, based on different principles
  - e.g., https://pypi.org/project/netbone/

BACKBONE EXTRACTION



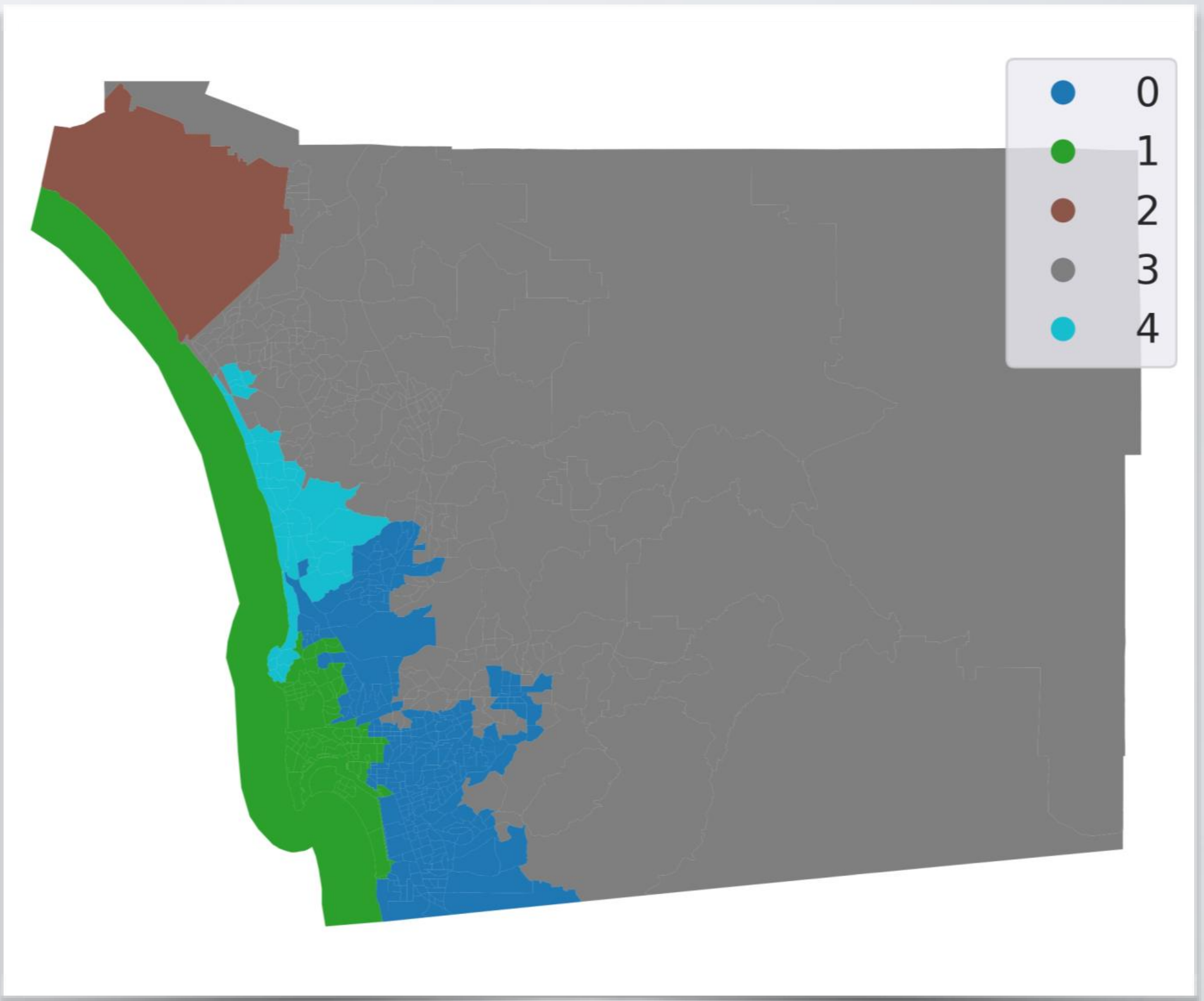
SPATIAL DATA ANALYSIS

REGIONALIZATION

- Clustering: finding groups of similar observations
- If the data has a spatial structure, we might want the clusters to be contiguous in space
- =>Add a spatial constraint

REGIONALIZATION

e.g., vote, weather…

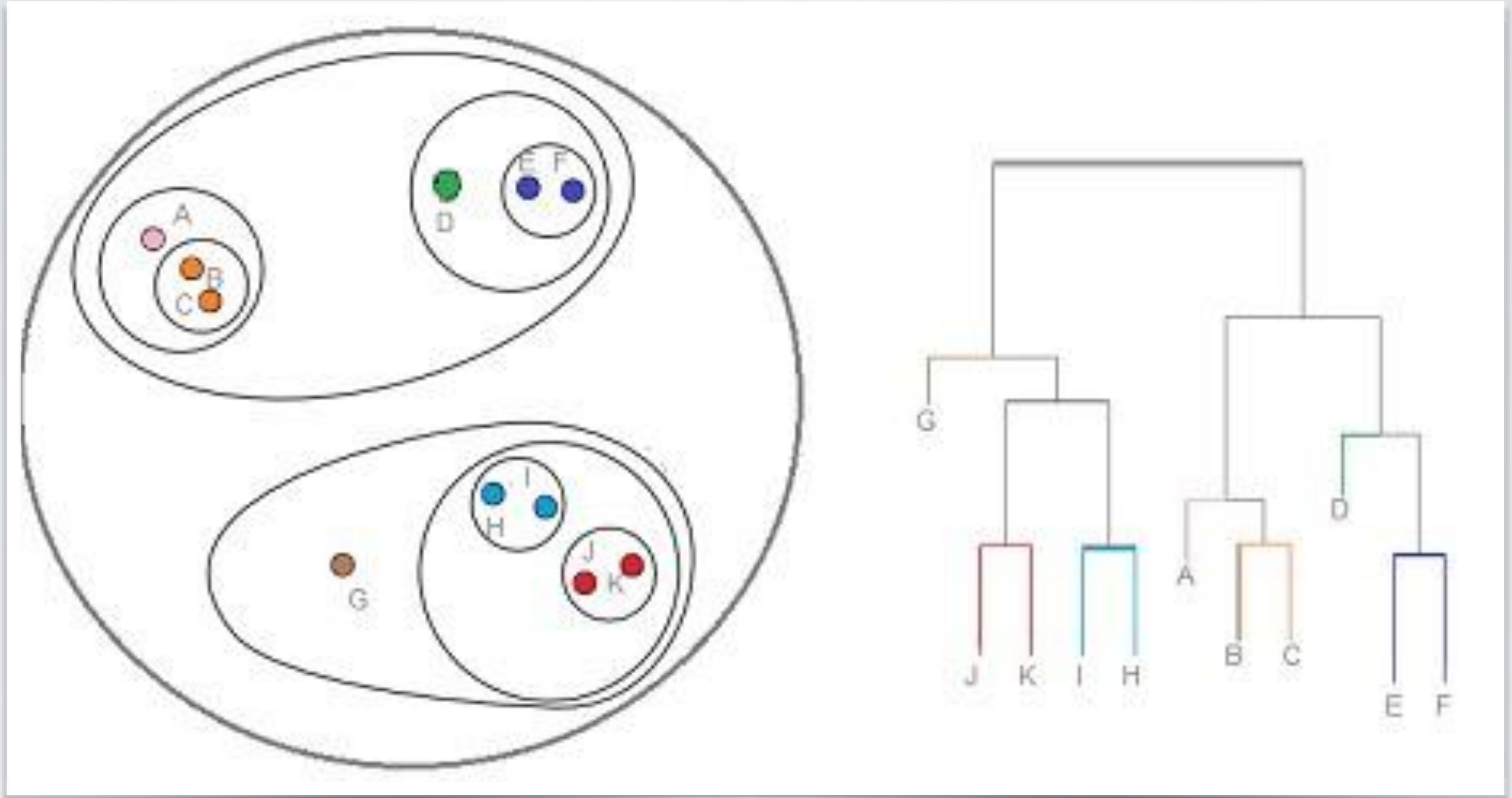


https://geographicdata.science/book/notebooks/10_clustering_and_regionalization.html

AGGLOMERATIVE CLUSTERING

- Agglomerative clustering is (yet another) clustering method
- Define a notion of distance between two sets of points, e.g.
  - Minimal distance between sets elements
  - Average distance between elements
  - …
- Start with each item in its own cluster
- **While** nb_cluster >1
  - Merge the two closest cluster

DENDROGRAM



https://www.statisticshowto.com/hierarchical-clustering/

CLUSTER DISTANCES

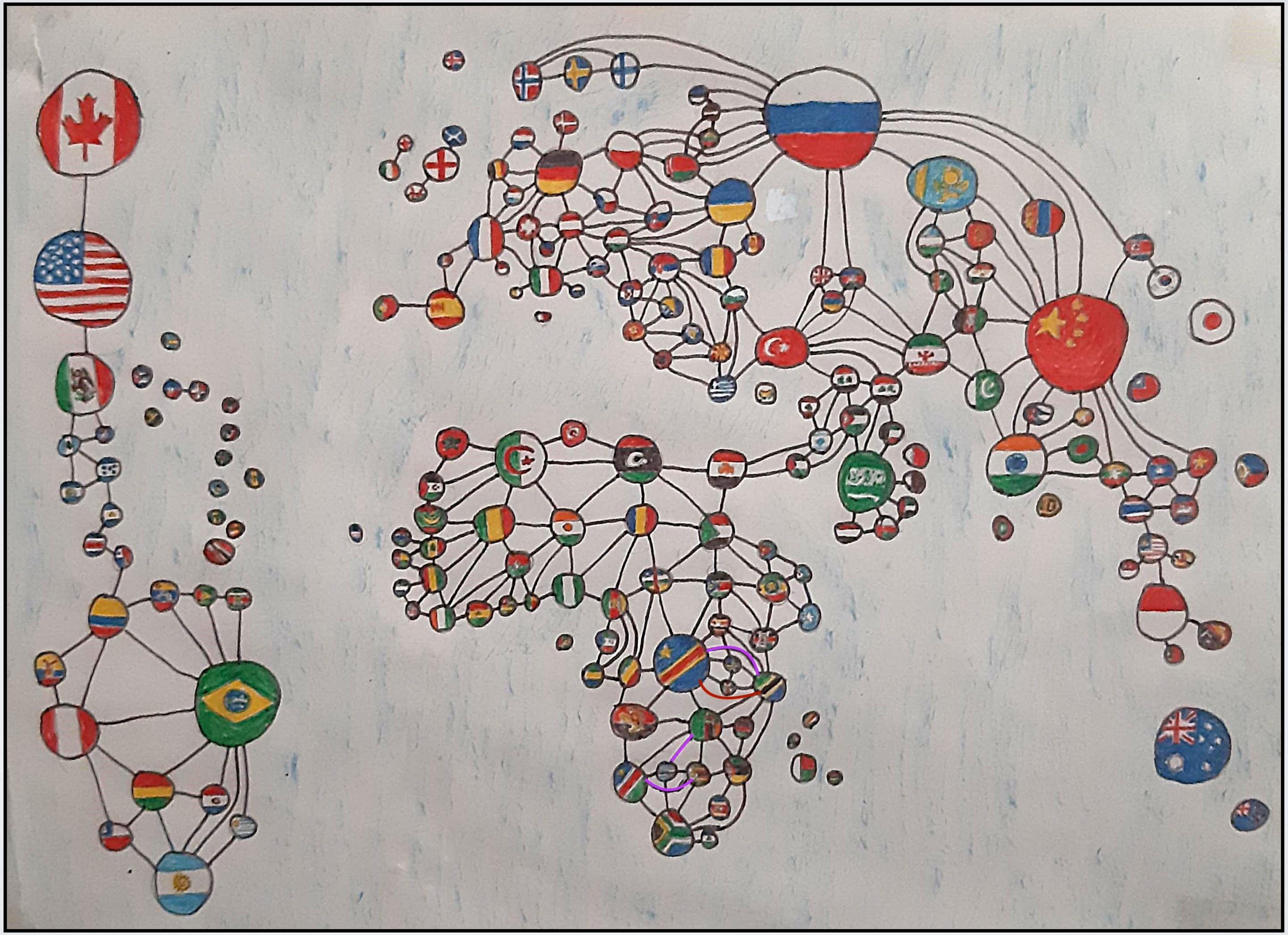
- Choose a distance function
  - Euclidean distance
  - Cosine distance
  - …

- Choose a cluster distance strategy
  - **single** uses the minimum of the distances between all observations of the two sets.
  - **complete** or 'maximum' linkage uses the maximum distances between all observations of the two sets.
  - **average** uses the average of the distances of each observation of the two sets.
  - **ward** minimizes the variance of the clusters being merged. (Within-Cluster Sum of Squares)
    - $\Delta WCSS = WCSSnew - (WCSSC_1 + WCSSC_2)$
    - Similar objective than k-means, but more greedy

REGIONALIZATION

- To discover spatial clusters, we want to allow merging only **spatially contiguous** clusters
- Solution: Connectivity matrix
  - A **graph** describing what element is a **neighbor** of another element.
  - Can merge only clusters with at least one edge between clusters

REGIONALIZATION



REGIONALIZATION

- Connectivity matrix (Binary graph)
  - Contiguity:
    - Contact between surface
    - Distance < threshold
  - KNN (K-nearest-neighbors)

- Spatial Weights Matrix (Weighted graph)
  - Put weights on edges
    - Inverse of the distance
    - Inverse of the squared distance…
  - Row normalized: sum of weights of neihgbors=1

REGIONALIZATION

- Other methods
  - K-means with constraints
    - Multiple variants
  - DBSCAN: principle of a graph with threshold…

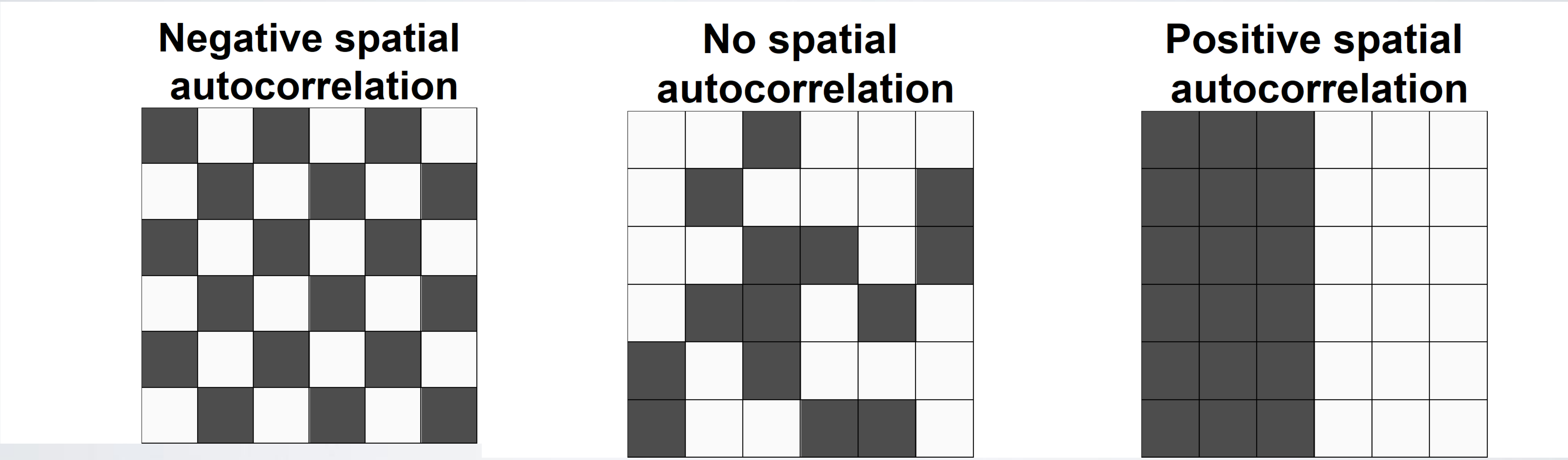
SPATIAL AUTOCORRELATION

Global

INTUITION

- Suppose you have attributes on observations
  - Binary (vote FOR/AGAINST, has covid cases or not, etc.)
  - Multi-label (candidate, type of apartments, etc.)
- Are those points distributed randomly/independently?
  - Or is there a correlation between the position of a point and the ones close to it
- Correlation between a variable and itself in space
  - =>Spatial autocorrelation

INTUITION


Negative spatial autocorrelation
No spatial autocorrelation
Positive spatial autocorrelation

INTUITION

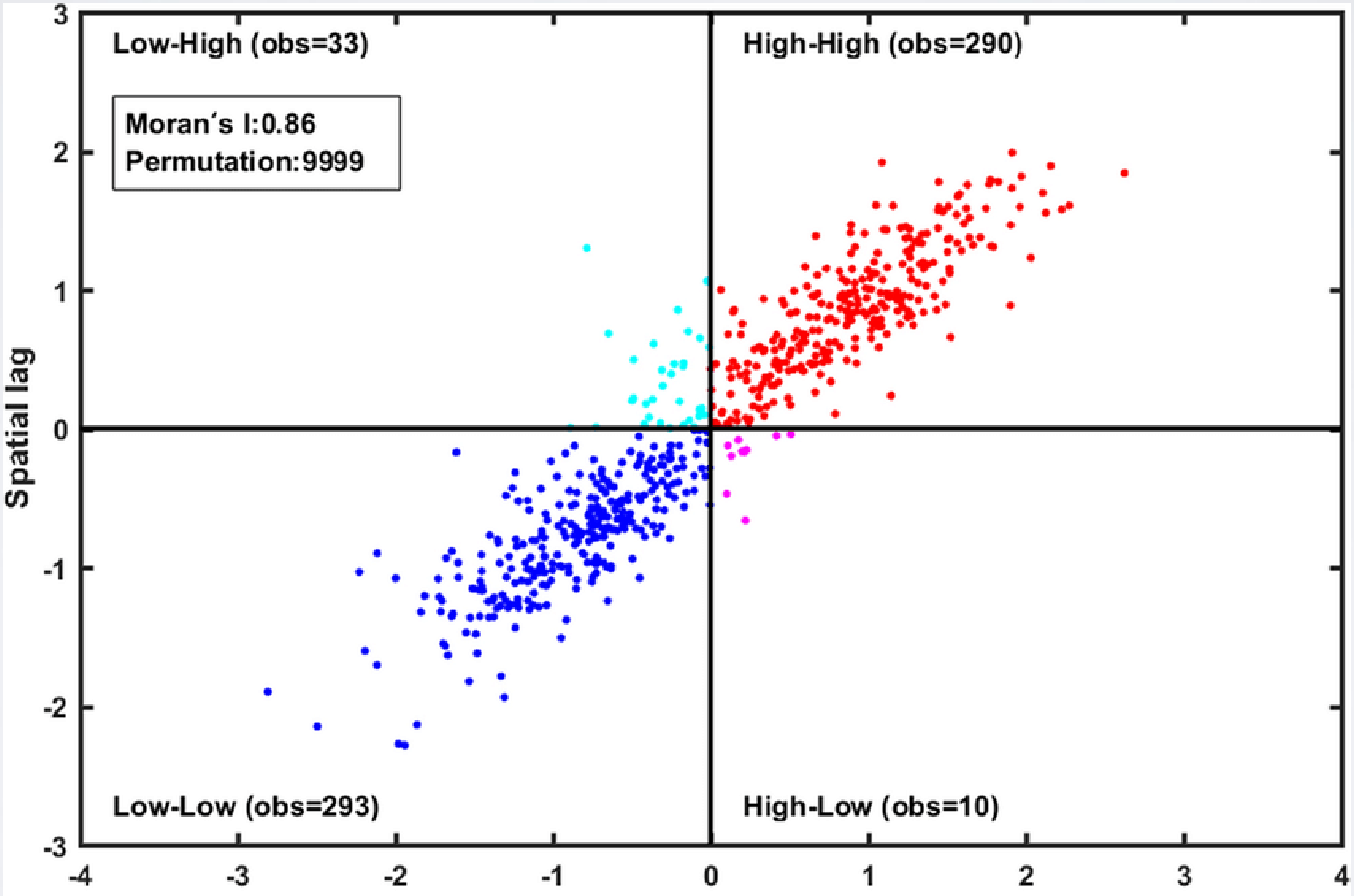
- Using a Spatial Weights Matrix
  - w_{ij}: weight of edge (i, j)

- Spatial lag: $y_i^{sl} = \sum_j w_{ij} y_j$
  - With y_j the variable of interest

- Weighted average of neighbors

MORAN'S PLOT

Plot relation between standardized values



Moran's I is the slope of a linear regression on this plot

LINEAR SPATIAL AUTOCORRELATION

- Compute Pearson's linear correlation between
  - Value for observation x
  - Spatial lag for observation x
- In practice, people rather use Moran's I
  - Generalization to take into account:
    - Different # of neighbors
    - Different weights
  - Slope of linear regression on Moran's plot

MORAN'S I

$$I = \frac{n}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j w_{ij} z_i z_j}{\sum_i z_i^2}$$

- w_{ij}: weight of edge (i, j)
- z_i: value at i, standardized
- n: nb. of observations

SPATIAL AUTOCORRELATION

Local

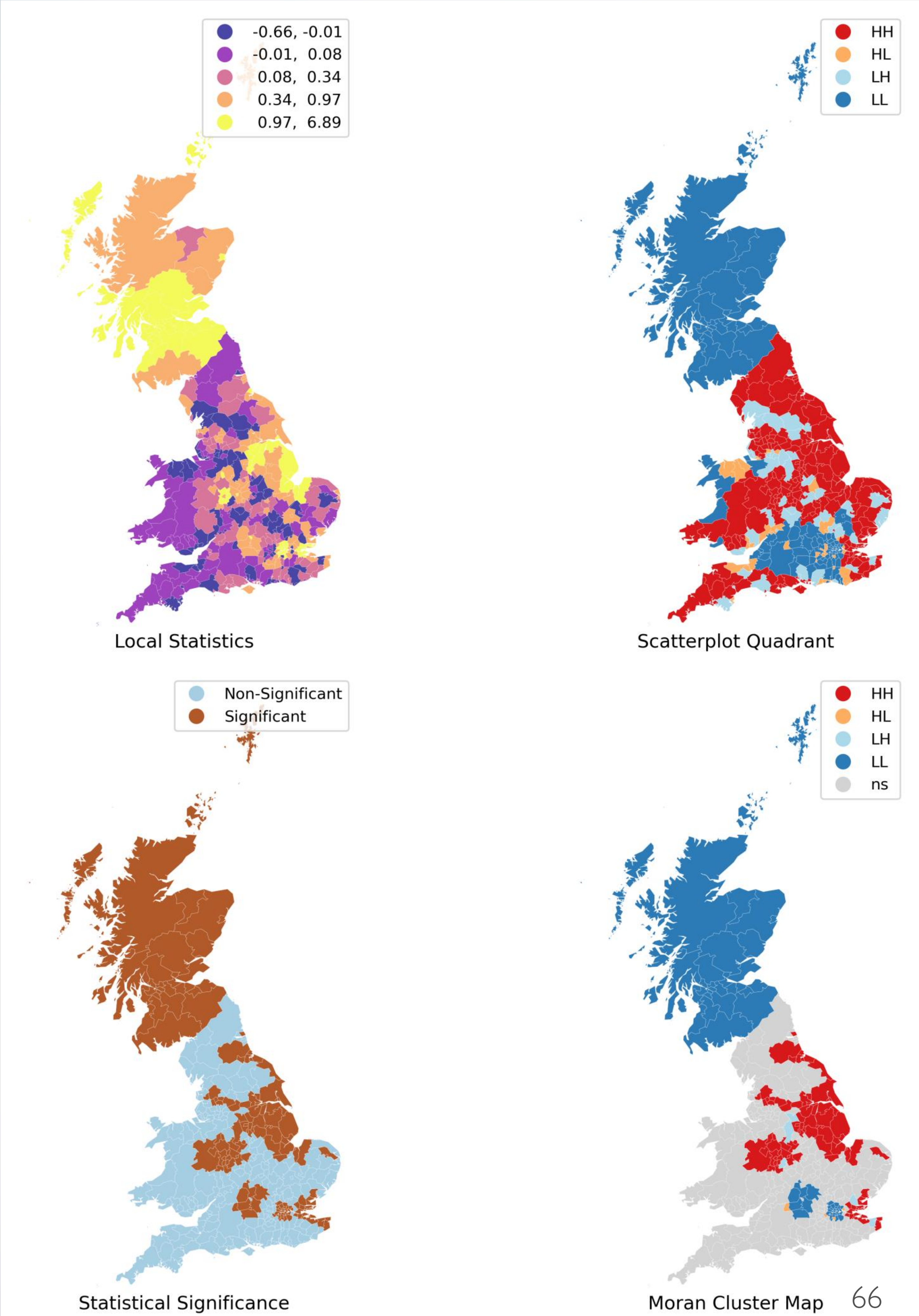
INTUITION

- Single scores are often misleading
- We can look at the details:
  - Where are positive/negative autocorrelations?
  - Where is the autocorrelation significant?
- Introduce LISA
  - Local Indicators of Spatial Association

LISA

- 1)Compute significance: Moran's Ii
  - $I_i = \frac{z_i}{m_2} \sum_j w_{ij} z_j; m_2 = \frac{\sum_i z_i^2}{n}$
    - m_2: variance of the variable of interest
    - z_i: standardized value
  - Positive value: positive spatial correlation at this point
  - Negative value: negative spatial correlation at this point
  - 0 or close to 0: no significant spatial autocorrelation

- Threshold on this value to decide significance



Brexit vote example (Support for Brexit)

HH: Hot spots
LL: Cold spots
LH: doughnuts
HL: diamonds in the rough

https://geographicdata.science/book/notebooks/07_local_autocorrelation.html