Clustering and Manifold Learning

Ludwig Krippahl

Clustering and Manifold Learning

Summary

- Fuzzy sets and clustering
- Fuzzy c-means
- Manifold learning
- tSNE and Isomap
- External indexes: Rand index
- Assignment 2

Fuzzy sets

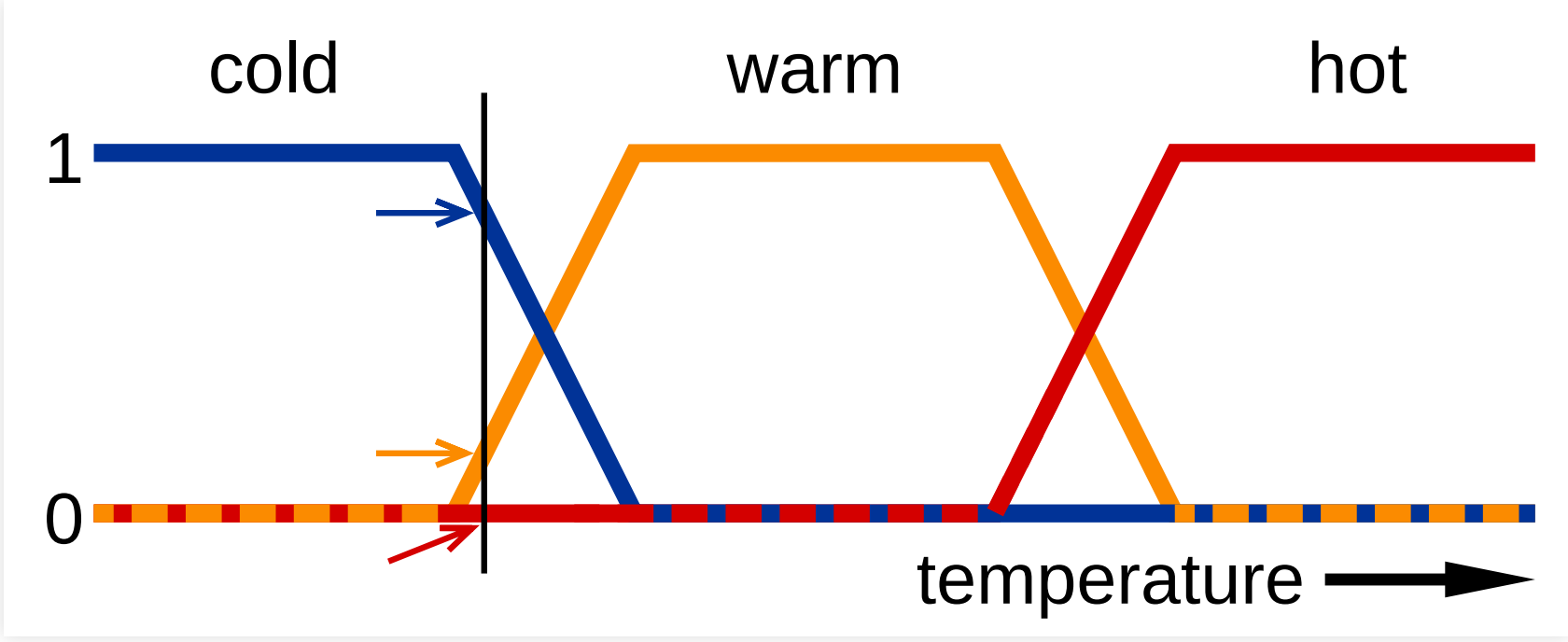
Fuzzy sets

- In conventional set theory, elements either belong or don't belong to a set
- In fuzzy set theory, each element has a $u_S(\mathbf{x}) \in [0, 1]$ indicating the degree of membership to set S
- More formally, a fuzzy set S is a set of ordered pairs

$$S = \{(\mathbf{x}, u_S(\mathbf{x}))|\mathbf{x} \in X\}$$

Fuzzy sets

- Fuzzy sets allow modelling uncertainty
- Linguistic and conceptual uncertainty (old, large, important, ...)



Wikimedia, CC BY-SA 3.0 fullofstars

Fuzzy sets

- Fuzzy sets allow modelling uncertainty
- Linguistic and conceptual uncertainty (old, large, important, ...)
- Informational uncertainty (credit, security risks, ...)
- Note: fuzzy membership is not probability. It's a measure of similarity to some imprecise properties

Fuzzy c-partition

- $\mathbf{U}(\mathbf{X})$ is a fuzzy c-partition of X if:

$$0 \leq u_k(\mathbf{x}_n) \leq 1 \qquad \forall k, n$$

$$\sum_{k=1}^{c} u_k(\mathbf{x}_n) = 1 \qquad \forall n$$

$$0 \leq \sum_{n=1}^{N} u_k(\mathbf{x}_n) \leq N \qquad \forall k$$

Fuzzy c-Means

Fuzzy c-Means

- From set X of N unlabelled data points
- Return the $c \times N$ membership matrix with $u_k(\mathbf{x}_n)$, defining a fuzzy c-partition of X
- Return the $\{C_1, \ldots, C_c\}$ centroids
- Minimizing the squared error function:

$$J_m(X, C) = \sum_{k=1}^{c} \sum_{n=1}^{N} u_k(\mathbf{x}_n)^m \|\mathbf{x}_n - \mathbf{c}_k\|^2 \qquad m \geq 1$$

Fuzzy c-Means

- Minimizing the squared error function:

$$J_m(X, C) = \sum_{k=1}^{c} \sum_{n=1}^{N} u_k(\mathbf{x}_n)^m \|\mathbf{x}_n - \mathbf{c}_k\|^2 \qquad m \geq 1$$

- Subject to:

$$\sum_{k=1}^{c} u_k(\mathbf{x}_n) = 1 \qquad \forall n$$

- and where m is the degree of fuzzification ; tipically, $m = 2$

Fuzzy c-Means

- The derivatives w.r.t. $u_k(\mathbf{x}_n)$ are zero at:

$$u_k(\mathbf{x}_n) = \frac{\left(\frac{1}{\|\mathbf{x}_n - \mathbf{c}_k\|^2}\right)^{\frac{2}{m-1}}}{\sum_{j=1}^{c}\left(\frac{1}{\|\mathbf{x}_n - \mathbf{c}_j\|^2}\right)^{\frac{2}{m-1}}}$$

- The derivatives w.r.t. c_k are zero at:

$$c_k = \frac{\sum_{n=1}^{N} u_k(\mathbf{x}_n)^m \mathbf{x}_n}{\sum_{n=1}^{N} u_k(\mathbf{x}_n)^m}$$

- That is, each centroid c_k is the weighted mean of the example vectors using the membership values.

Fuzzy c-Means algorithm

- Random initial centroids $\{C_1, \ldots, C_c\}$
- Compute u:

$$u_k(\mathbf{x}_n) = \frac{\left(\frac{1}{\|\mathbf{x}_n - \mathbf{c}_k\|^2}\right)^{\frac{2}{m-1}}}{\sum_{j=1}^{c}\left(\frac{1}{\|\mathbf{x}_n - \mathbf{c}_j\|^2}\right)^{\frac{2}{m-1}}}$$

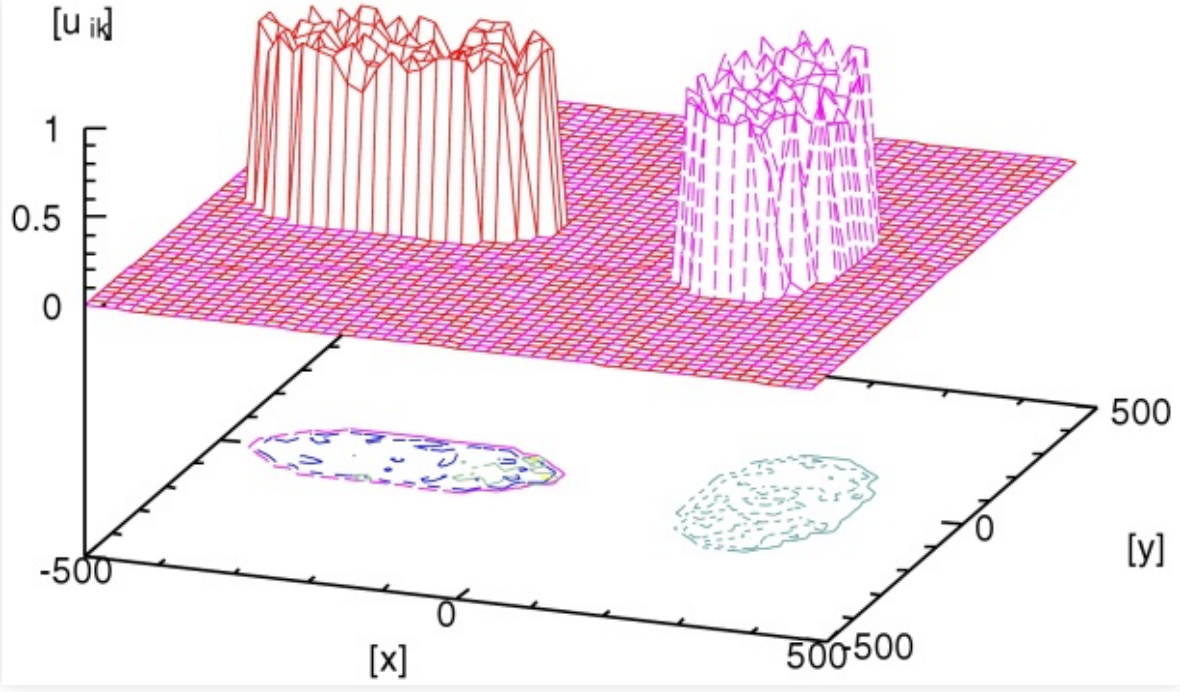
- Recompute c_k:

$$c_k = \frac{\sum_{n=1}^{N} u_k(\mathbf{x}_n)^m \mathbf{x}_n}{\sum_{n=1}^{N} u_k(\mathbf{x}_n)^m}$$

- Repeat until t iterations or change $< \epsilon$

Fuzzy c-Means algorithm

- The result is a clustering similar to k-means, but with membership values:



Source: "Simulated Annealing - Advances, Applications and Hybridizations" Ed. Marcos de Sales Guerra Tsuzuki, CC BY 3.0

Fuzzy c-Means algorithm

Defuzzification

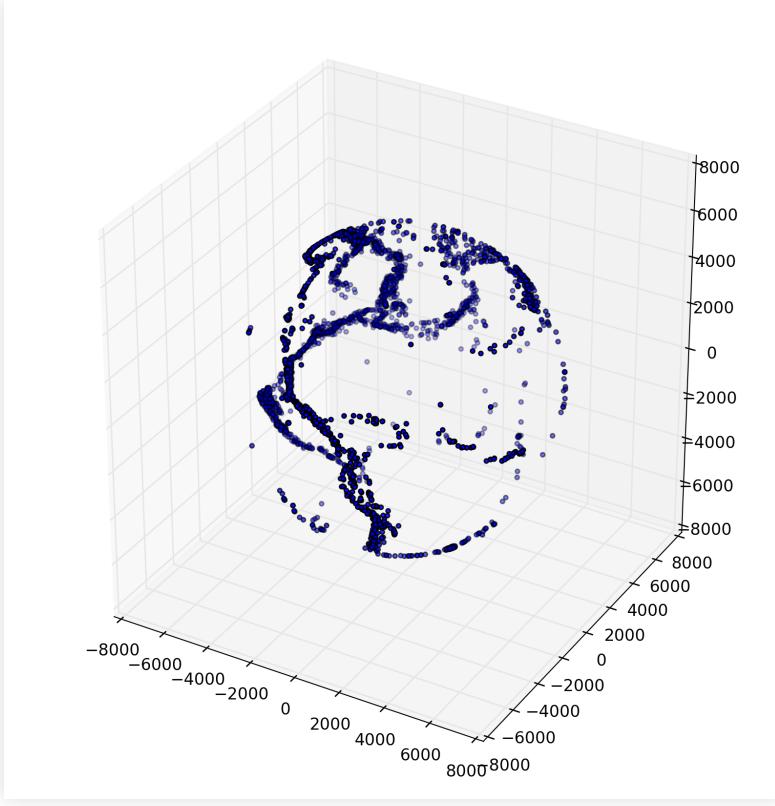
- If we want a crisp clustering, we'll need to convert the fuzzy membership function into $\{0, 1\}$

- Maximum membership
- Nearest centroid

Manifold Learning

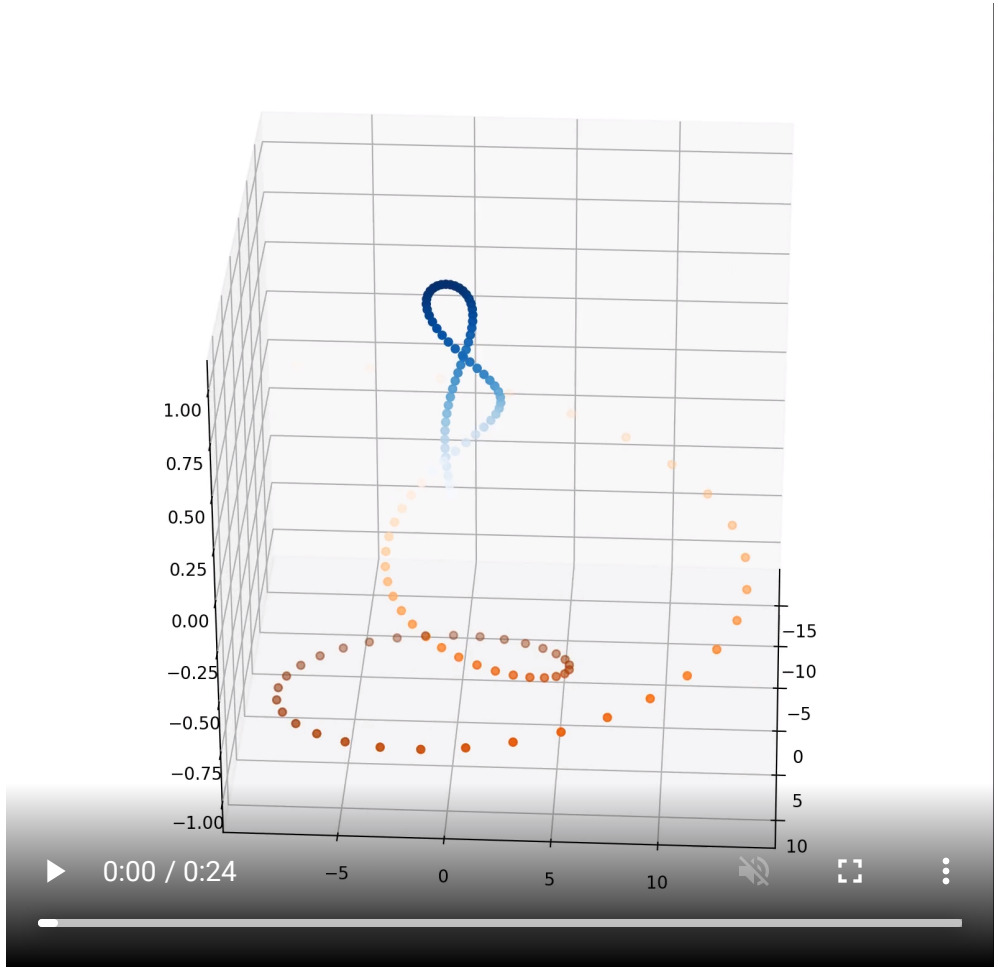
Manifold

- A set of points such that the neighbourhood of each is homomorphic to an euclidean space
  - Example: the surface of a sphere



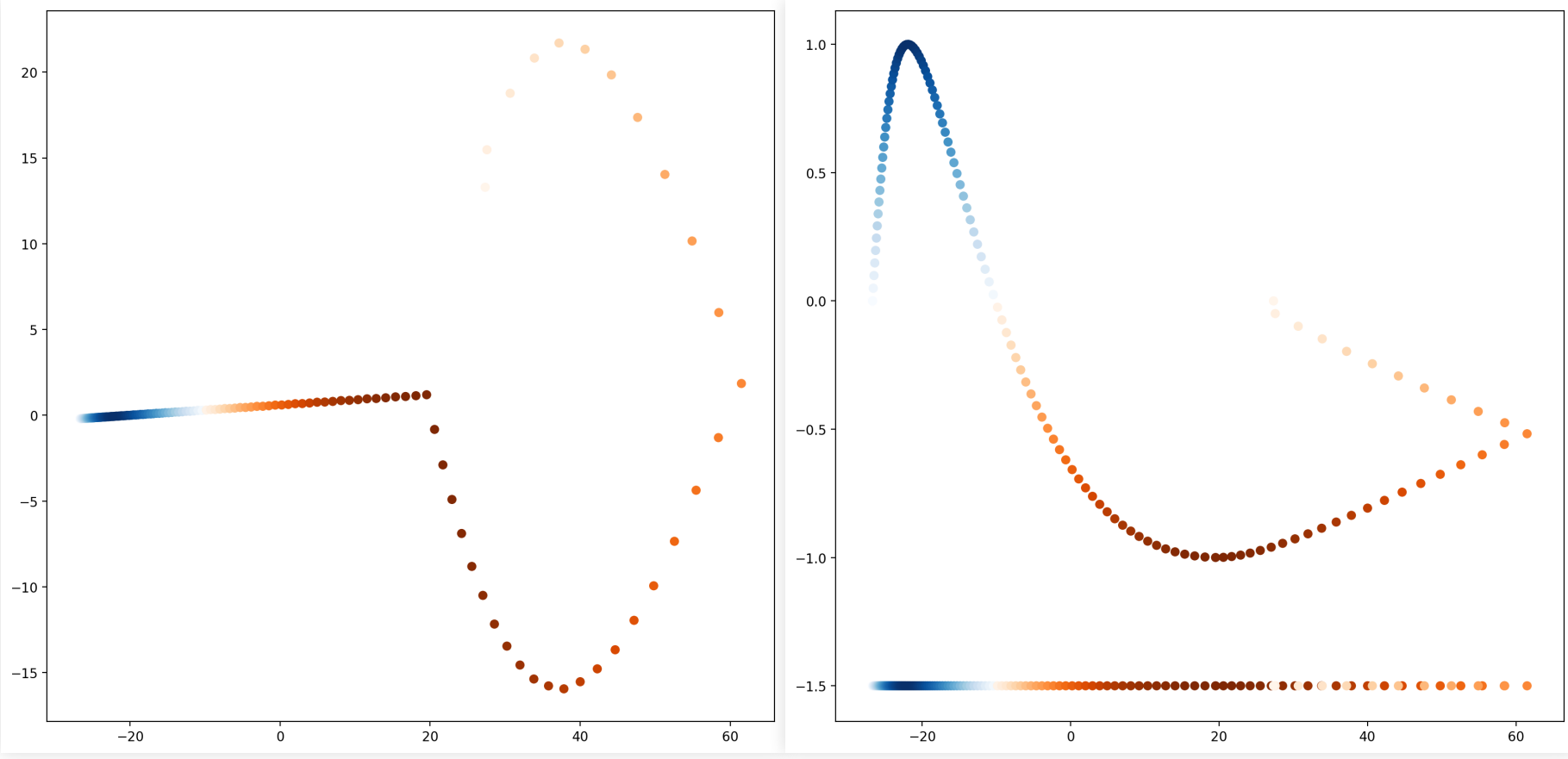
Manifold Learning

- Data may cover a lower dimension manifold of the space



Manifold Learning

- Learn lower dimension embeddings of data manifold



t-distributed stochastic neighbor embedding

- Probability $p_{j|i}$ of point x_i choosing x_j as neighbour with Gaussian distribution:

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{i \neq j} exp(-||x_i - x_j||^2/2\sigma_i^2)} \qquad (p_{i|i} = 0)$$

- From here, joint probability $p_{ji} = (p_{j|i} + p_{i|j})/2$
- Joint probability of the low-dimensional counterparts y_i and y_j uses a t-Student distribution with 1 degree of freedom:

$$q_{j|i} = \frac{(1 + ||y_i - y_j||)^{-1}}{\sum_{i \neq j}(1 + ||y_i - y_j||)^{-1}} \qquad (q_{i|i} = 0)$$

t-distributed stochastic neighbor embedding

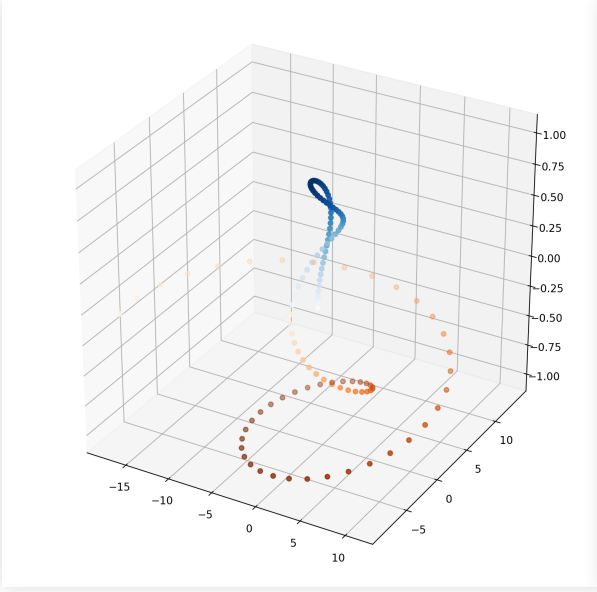
- Compute the values of $\mathbf{y}$ that minimize the Kullback–Leibler divergence of q_{ij} with respect to p_{ij}:

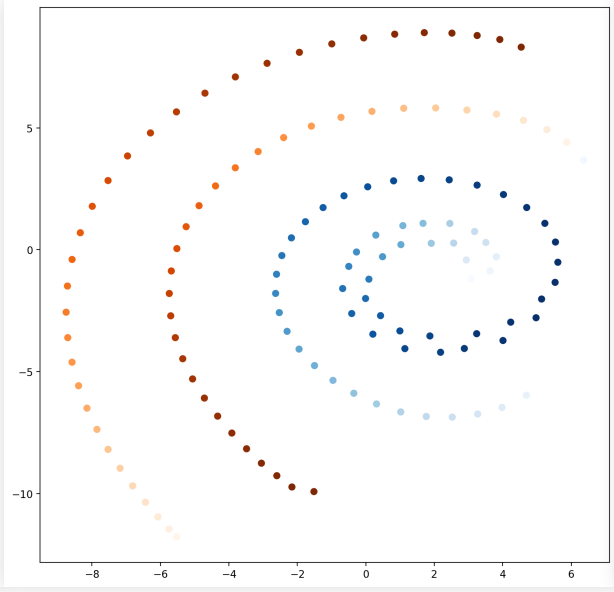
$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

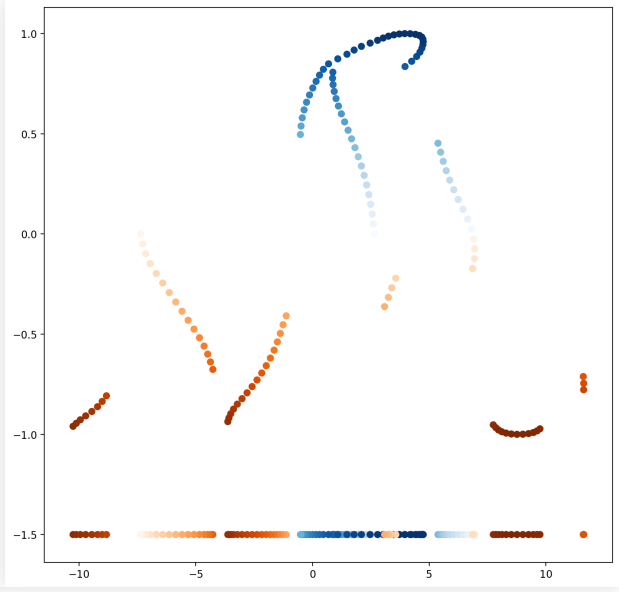
- This makes the neighbourhood distribution of $\mathbf{y}$ similar to $\mathbf{x}$

t-distributed stochastic neighbor embedding

- From 3D to 2D and 1D





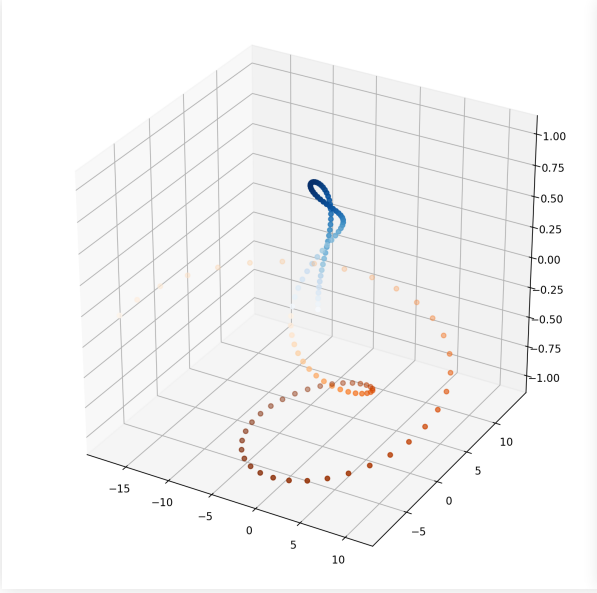


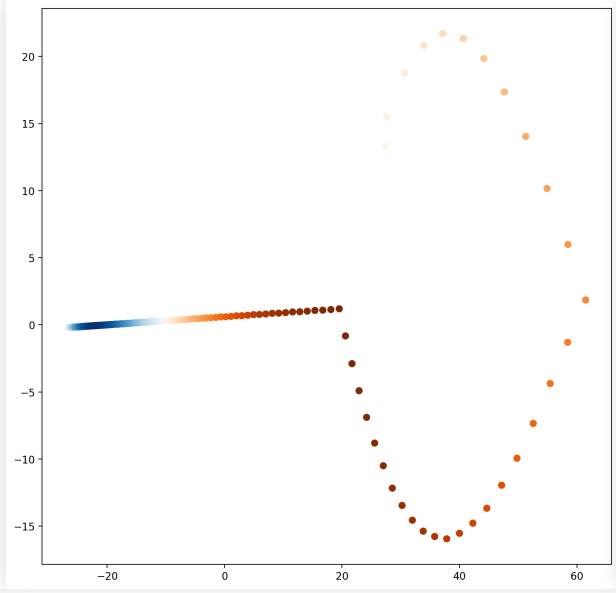
Isomap algorithm

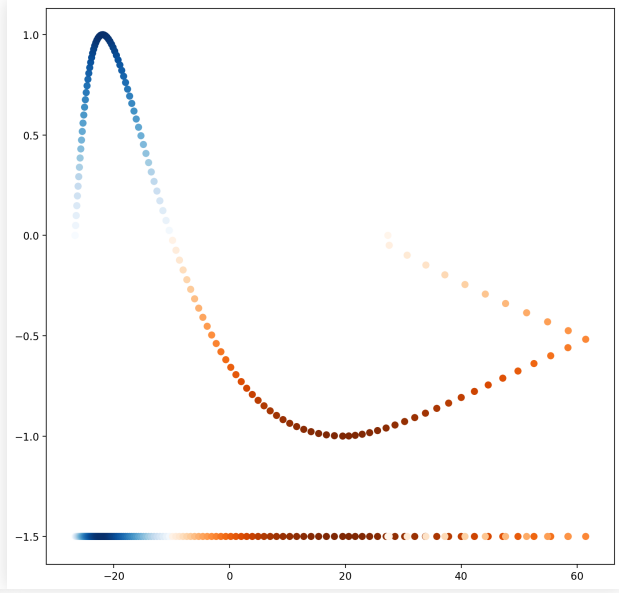
- For each point, list neighbours (e.g. K-nearest neighbours)
- Create a graph connecting each point to its neighbours
  - Edge is the euclidean distance in the original space
- Compute pairwise distances between points from paths in the neighbourhood graphs
  - Euclidean distance but only along the neighbourhoods
  - Thus, distance along the manifold
- Finally, compute low-dimensional embedding that respects distances
  - Multidimensional scaling: compute coordinates that approximate distances.

Isomap

- From 3D to 2D and 1D

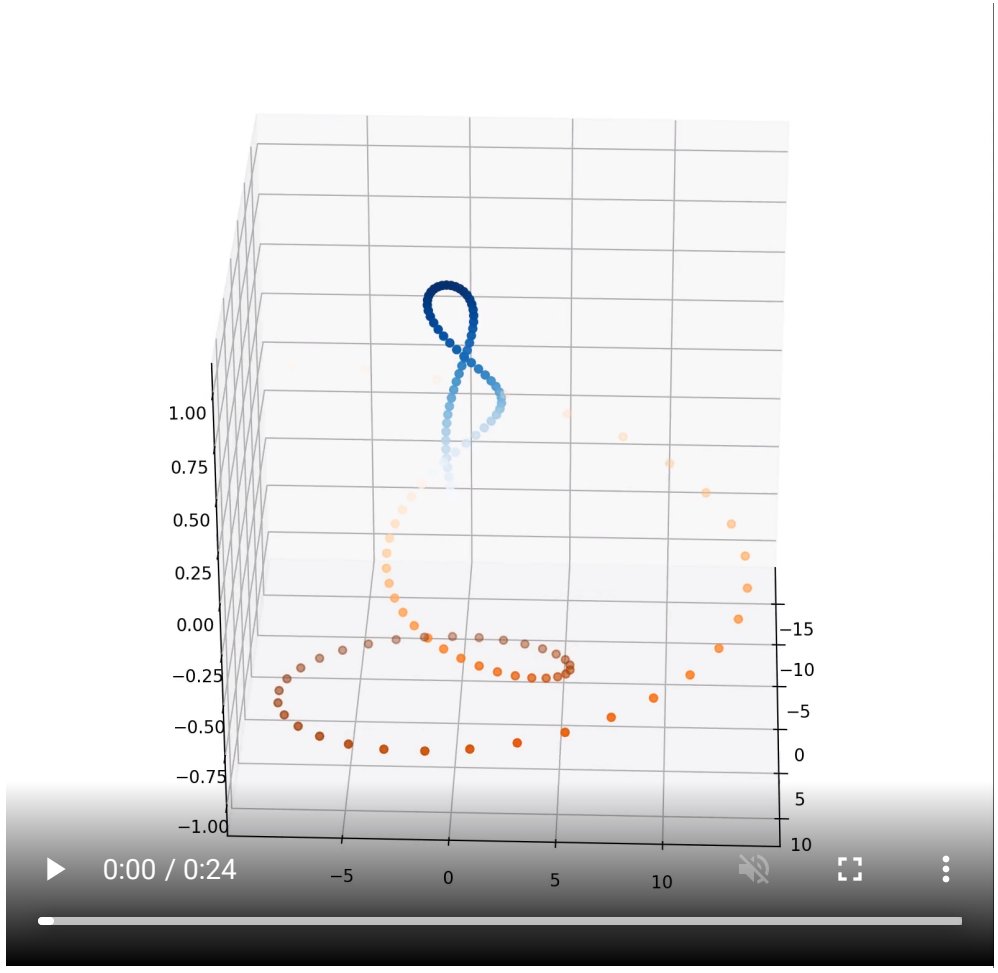






Manifold Learning

Capture the data manifold in lower dimensions



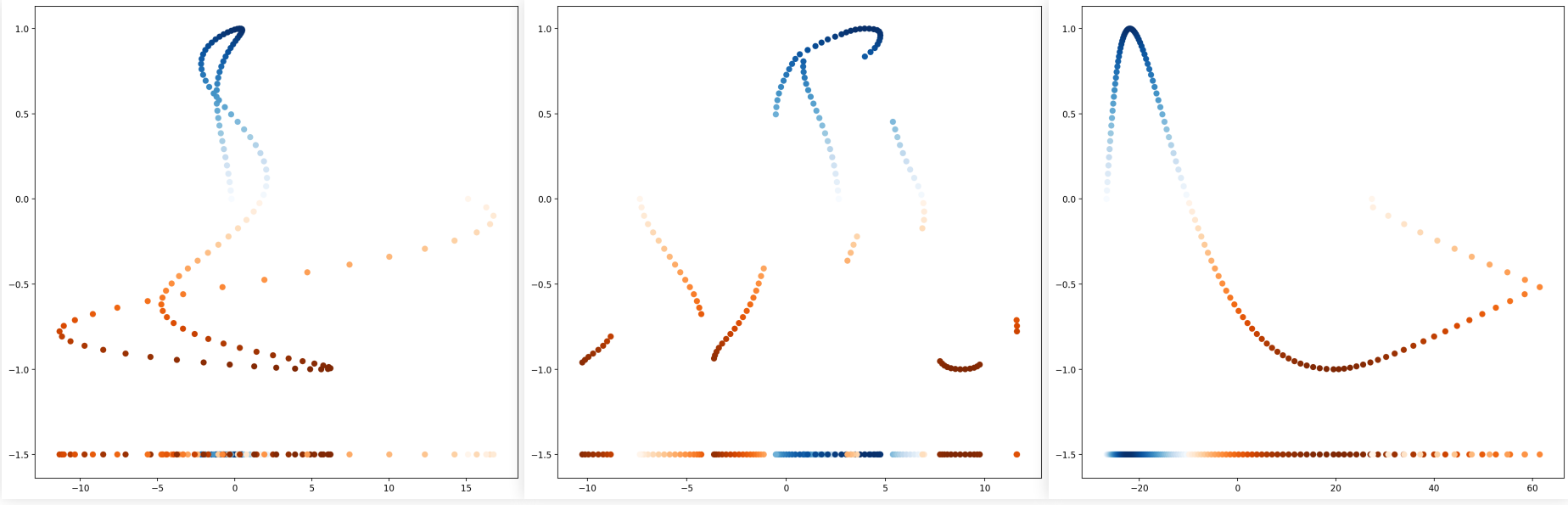
Manifold Learning

Capture the data manifold in lower dimensions

- This generally cannot be done with a linear transformation

- PCA just chooses a straight line along largest variance
- Cannot follow the manifold

- Comparison: PCA, t-SNE, Isomap



Manifold Learning

Nonlinear dimensionality reduction

- Other examples:

- Self-organizing maps
- Kernel PCA
- Autoencoders

- In Scikit-learn:

- Isomap
- t-SNE
- Locally Linear Embedding
- Spectral Embedding
- Multi-dimensional Scaling

Rand index

Rand index

- In supervised learning, we saw the confusion matrix

	Example Class	
Prediction	**Class 1**	**Class 0**
Class 1	True Positive	False Positive
Class 0	False Negative	True Negative

- In unsupervised learning we cannot match examples with classes but we can consider all $N(N-1)/2$ pairs of points:

- True Positive: a pair from the same group placed in the same cluster
- True Negative: a pair from different groups placed in different clusters
- False Positive: a pair from different groups placed in the same cluster
- False Negative: a pair from the same group placed in different clusters

Rand index

	For all $N(N-1)/2$ pairs of examples	
	Same Group	**Different Group**
Same Cluster	True Positive	False Positive
Different Cluster	False Negative	True Negative

- Using this analogy, we can compute the Rand index, which is analogous to accuracy, and other scores:

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN}$$

$$Rand = \frac{TP + TN}{N(N-1)/2} \qquad F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Rand index

Adjusted rand index

- The Rand index can be artificially high by mere chance
- Evem if clusters are assigned at random in many cases pairs from different groups will fall into different clusters
- The adjusted Rand index solves this problem by correcting the index with the index value expected by chance:

$$ARI = \frac{RI - E(RI)}{1 - E(RI)}$$

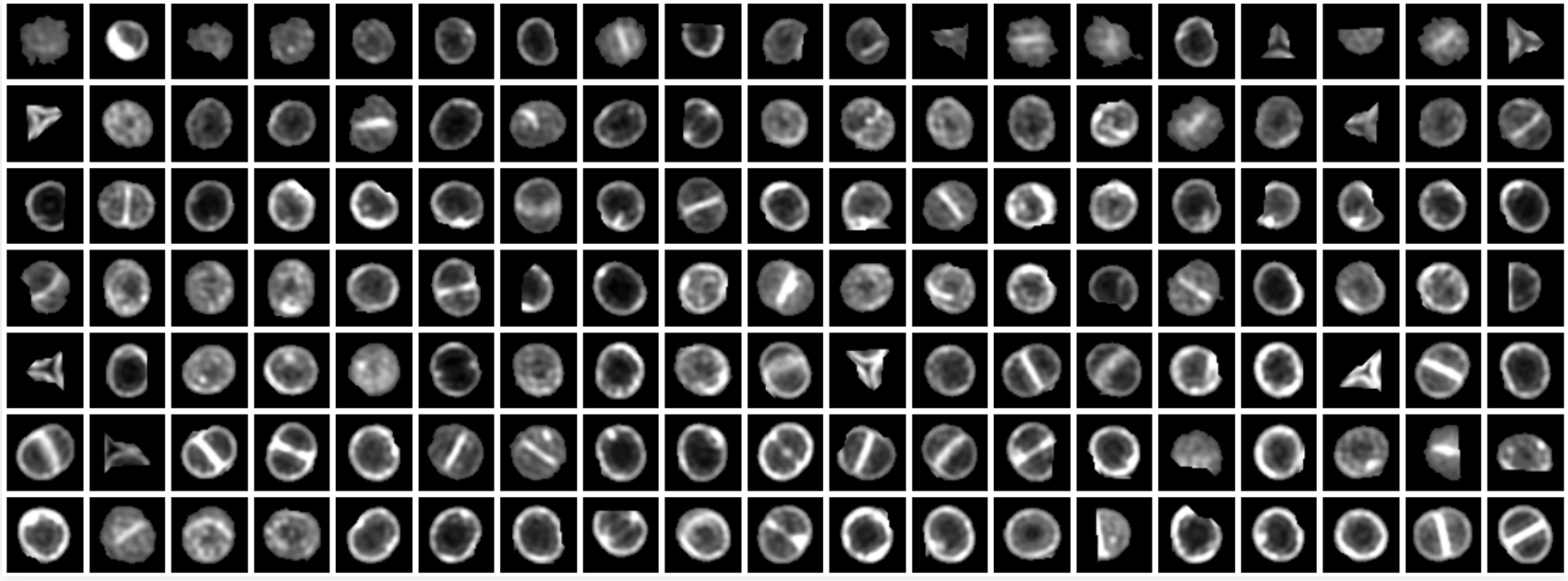
- Note: while the Rand index must be between 0 and 1, the ARI can be negative.
- In `sklearn`:

```
sklearn.metrics.adjusted_rand_score
```

Assignment 2

Assignment 2

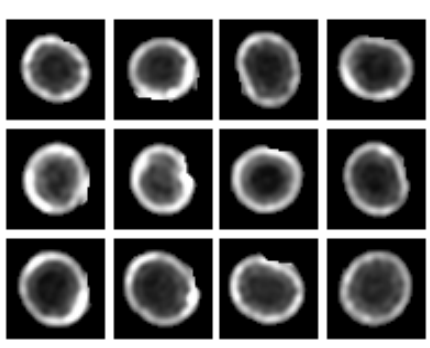
Clustering of cell images (*Staphylococcus aureus*)

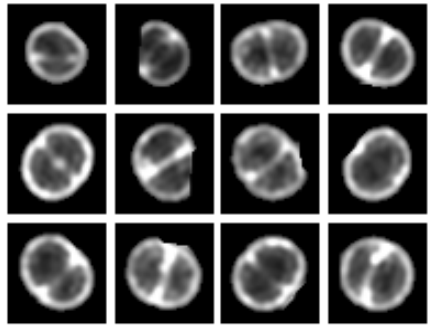


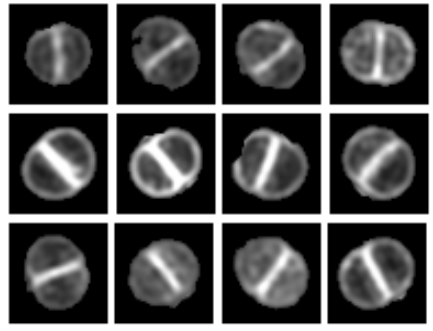
- Images from fluorescence microscopy
- Automated segmentation (images include segmentation errors)

Assignment 2

- Data set of 563 images of 50x50 pixels each, in `images` folder
- Each image has a black background and the segmented region centered
- Some images are labelled (1, 2 and 3) with the cell cycle phase







- Labels are in file `labels.txt` (image ID, label)

- A label of 0 indicates the image was not labelled

Assignment 2

Objective: help the biologists

- Propose a procedure to aid the biologists

- Reject segmentation errors
- Select cells of some type
- Classify according to the cell cycle phase
- ...

- Clustering similar cells can help deal with groups instead of individual examples

Assignment 2

Different from assignment 1

- Assignment 1 was mostly implementation and following the recipes
- Assignment 2 is more realistic: you are the data scientists

- You must figure out what to do with the data
- (The biologists know nothing about clustering)

- Part of the assignment is scripted:

- Use PCA, t-SNE and Isomap to extract 18 features from the images
- (6 components each)
- Test at least 2 clustering algorithms (K-Means and DBSCAN)
- Find ϵ parameter with method from DBSCAN paper.
- Implement and measure cluster quality metrics (rand index, ARI, Silhouette, etc)

- The rest is up to you:

- Select features, optimize clustering, evaluate results and decide how to use them

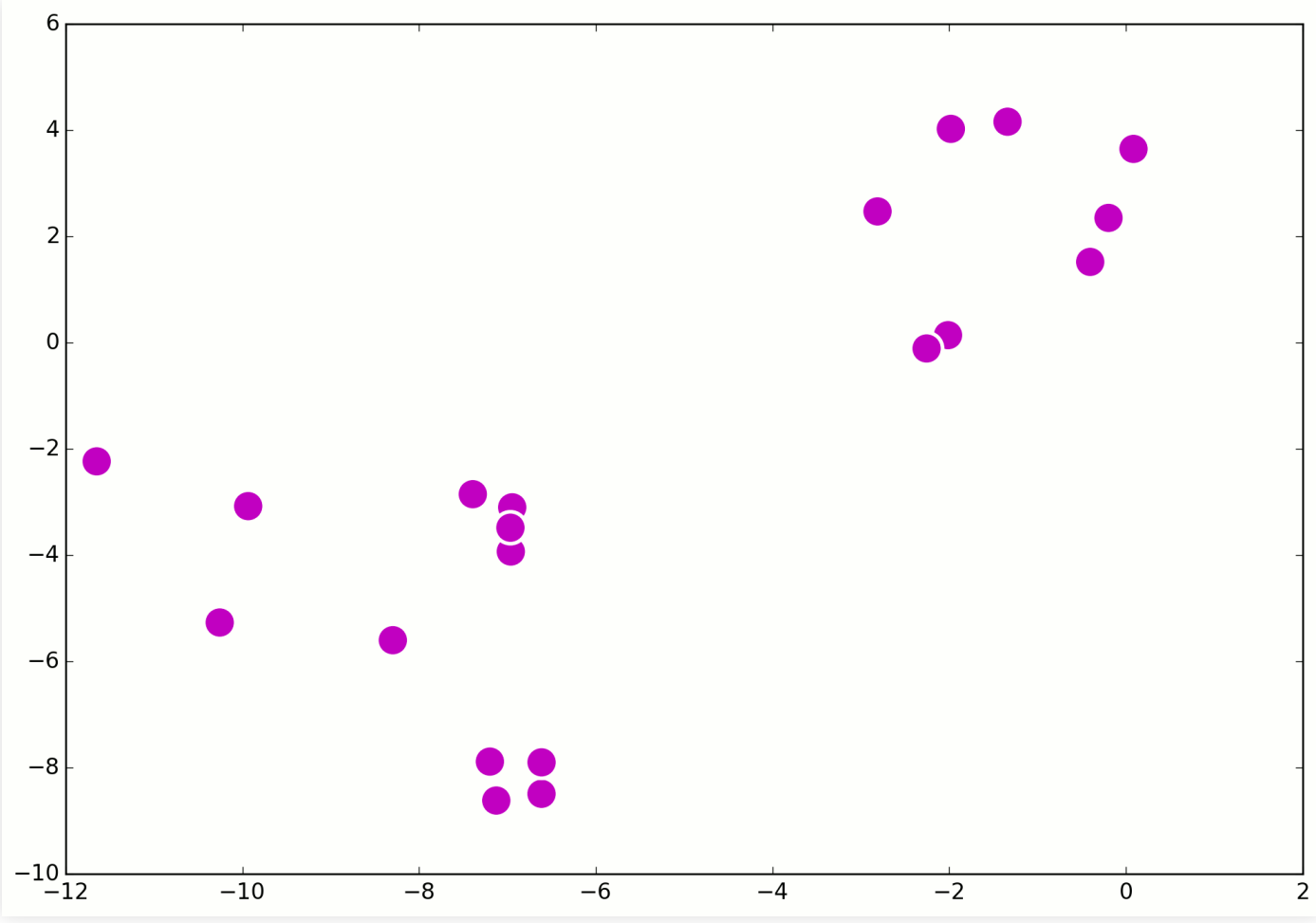
Assignment 2

Practical details

- Read the assignment page carefully
  - And check it occasionally, as I may update it for clarifications
- Download zip file with:
  - Images in the `images` folder
  - Labels (in `labels.txt`)
  - Auxiliary functions for loading data and generating HTML reports (`tp2_aux.py`)
  - Report example using the manual labels (`example_labels.html`)
- Implement the code, examine the results
- Answer the questions (`TP2.txt`)

Assignment 2

Optional: implement bisecting k-means



Summary

Summary

- Fuzzy sets: model uncertainty
- Fuzzy c-means: clustering with membership values
- Manifold learning
- Rand index and Assignment 2

Further reading

- Manifold learning on Scikit-Learn

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