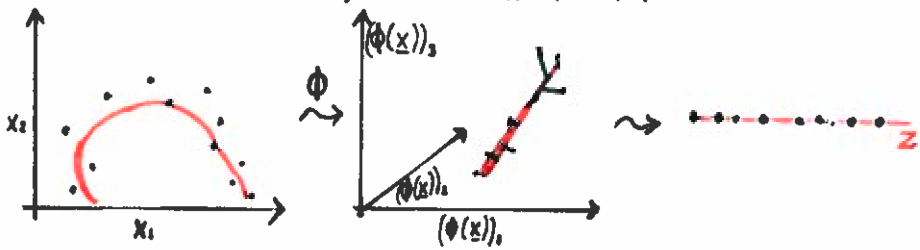
Dimensionality reduction

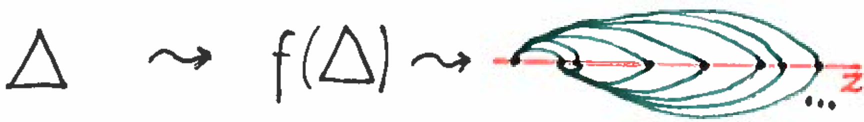
- Linear:
  - Principal Component Analysis

- Nonlinear:
  - Nonlinear PCA, kernel PCA



  - Metric MDS

  - Nonmetric MDS



- Classical MDS (= "Classical scaling")

  Precise numerical values taken into account

  Assumption: Δ directly represents Euclidean distances between <u>hypothetical</u> points $\underline{z}_1, \dots, \underline{z}_n \in \mathbb{R}^k$, k unknown

  $\rightsquigarrow$ Apply "PCA from distances" procedure (use Gram matrix $G = -\frac{1}{2} C_n \Delta C_n$, calculate Z)

But: If assumption violated:

some eigenvalues $\tilde{\lambda}_j$ can be negative

$\rightsquigarrow$ discard corresponding EV

Optimisation problem:

$$\boxed{\begin{aligned} Z &= \underset{M}{\text{minimise}} \left\| \left(-\tfrac{1}{2} C_n \Delta C_n\right) - M^T M \right\|_F \\ \text{s.t. } &\text{rank } (M^T M) = k \end{aligned}}$$

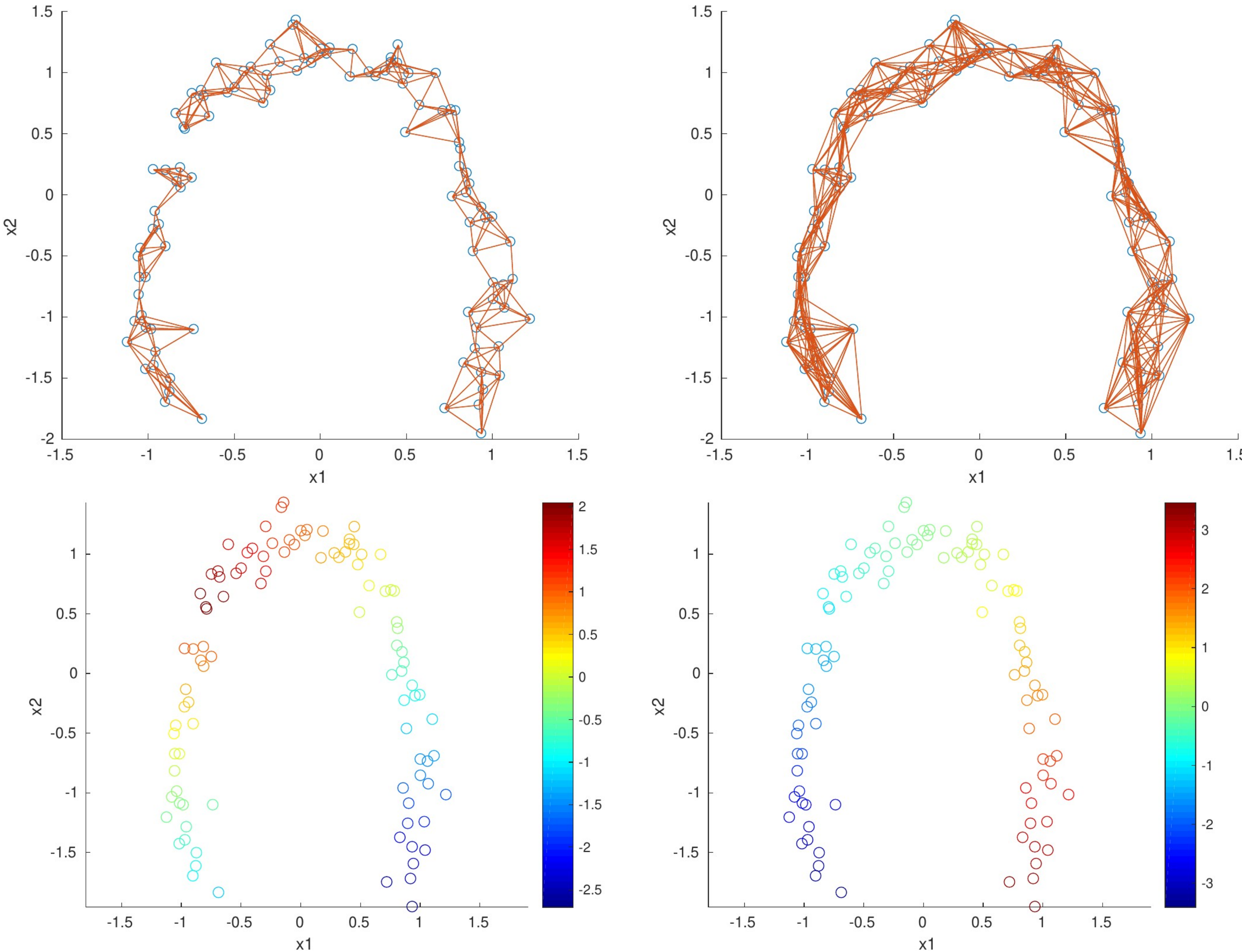
Isomap

Classical MDS with particular Δ:

For $\underline{x}_i, \underline{x}_j \in \mathbb{R}^d$:

δ_{ij} = shortest distance when travelling from one neighbouring data point to the next, where neighbourhood defined by m-nearest neighbours (m-NN)

Isomap



(a) 5 neighbours

(b) 10 neighbours

Uniform Manifold Approximation and Projection (UMAP)

Nearest neighbour - based graph dimensionality reduction method (like Isomap)

For Isomap: used graph to transform data to Δ

For UMAP: calculate a graph for data $\tilde{X}$ and for each potential Z; minimise graph distance

Graph construction: for given $Y = (y_1, \cdots y_n)$

m-NN graph for Y; add weights to each edge:

$$w_i^{(Y)}(y_i, y_j) = \exp\left(-\frac{\|y_i - y_j\| - \rho_i}{\sigma_i}\right)$$

where σ_i : diameter of neighbourhood

ρ_i : distance to nearest neighbour

Weight: "Probability that edge exists"

"Make data uniform on manifold"

$\rightsquigarrow$ two weights per edge - can be asymmetric

$\rightsquigarrow$ set
$$w^{(Y)}(y_i, y_j) = w_i^{(Y)}(y_i, y_j) + w_j^{(Y)}(y_i, y_j) - w_i^{(Y)}(y_i, y_j)\, w_j^{(Y)}(y_i, y_j)$$

"Probability that at least one of the edges exists" $\rightsquigarrow$ Weight matrix $(W^{(Y)})_{ij} = w^{(Y)}(y_i, y_j)$

- Calculate $W^{(X)}$ for given data X
- Calculate $W^{(Z)}$ for potential Z
- Calculate cross-entropy between $W^{(\tilde{X})}$ and $W^{(Z)}$:

$$C\left(W^{(\tilde{X})}, W^{(Z)}\right) = \sum_{i<j} w_{ij}^{(\tilde{X})} \log\left(\frac{w_{ij}^{(\tilde{X})}}{w_{ij}^{(Z)}}\right) + \left(1 - w_{ij}^{(\tilde{X})}\right) \log\left(\frac{1 - w_{ij}^{(\tilde{X})}}{1 - w_{ij}^{(Z)}}\right)$$

Optimisation problem:

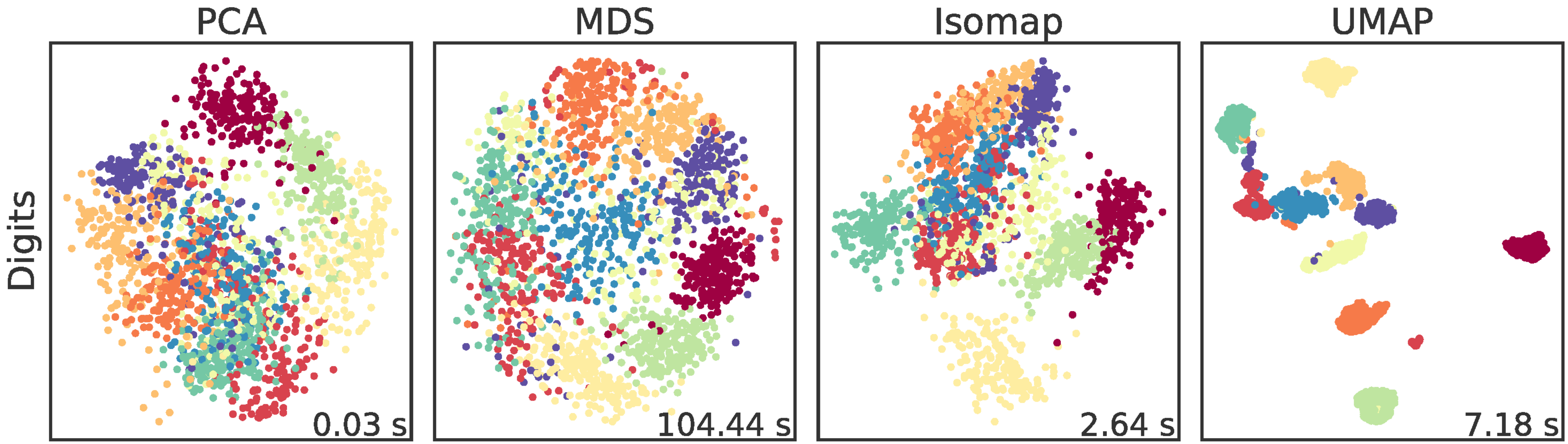
$$\boxed{\underset{Z}{\text{minimise}}\ C\left(W^{(\tilde{X})}, W^{(Z)}\right)}$$

Optimise via gradient descent

Recent method:

McInnes, Healy & Melville 2018

Comparison on Handwritten Digits



Adapted from https://umap-learn.readthedocs.io/