Projections and clusterings

AI for ecologists

Paul Tresson

21/05/25

Institut de Recherche pour le Développement FRANCE

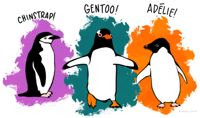
AMAPlab

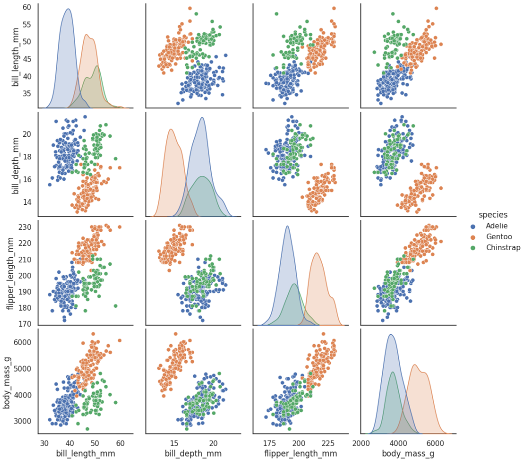
FRB

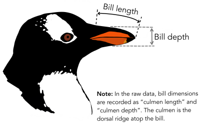
CESAB

Introduction

Visualize what is happening in higher dimensions

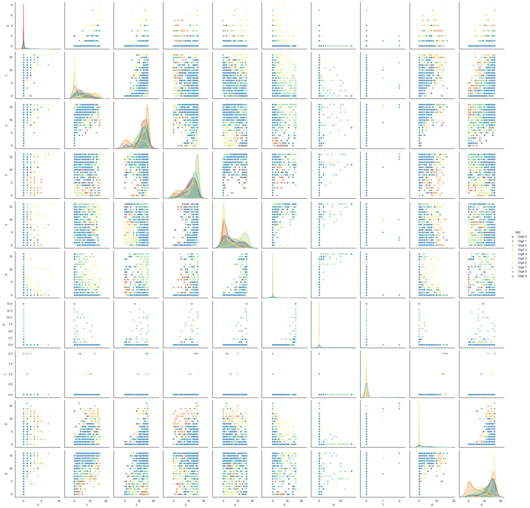




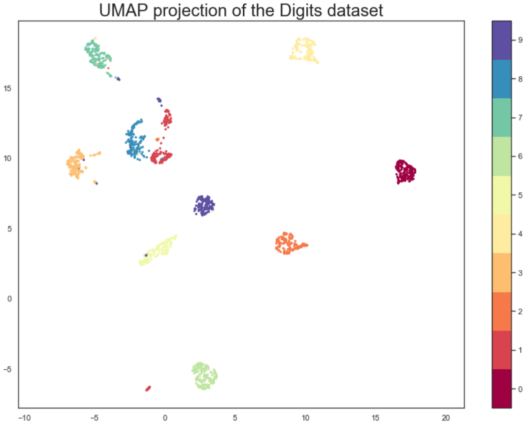


Visualize what is happening in higher dimensions

Visualize what is happening in higher dimensions



Visualize what is happening in higher dimensions

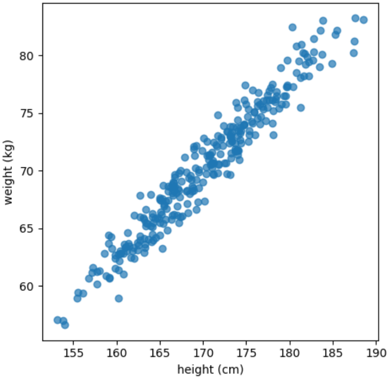


Projections

Projections

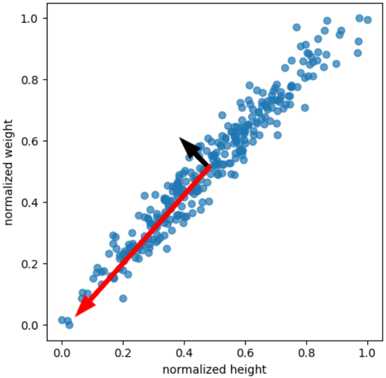
Principa Component Analysis

PCA



weight (kg)
80
75
70
65
60
155
160
165
170
175
180
185
190
height (cm)

PCA



1.0
0.8
0.6
0.4
0.2
0.0
normalized weight
0.0
0.2
0.4
0.6
0.8
1.0
normalized height

PCA calculation

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \end{bmatrix}$$

PCA calculation

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \end{bmatrix}$$

$$C = \frac{1}{n-1} X^T X$$

$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

PCA calculation

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \end{bmatrix}$$

$$C = \frac{1}{n-1} X^T X$$

$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

$$Cv = \lambda v$$

$$\begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \lambda \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

PCA calculation

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \end{bmatrix}$$

$$C = \frac{1}{n-1} X^T X$$

$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

$$Cv = \lambda v$$

$$\begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \lambda \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$P = \begin{bmatrix} v_x & v_y \end{bmatrix}$$

PCA calculation

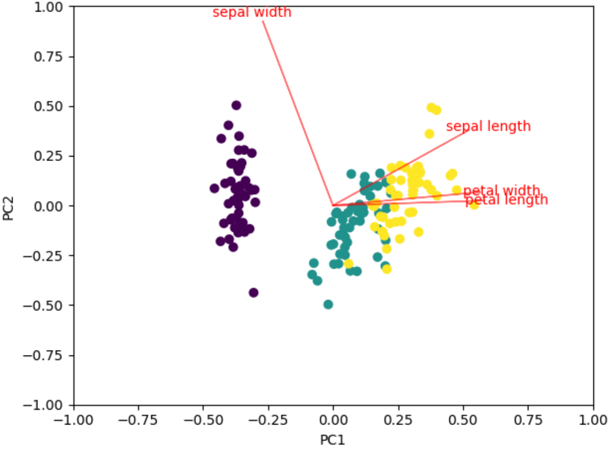
```python
# normalize
normalized_data = (points - np.min(points, axis=0)) / (
    np.max(points, axis=0) - np.min(points, axis=0)
)

# get covariance
cov = np.cov(normalized_data, rowvar=False)

# calculate eigenvalues and eigenvectors of the covariance matrix
eigvals, eigvecs = np.linalg.eig(cov)

# scale eigenvectors
scaled_eigvecs = eigvecs * np.sqrt(eigvals)
```

PCA interpretability



PCA advantages and drawbacks

Advantages

- fast

Drawbacks

PCA advantages and drawbacks

Advantages

- fast
- scales well

Drawbacks

PCA advantages and drawbacks

Advantages

- fast
- scales well
- explainatory

Drawbacks

PCA advantages and drawbacks

Advantages

- fast
- scales well
- explainatory

Drawbacks

- **not suited for non-linear data**

Non linear data

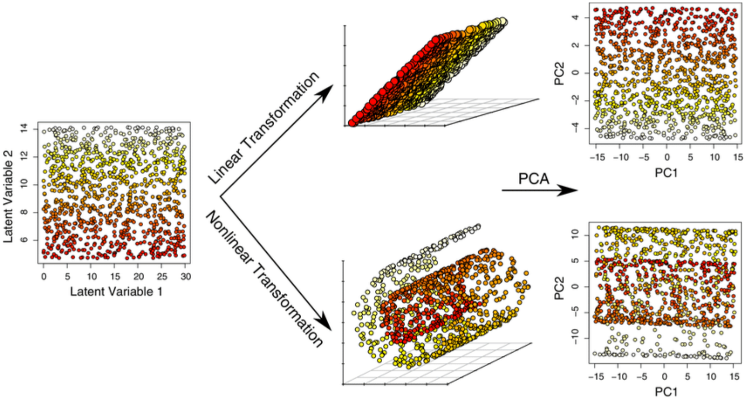


Figure from Du, 2019

Projections

UMAP

How to work with non linear data

1. Find a good representation of the data in high dimension
2. Fit a good representation of in low dimension

Classics

- **SNE** (Stochastic Neighbor Embedding) Hinton and Roweis, 2002

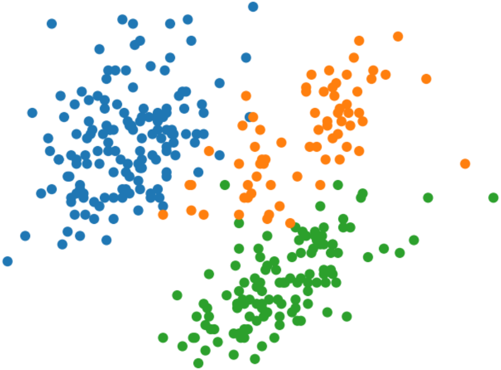
Classics

- **SNE** (Stochastic Neighbor Embedding) Hinton and Roweis, 2002
- **T-SNE** (T-distributed SNE) Van der Maaten and Hinton, 2008

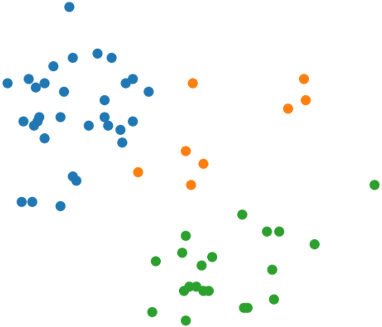
Classics

- **SNE** (Stochastic Neighbor Embedding) Hinton and Roweis, 2002
- **T-SNE** (T-distributed SNE) Van der Maaten and Hinton, 2008
- **UMAP** (Uniform Manifold Approximation and Projection) McInnes et al., 2018

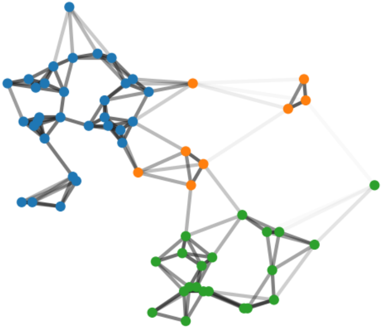
UMAP : Finding high dimension graph



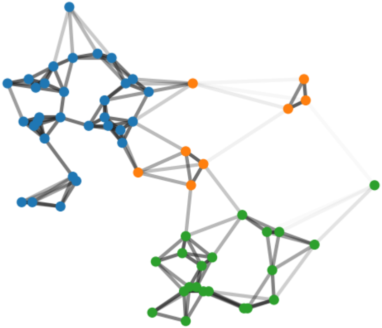
UMAP : Finding high dimension graph

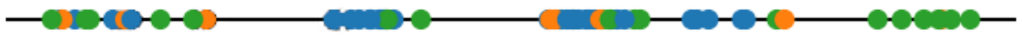


UMAP : Finding high dimension graph

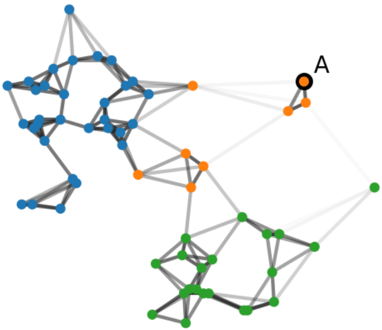


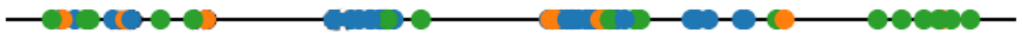
UMAP : Fitting low dimensional representation



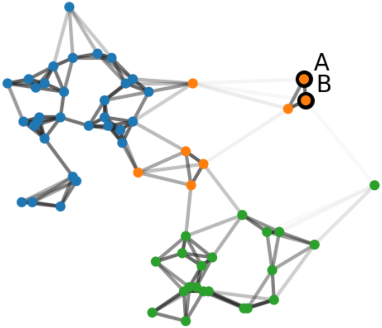


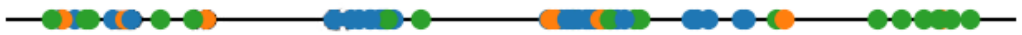
UMAP : Fitting low dimensional representation



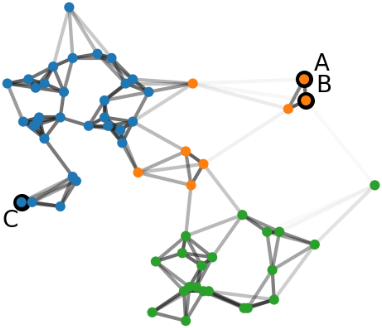


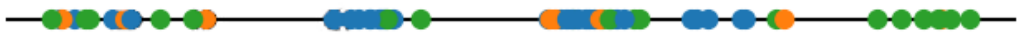
UMAP : Fitting low dimensional representation



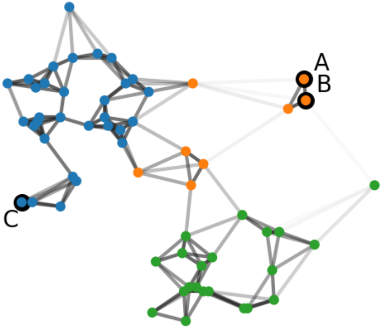


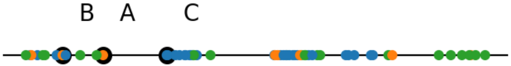
UMAP : Fitting low dimensional representation



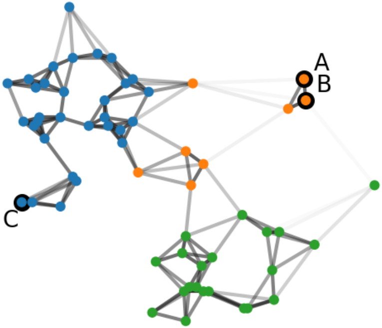


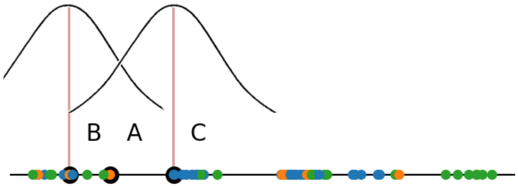
UMAP : Fitting low dimensional representation



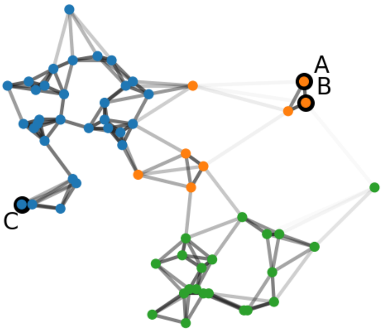


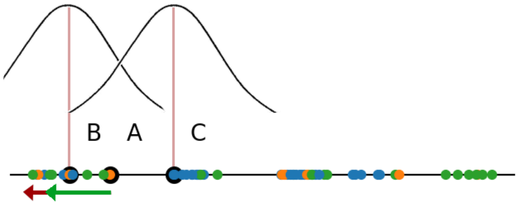
UMAP : Fitting low dimensional representation



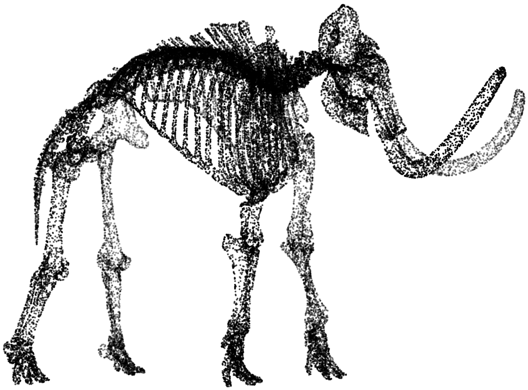


UMAP : Fitting low dimensional representation



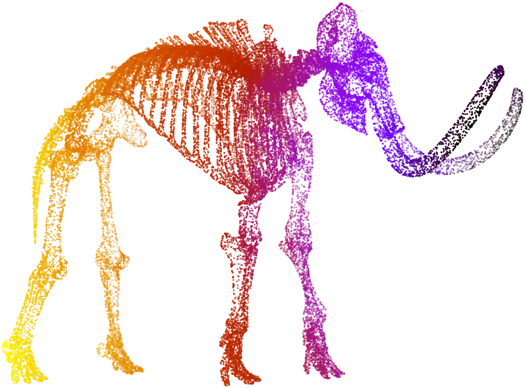


Mammoth example



Mammoth dataset by the Smithonian, adatpted from deepia

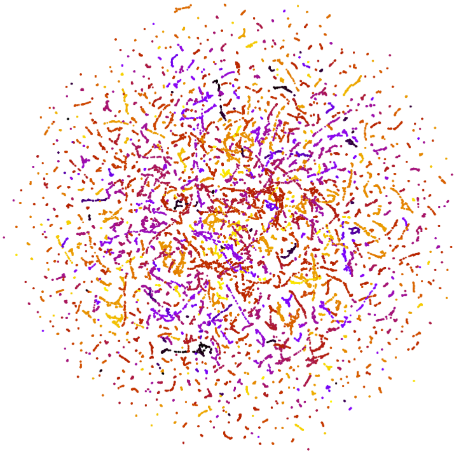
Mammoth example



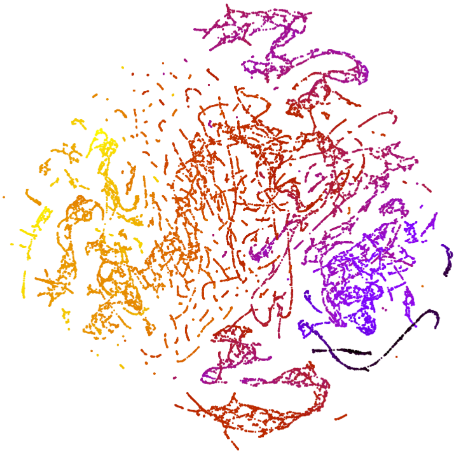
Mammoth dataset by the Smithonian, adatpted from deepia

Mammoth example - PCA

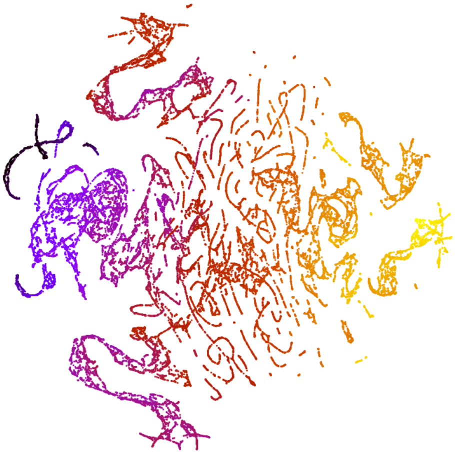
Mammoth example - UMAP



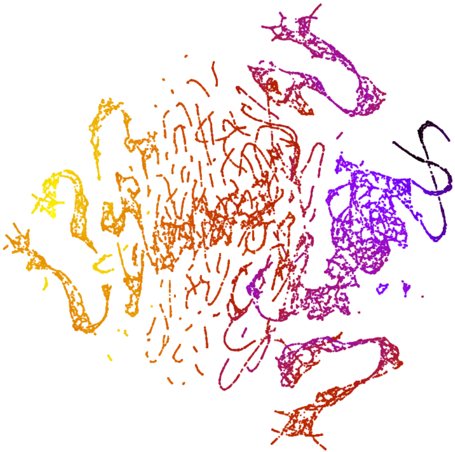
Mammoth example - UMAP



Mammoth example - UMAP

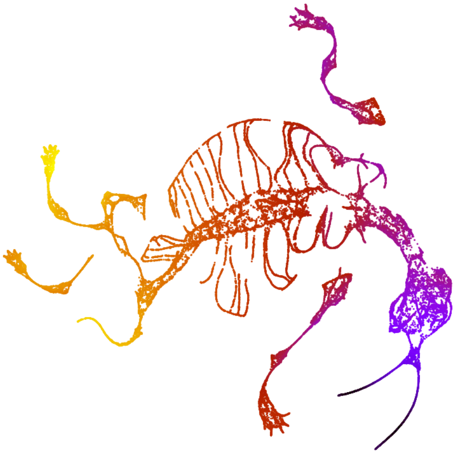


Mammoth example - UMAP



Mammoth example - UMAP

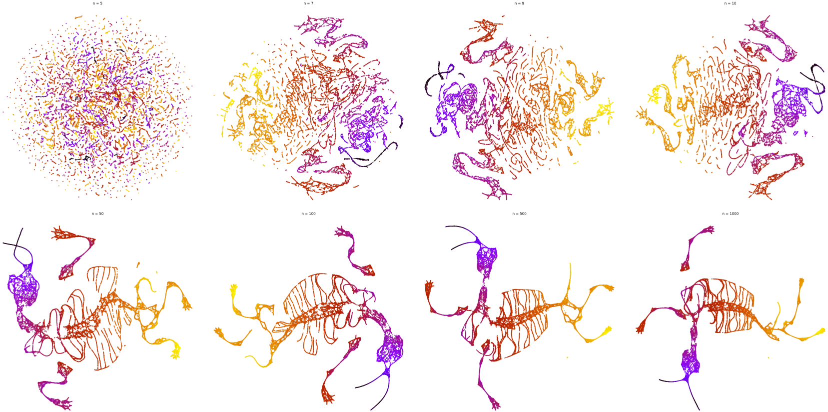
Mammoth example - UMAP



Mammoth example - UMAP

Mammoth example - UMAP

Mammoth example - Number of neighbors



UMAP vs. T-SNE

UMAP

- graph theory, fuzzy logic

T-SNE

- probabilities

UMAP vs. T-SNE

UMAP

- graph theory, fuzzy logic
- determinist initialization

T-SNE

- probabilities
- random initialization

UMAP vs. T-SNE

UMAP

- graph theory, fuzzy logic
- determinist initialization
- log_2 similarity scores

T-SNE

- probabilities
- random initialization
- gaussian distribution similarity

UMAP vs. T-SNE

UMAP

- graph theory, fuzzy logic
- determinist initialization
- log_2 similarity scores
- update by pairs

→ scales well for large datasets

T-SNE

- probabilities
- random initialization
- gaussian distribution similarity
- update all points

scales less

Scaling

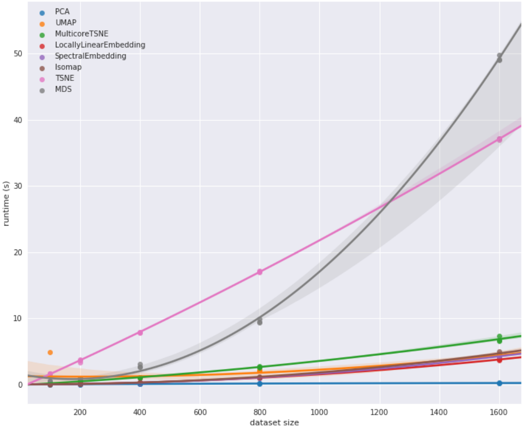


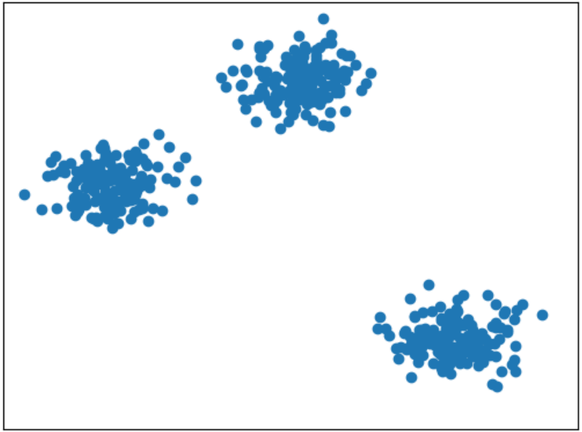
Figure from umap-learn documentation

Clustering

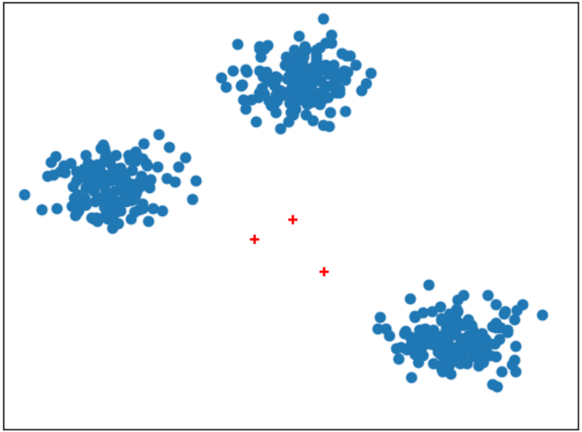
Clustering

K-means

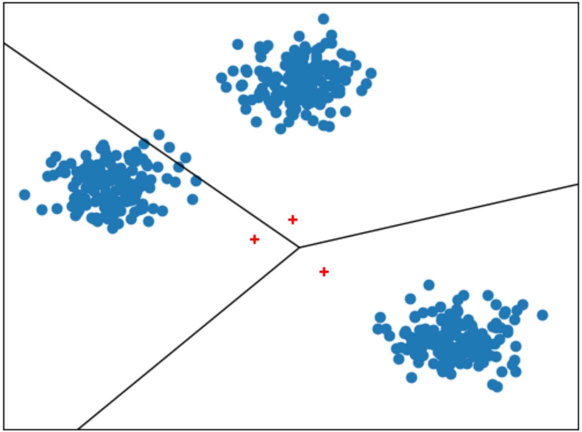
K-means



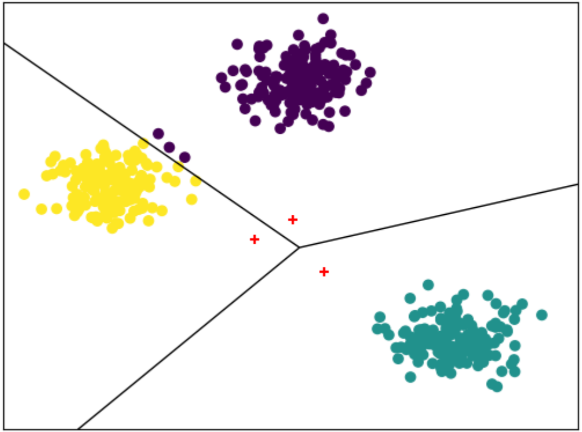
K-means



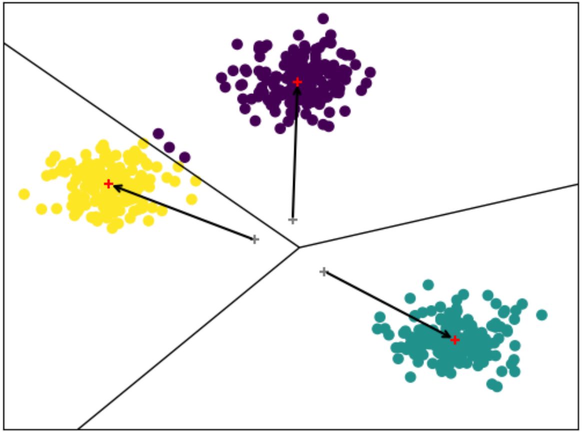
K-means



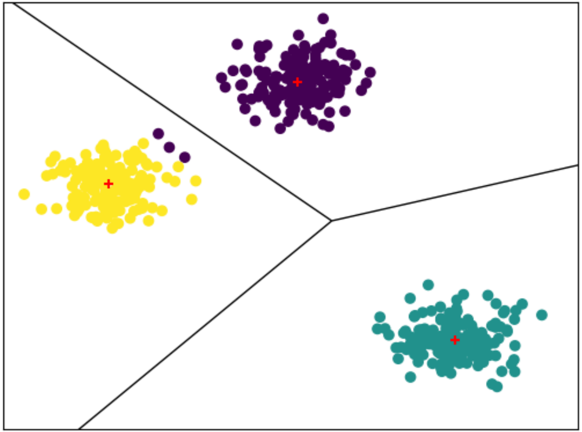
K-means



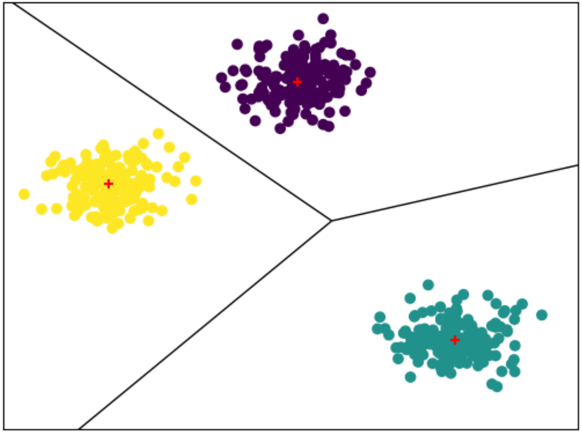
K-means



K-means



K-means



Determine the good K ? Elbow method

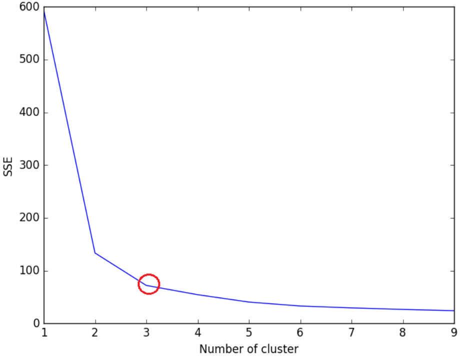
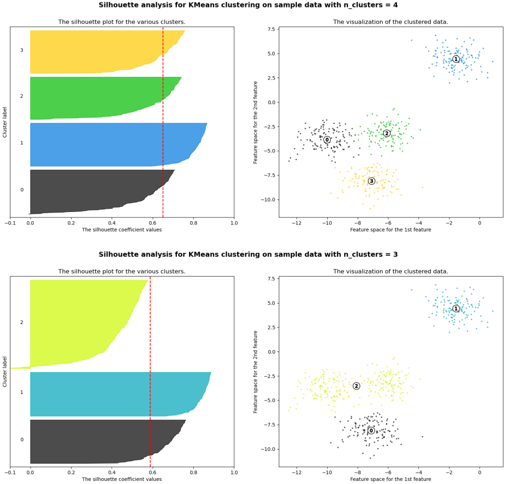


Figure from SO

Determine the good K ? Silouhette



Figures from scikit-learn documentation

K-means advantages and drawbacks

Advantages

- fast

Drawbacks

K-means advantages and drawbacks

Advantages

- fast
- scales well

Drawbacks

K-means advantages and drawbacks

Advantages

- fast
- scales well
- converges well

Drawbacks

K-means advantages and drawbacks

Advantages

- fast
- scales well
- converges well
- robust

Drawbacks

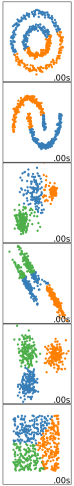
K-means advantages and drawbacks

Advantages

- fast
- scales well
- converges well
- robust

Drawbacks

- **not suited for non-linear data**



K-means advantages and drawbacks

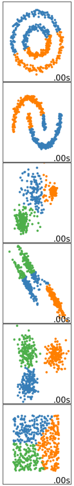
Advantages

- fast
- scales well
- converges well
- robust

Drawbacks

- **not suited for non-linear data**

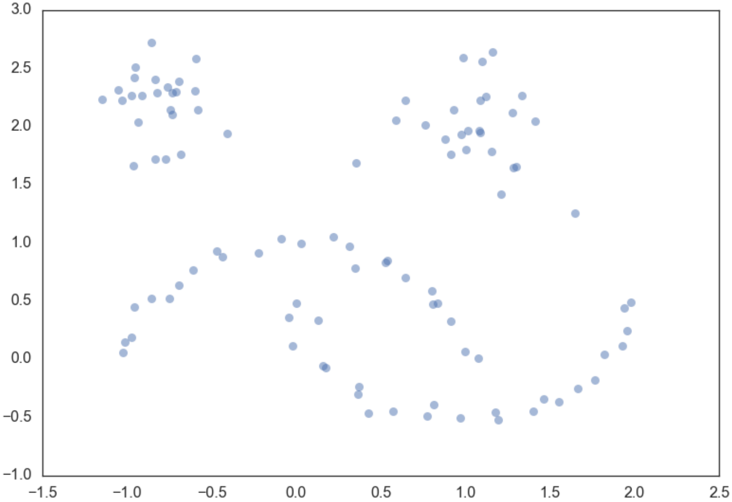
$\rightarrow$ Density based clustering



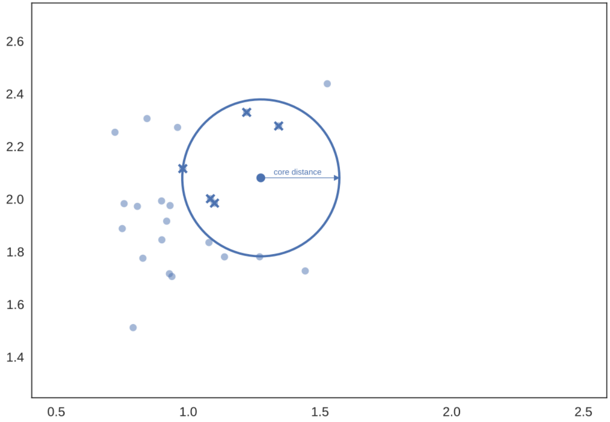
Clustering

HDBSCAN

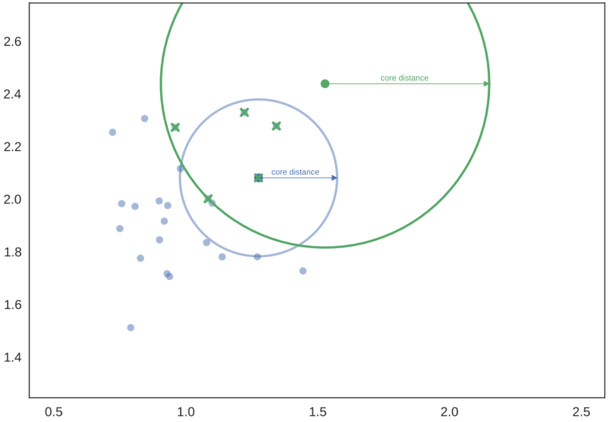
HDBSCAN



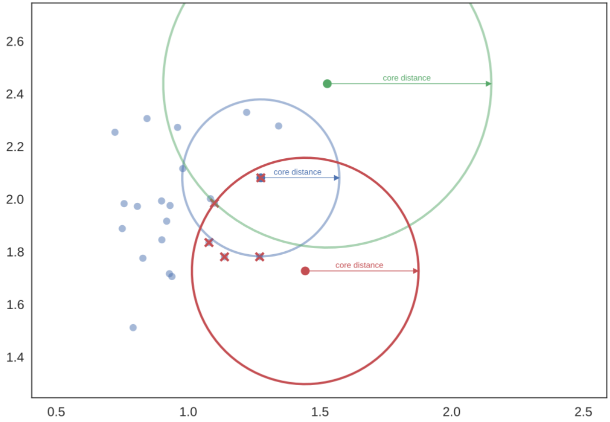
HDBSCAN



HDBSCAN

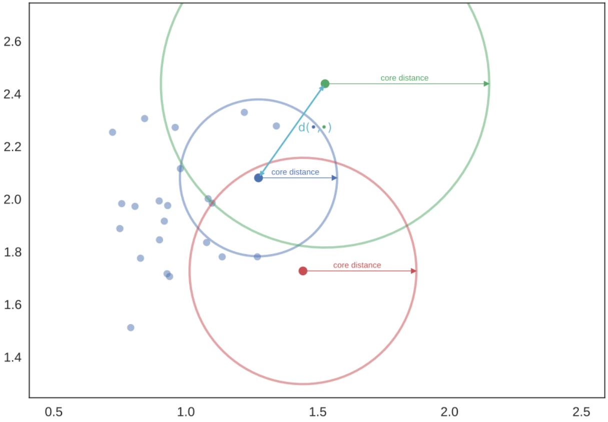


HDBSCAN



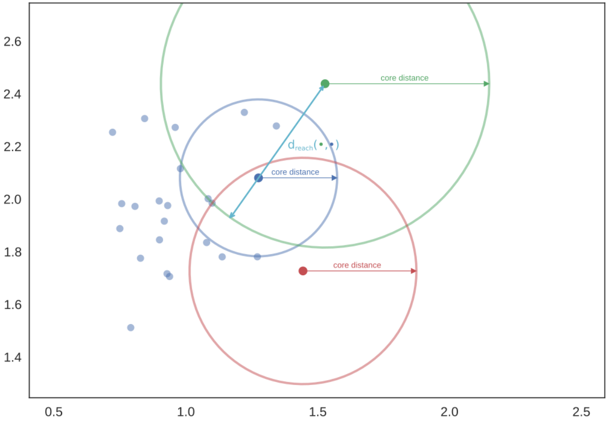
core distance
core distance
core distance
2.6
2.4
2.2
2.0
1.8
1.6
1.4
0.5
1.0
1.5
2.0
2.5

HDBSCAN



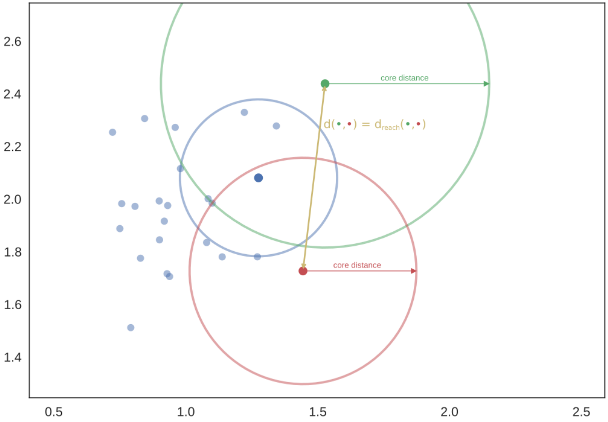
2.6
2.4
2.2
2.0
1.8
1.6
1.4
0.5
1.0
1.5
2.0
2.5
core distance
core distance
core distance

HDBSCAN



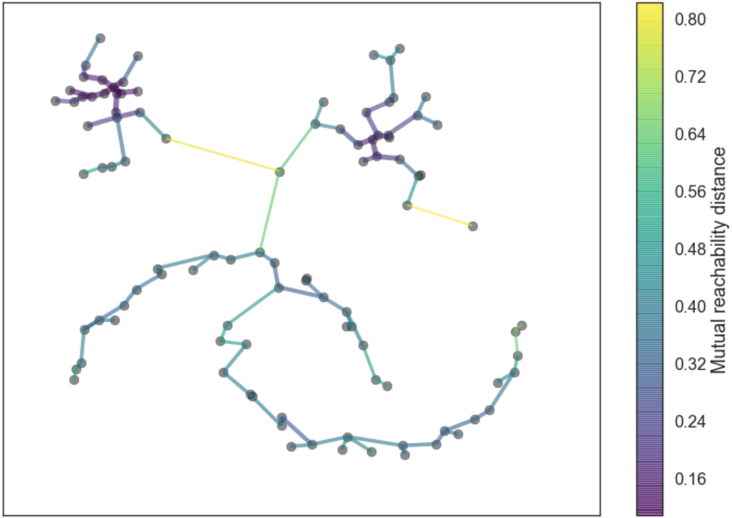
core distance
core distance
core distance
2.6
2.4
2.2
2.0
1.8
1.6
1.4
0.5
1.0
1.5
2.0
2.5

HDBSCAN



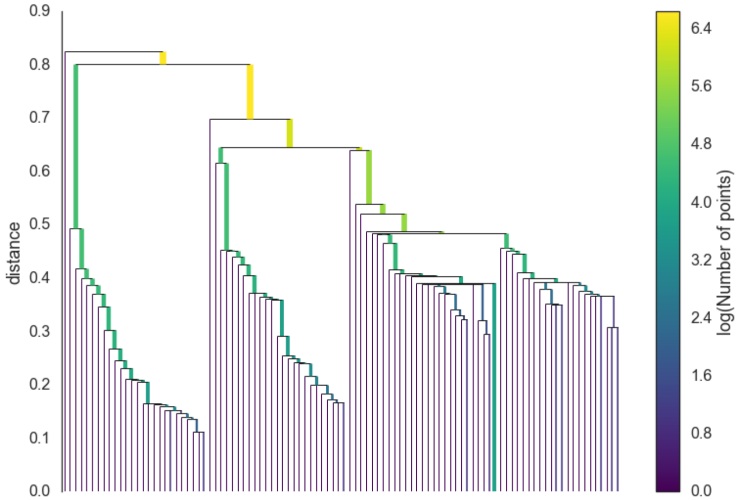
core distance
d(•,•) = d_reach(•,•)
core distance
2.6
2.4
2.2
2.0
1.8
1.6
1.4
0.5
1.0
1.5
2.0
2.5

HDBSCAN



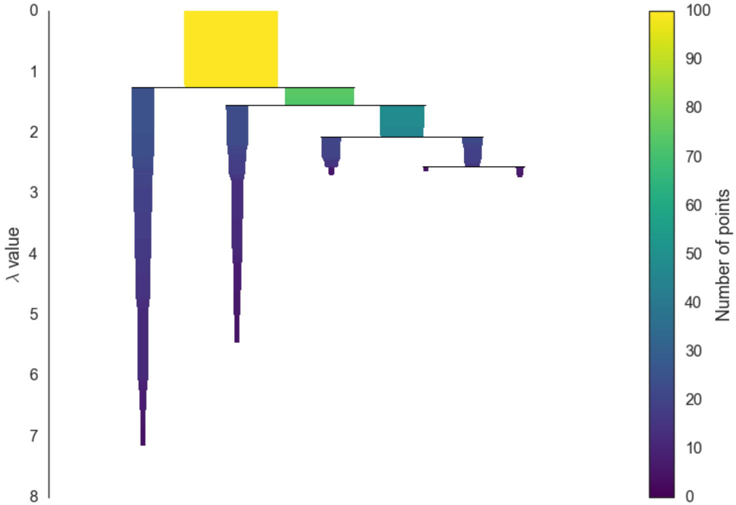
0.80
0.72
0.64
0.56
0.48
0.40
0.32
0.24
0.16
Mutual reachability distance

HDBSCAN

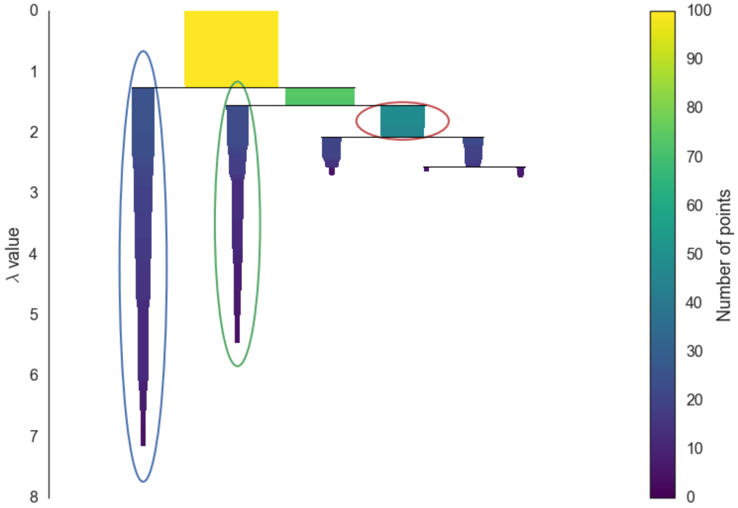


distance
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
log(Number of points)
6.4
5.6
4.8
4.0
3.2
2.4
1.6
0.8
0.0

HDBSCAN

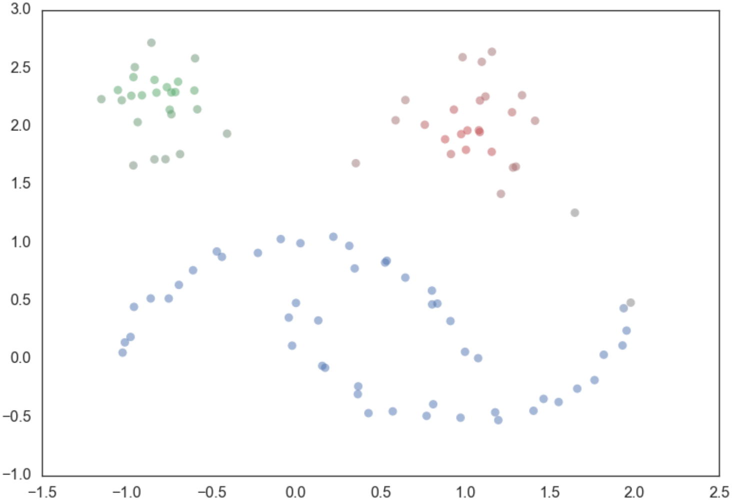

0
1
2
3
4
5
6
7
8
λ value
100
90
80
70
60
50
40
30
20
10
0
Number of points

HDBSCAN

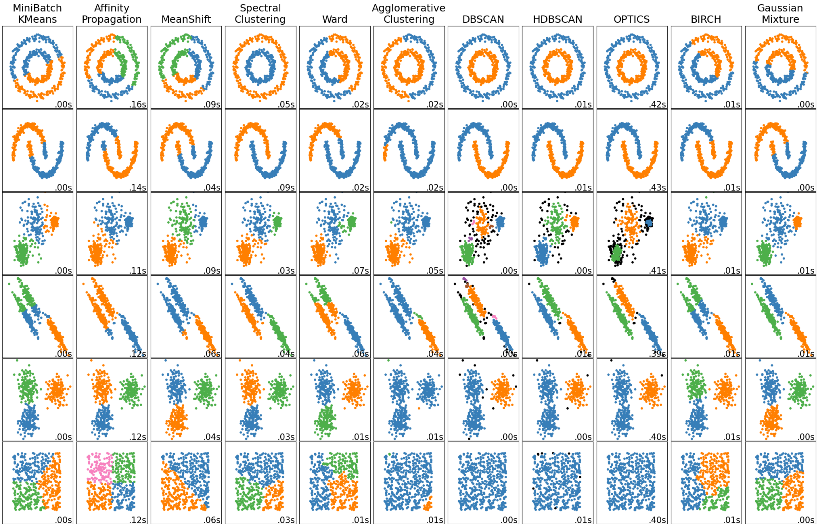


0
1
2
3
4
5
6
7
8
λ value
100
90
80
70
60
50
40
30
20
10
0
Number of points

HDBSCAN



A lot of options !



Useful ressources

- `scikit-learn` docs !
- deepia
- StatQuest

Thanks for you attention !

Let's practice !

References i

Du, Trina Y (2019). **"Dimensionality reduction techniques for visualizing morphometric data: Comparing principal component analysis to nonlinear methods".** In: *Evolutionary Biology* 46.1, pp. 106–121.

Hinton, Geoffrey E and Sam Roweis (2002). **"Stochastic neighbor embedding".** In: *Advances in neural information processing systems* 15.

McInnes, Leland, John Healy, and James Melville (2018). **"Umap: Uniform manifold approximation and projection for dimension reduction".** In: *arXiv preprint arXiv:1802.03426*.

Van der Maaten, Laurens and Geoffrey Hinton (2008). **"Visualizing data using t-SNE.".** In: *Journal of machine learning research* 9.11.