Multi-Dimensional Scaling (MDS)

ML2: AI Concepts and Algorithms (SS2025)

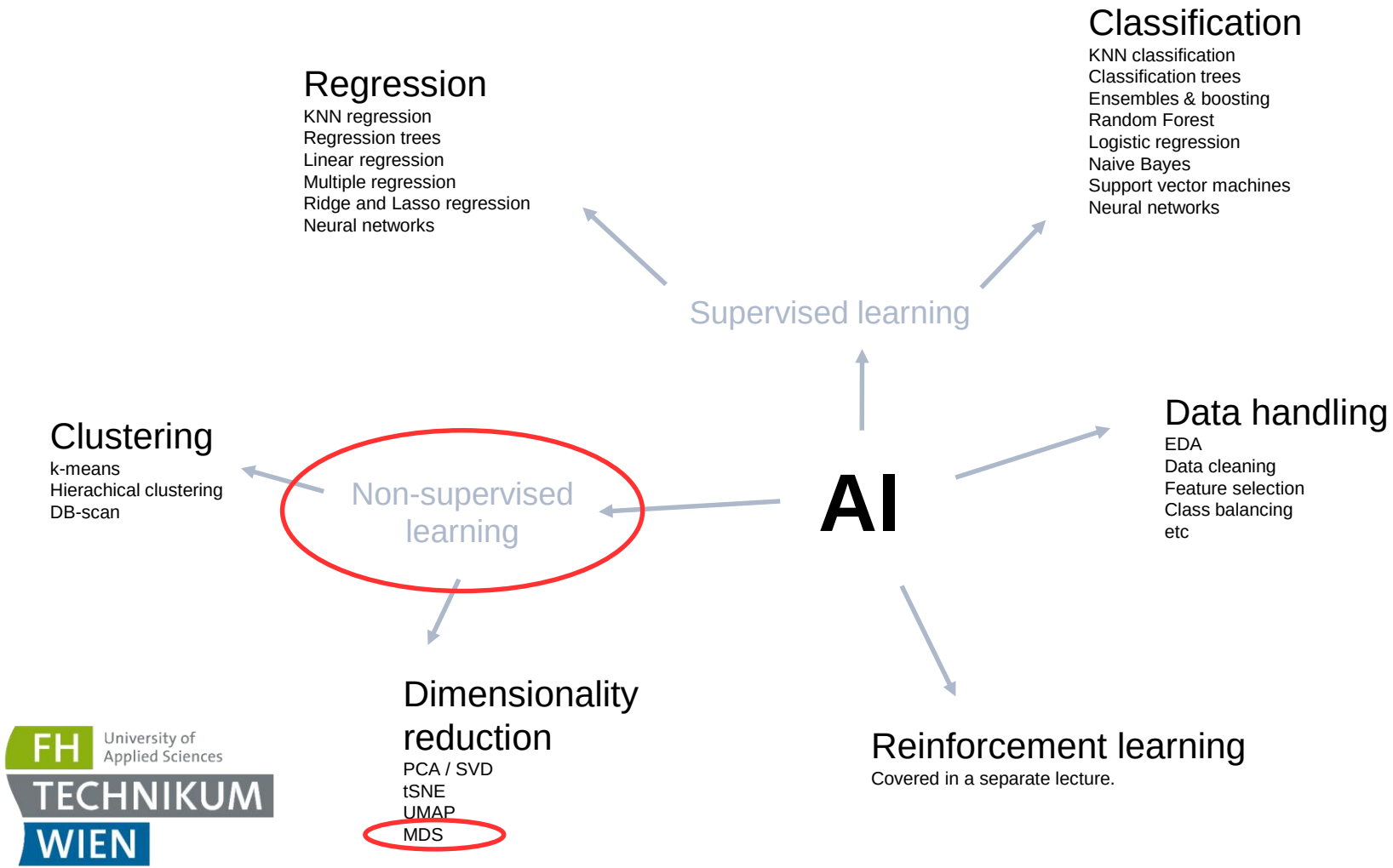
Faculty of Computer Science and Applied Mathematics

University of Applied Sciences Technikum Wien

FH University of Applied Sciences TECHNIKUM WIEN

Lecturer: Rosana de Oliveira Gomes

Authors: S. Rezagholi, R.O. Gomes



FH University of Applied Sciences TECHNIKUM WIEN

Dimensionality Reduction Recap

Dimensionality curse: challenges and complexities that arise when working with high-dimensional data, where the number of features or variables is significantly large

High dimensional systems: e.g. genomics, environmental science, NLP, customer segmentation.

Dimensionality reduction transforms data from high-dimensional space into a low-dimensional space while preserving as much of the dataset's original information as possible.

How to convert a multi-dimensional dataset into a small dimension visualization?

Image Source: Biostatsquid

Common algorithms

- Principle Component Analysis
- T-SNE
- UMAP

FH University of Applied Sciences TECHNIKUM WIEN

Recap

PCA (Principal Component Analysis): Captures **linear** relationships among variables while maximizing **variance.** It is fast and interpretable, but limited to linear patterns.

t-SNE (t-Distributed Stochastic Neighbor Embedding): Focuses on **local** structures and is great for **cluster visualization.** It captures nonlinear relationships, but is computationally intensive and less interpretable.

UMAP (Uniform Manifold Approximation and Projection): Balances **local** and **global** structures. It is very fast, flexible, and effective for large, nonlinear data. UMAP maintains more global structure than t-SNE.

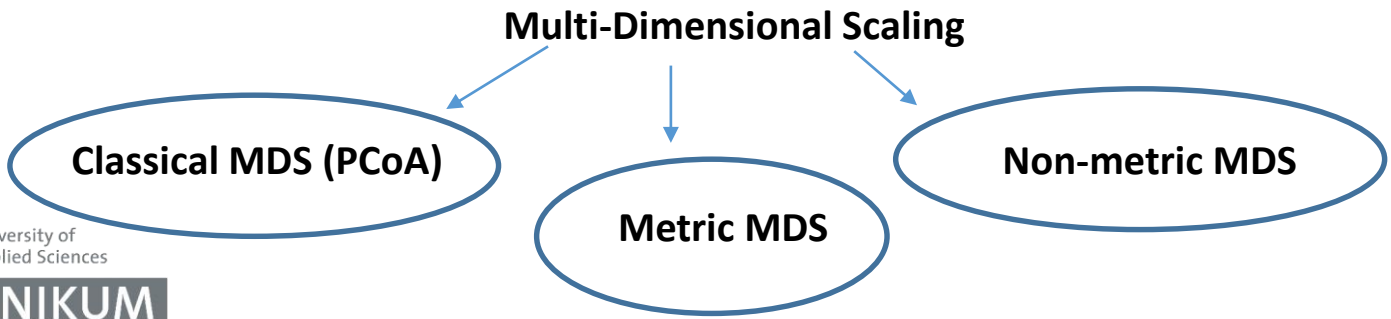
Multi-Dimensional Scaling

Goal: Given pairwise dissimilarities,
reconstruct a map that preserves distances.

MDS is a class of algorithms designed to reach an optimal low-dimensional configuration.

Visual representation of distances or dissimilarities between datapoints.

Use Cases: relationship among datapoints (e.g. social sciences, epidemiology, etc).



Multi-Dimensional Scaling

We are given n datapoints $\{x_1, ..., x_n\}$ with corresponding distance matrix D.
We want to find a set of n points $\{\hat{x}_1, ..., \hat{x}_n\}$ in $\mathbb{R}^p$ such that

$$\text{dist}(\hat{x}_i, \hat{x}_j) \approx \text{dist}(x_i, x_j)$$

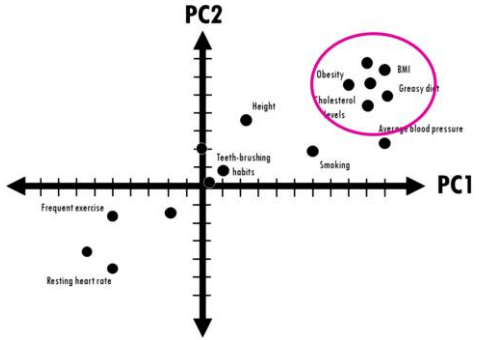
holds as best as it can.

We want to embed the datapoints into *lower-dimensional* space with minimal distortion of the distances!

Classical Multi-Dimensional Scaling (cMDS)

Given a dataset with respective features.

Principal Component Analysis converts correlations into a 2D graph.



cMDS: converts distances among the samples into a 2D graph. Assumes Euclidean distance matrix.

cMDS is also known as **Principal Coordinate Analysis (PCoA)**

	Cell1	Cell2
Gene1	3	0.25
Gene2	2.9	0.8

Euclidean distance

$$\sqrt{(3 - 0.25)^2 + (2.9 - 0.8)^2}$$

$(2.9 - 0.8) = 2.1$

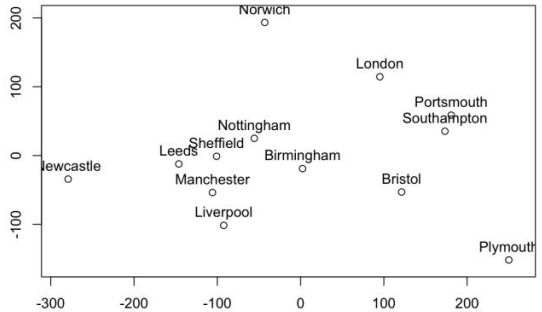
$(3 - 0.25) = 2.75$

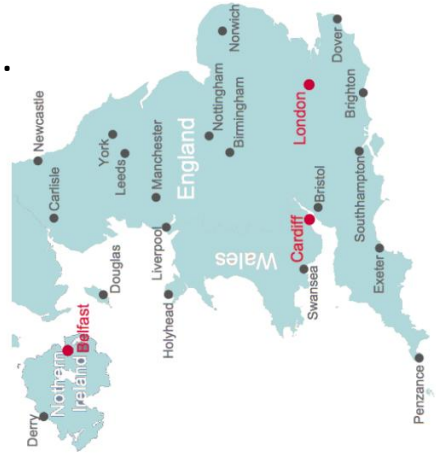
FH University of Applied Sciences TECHNIKUM WIEN

cMDS Example

Minimizing the Euclidean distance is the same as maximizing the correlation. PCA and PCoA are the same if considering linear distances.

E.g. MDS creates map of distance across cities.
Distances relationships are real. Does not know orientations.





FH University of Applied Sciences TECHNIKUM WIEN

Classical Multidimensional Scaling

Algorithm Steps: from Standardized data

1. Distance Matrix

Standardized data

	DBP	SBP	BMI	Chol
Patient 1	-0.917	-1.086	-0.955	0.000
Patient 2	-0.262	-0.926	-0.665	1.222
Patient 3	1.703	0.990	1.504	-1.324
Patient 4	-0.426	0.032	-0.376	-0.560
Patient 5	-0.098	0.990	0.492	0.662

Distance matrix

	1	2	3	4	5
1	0	1.426	4.356	1.463	2.741
2	1.426	0	4.327	2.051	2.314
3	4.356	4.327	0	3.093	2.866
4	1.463	2.051	3.093	0	1.809
5	2.741	2.314	2.866	1.809	0

$$d_{1,3} = \sqrt{((-0.917) - 1.703)^2 + ((-1.086) - 0.990)^2 + ((-0.955) - 1.504)^2 + (0 - (-1.324))^2} = 4.356$$

2. Double Centering Matrix (B)

Computed from squared distance D.

The centered matrix is computed from D so that the mean of each variable is zero. The double centering matrix is:

$$B = -\frac{1}{2} C D^{(2)} C$$

Interpretation:

- **Transforms distance information into geometry**.
- Ensures that the new points are.

centered at the origin (mean 0)

- Allows to work with **distances only**

to derive coordinates.

$$B = \begin{bmatrix} 2.932 & 1.881 & -4.073 & 0.715 & -1.455 \\ 1.881 & 2.864 & -3.983 & -0.353 & -0.409 \\ -4.073 & -3.983 & 7.897 & -0.517 & 0.676 \\ 0.715 & -0.353 & -0.517 & 0.638 & -0.482 \\ -1.455 & -0.409 & 0.676 & -0.482 & 1.670 \end{bmatrix}$$

FH University of Applied Sciences TECHNIKUM WIEN

Classical Multidimensional Scaling

3. Calculate Eigenvectors and Eigenvalues

Consider only the first 2 eigenvectors and eigenvalues which will be used for creating the lower dimensional space.

$$E_m = \begin{bmatrix} -0.441 & 0.452 & 0.042 & -0.632 & 0.447 \\ -0.416 & -0.330 & -0.641 & 0.327 & 0.447 \\ 0.783 & 0.219 & -0.359 & -0.098 & 0.447 \\ -0.054 & 0.369 & 0.488 & 0.650 & 0.447 \\ 0.127 & -0.709 & 0.469 & -0.247 & 0.447 \end{bmatrix} \quad \lambda_1 = 12.447 \quad \lambda_2 = 2.448$$

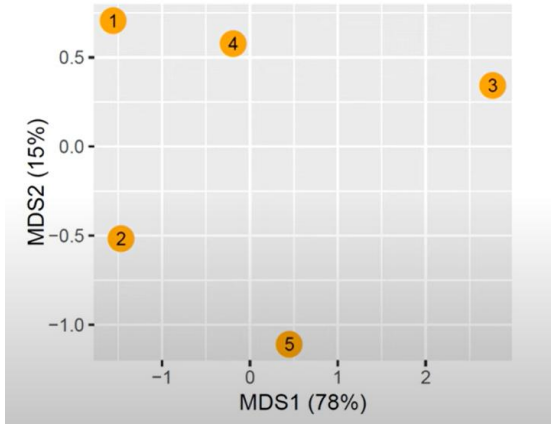
Represent the eigenvalues in a diagonal matrix:

$$\Lambda_m = \begin{bmatrix} 12.447 & 0 \\ 0 & 2.448 \end{bmatrix}$$

4. Calculate Scores

Compute the score matrix that represents the data points in the lower-dimension space:

$$X = E_m \Lambda_m^{1/2} = \begin{bmatrix} -1.556 & 0.706 \\ -1.466 & -0.517 \\ 2.764 & 0.343 \\ -0.190 & 0.577 \\ 0.448 & -1.110 \end{bmatrix}$$



FH University of Applied Sciences TECHNIKUM WIEN

Metric Multi-Dimensional Scaling

In mathematics, a distance function (that gives a distance between two objects) is also called *metric*, satisfying

$d(x, y) \geq 0,$ → Distances are positive

$d(x, y) = 0$ if and only if $x = y,$ → Distance to itself is zero

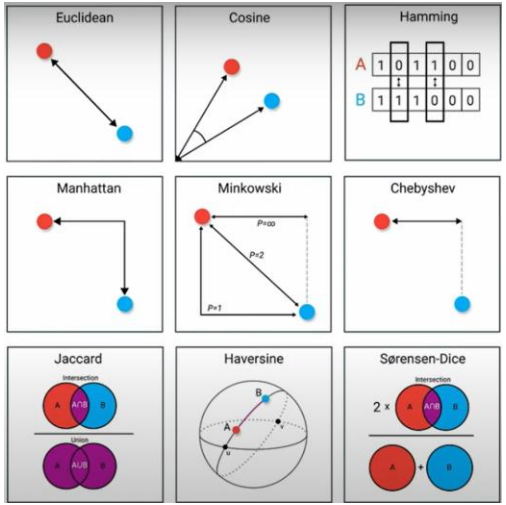
$d(x, y) = d(y, x),$ → Symmetry

$d(x, z) \leq d(x, y) + d(y, z).$ → Triangle Inequality: sum of the shorter distances between points in a triangle is larger than the longest distance.

If the last inequality does not hold, one often speaks of a ***dissimilarity*** instead of a metric or distance.

Metric Multi-Dimensional Scaling

Metric MDS is a more general method that finds a metric for any distance function instead of assuming an Euclidean distance.



classical MDS

seeks to find an optimal configuration $\mathbf{x}_i$ that gives $d_{ij} \approx \hat{d}_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ as close as possible.

Distance Scaling

- Relaxing $d_{ij} \approx \hat{d}_{ij}$ from cMDS by allowing

$$\hat{d}_{ij} \approx f(d_{ij}), \text{ for some monotone function } f.$$

Distance Scaling:

- **Metric MDS** if dissimilarities *dij* are quantitative.
- **Non-metric MDS** if dissimilarities *dij* are qualitative (e.g. ordinal).

Unlike cMDS, distance scaling is an optimization process minimizing *Stress function*, and is solved by iterative algorithms.

FH University of Applied Sciences TECHNIKUM WIEN

Metric Multi-Dimensional Scaling

Stress Function: **S**tandardized **R**esidual **S**um of **S**quares

$$\text{stress} = \mathcal{L}(\hat{d}_{ij}) = \left(\sum_{i<j} (\hat{d}_{ij} - f(d_{ij}))^2 \Big/ \sum d_{ij}^2 \right)^{\frac{1}{2}}$$

Normalization to make the expression scale-invariant.

Stress values are between 0 and 1:

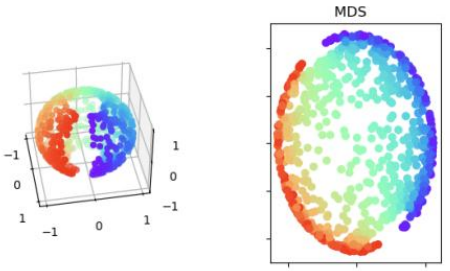
Stress	Goodness of fit
20%	poor
10%	fair
5%	good
2½%	excellent
0%	"perfect"

The function f can be taken to be a parametric monotonic function, such as $f(d_{ij}) = \alpha + \beta d_{ij}$.

Metric MDS minimizes the stress as a function of the parameters in f.

FH University of Applied Sciences TECHNIKUM WIEN

Metric MDS



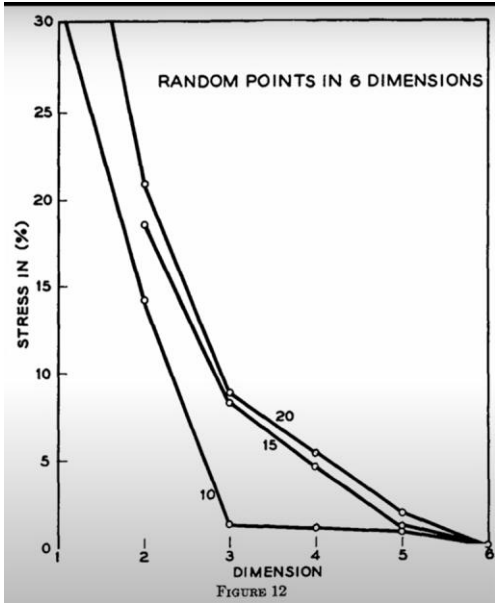
Algorithm Steps

1. **Initialization:** initialize points in random positions
2. **Compute Distances:** obtain distrance matrix for the configuratio
3. **Compute Loss:** Evaluate *Stress function*.
4. **Optimize:** Apply gradient descent to update/ minimize *stress*

Use (a variant of) gradient descent with an appropriate stopping condition.

Note:

- The number of dimensions influences the outcome.
- The parameters of the optimization procedure influence the outcome.
- The random initialization of points influences the outcome.



FH University of Applied Sciences TECHNIKUM WIEN

MDS iMPLEMENTATION

Parameters

dissimilarity: 'euclidean', 'precomputed', etc.

metric: Boolean (True = metric MDS, False = non-metric)

n_components: Target dimension (usually 2 or 3)

max_iter, **eps**: Convergence settings

random_state: For reproducibility

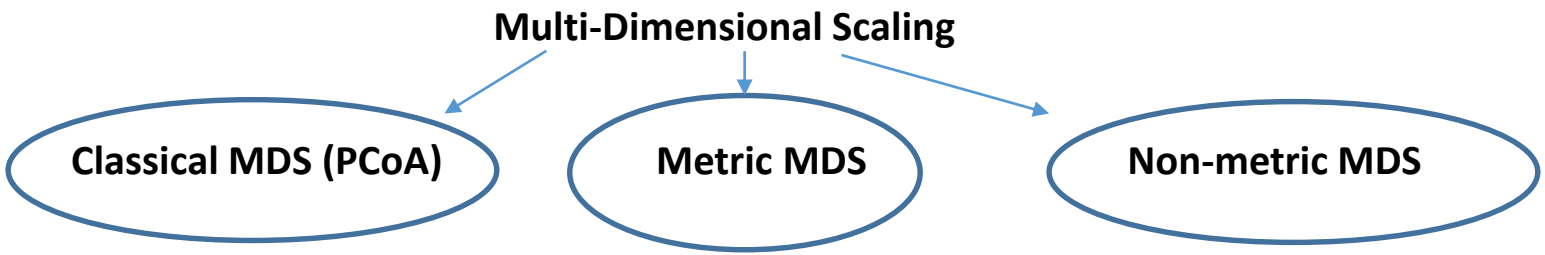
```
from sklearn.manifold import MDS
mds = MDS(n_components=2, metric=True, random_state=42)
X_transformed = mds.fit_transform(distance_matrix)
```

https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html

Non-Metric MDS

- In some applications (e.g. psychology or marketing) data is often me
  ordinal scales.
- Use case: sound/music/visual perception, facial expression, gene expression, etc
- For ordinal variables there is a version of MDS that is only sensitive to the relative ranks of ‘distances’.
- Ordinal MDS is available in Python via: *sklearn.manifold.MDS(metric=False)*.
- Ordinal MDS is an iterative numerical procedure combining steps of gra
  descent and isotonic regression.

When to use what?



•**Euclidean distances** between points.
•For a **fast and direct solution** (linear algebra only).
• The dissimilarity matrix is symmetric.

•**Non-Euclidean distances** (like cosine or correlation).
•**Noisy data**.
•**Rank orders** must be preserved.

•**Meaningful quantitative dissimilarities** (e.g. distances between cities).
•Dissimilarities are **non-Euclidean** but **metric**.
•Flexibility with distance types (e.g., correlation Mahalanobis).

•Dissimilarities are **rank-based or subjective**.
•**Large datasets** (slow optimization).

•**Subjective** (e.g., perceived similarity of songs, faces).
•**Ordinal relationships** are more impotant than exact distances.
•Dissimilarities are **not metric** (e.g., some psychological ratings).

• **Precise distances** must be preserved.
•Dissimilarities are **already Euclidean** (use classical instead).

FH University of Applied Sciences TECHNIKUM WIEN

Takeaway

- MDS is a form of nonlinear dimensionality reduction.
- PCA is related to classic MDS, since both use matrix decomposition, but they are not the same: PCA leads to linear dimensionality reduction.
- In certain disciplines it is common to apply (ordinal) MDS to dissimilarities obtained from correlation matrices.
- The concept of Multidimensional Scaling based on elementary mathematics *(Kruskal and Wish 1978)* allow for a more generalized method.

Next up: Support Vector Machines

Assignment: MDS

1 Explanation

- Explain metric multidimensional scaling.
- Explain ordinal MDS.

Prepare a presentation (slides)!

2 Experimentation

Choose an interesting high-dimensional dataset and compare the results of metric MDS and UMAP.

FH University of Applied Sciences TECHNIKUM WIEN