Unsupervised learning for data exploration

Navigating data's hidden patterns

Siham El Garroussi

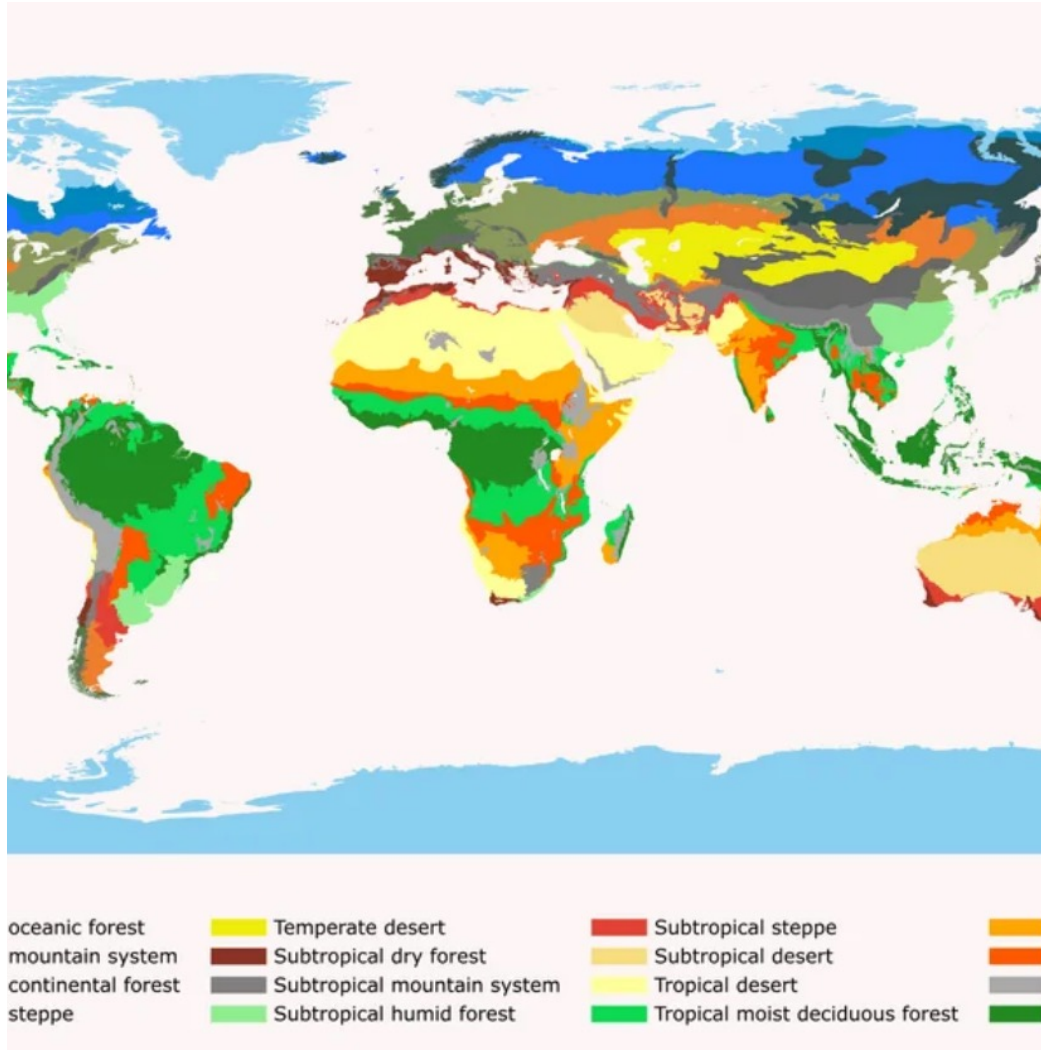
ECMWF

siham.garroussi@ecmwf.int

ECMWF

Motivation

- **Grouping areas into climate zones** such as “tropical”, “temperate”, and “polar”.
- Why?
  - Creating a unique climate model for each small region would be complex and resource-intensive.
  - Using a single global climate model might fail to capture local nuances and regional characteristics.
  - Thus, climate zones help in applying general but relevant models to large areas sharing similar climate features.



Introduction

Reminder: previous talk

Semi-supervised learning

Supervised learning

- Discover patterns in the data with **known target or label.**
  - ❑ These patterns are then utilised to predict the values of the target attribute in future data instances.

Unsupervised learning

- The data have **no target** attribute.
  - ❑ We want to explore the data to find some intrinsic structures in them.

Reinforcement learning

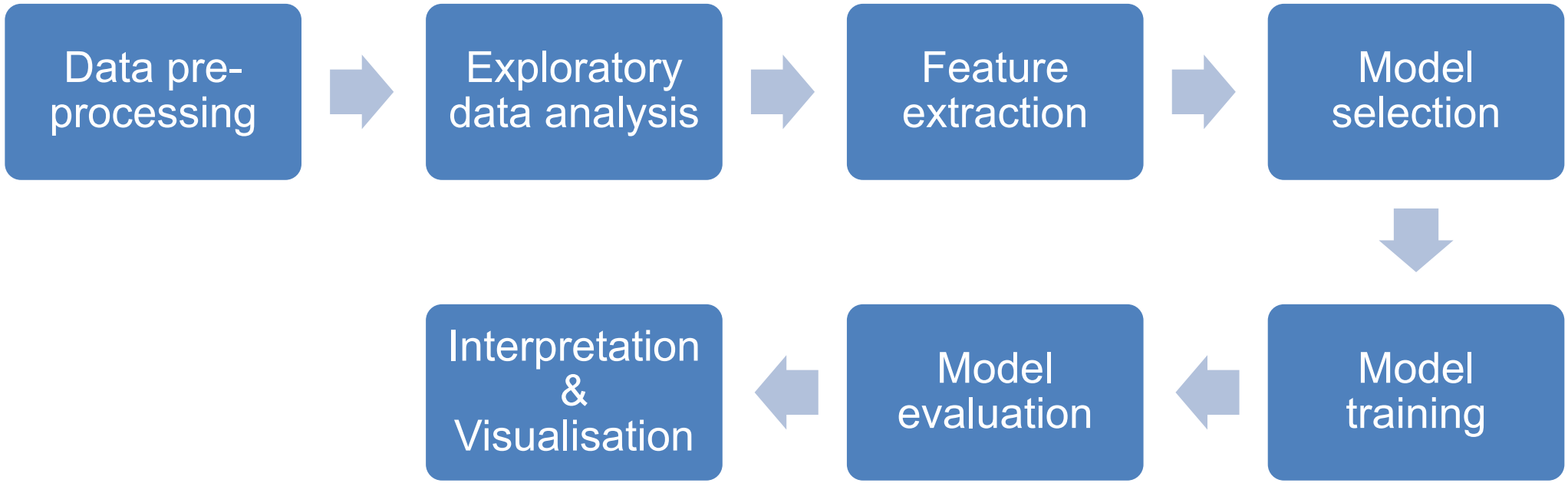
- Trains algorithms to **learn from their environments**.
  - ❑ Decision making through subsequent interactions with the environment that result in rewards.

Introduction

Q: Can we perform a regression task using unsupervised learning techniques?

Framework of unsupervised learning

- Unsupervised learning is a data-driven framework that autonomously uncovers hidden patterns in data without predefined labels.
- The model's effectiveness heavily depends on the data quality,
- **Poor data quality can lead to unreliable models.**



Applications of unsupervised learning

- Inferring hidden structures in an unlabelled data.
- Each type of unsupervised learning has its strengths and weaknesses, and the choice of algorithm depends on the specific characteristics of the data and the goals of the analysis.

Clustering

Divide by similarity

Anomaly detection

Identify outliers

Association rule learning

Identify sequences

Dimensionality reduction

Image segmentation

Preparing data for supervised learning

Generative modelling

Learn a distribution to generate new, similar points

Applications of unsupervised learning

- Inferring hidden structures in an unlabelled data.
- Each type of unsupervised learning has its strengths and weaknesses, and the choice of algorithm depends on the specific characteristics of the data and the goals of the analysis.

Clustering

Divide by similarity

Dimensionality reduction

Anomaly detection

Identify outliers

Image segmentation

Generative modelling

Learn a distribution to generate new, similar points

Association rule learning

Identify sequences

Preparing data for supervised learning

Clustering

Clustering: Task of grouping a set of data points such that data points in the same group (cluster) are more similar to each other than data points in another group (cluster).

- Cluster is represented by a single point, known as centroid (or cluster center) of the cluster.
- Centroid is computed as the mean of all data points in a cluster:

$$C_j = \sum x_i,$$

- Cluster boundary is decided by the farthest data point in the cluster.

Clustering: Applications

- **Example 1: Segregating urban areas based on their vulnerability to specific climate threats** , such as sea-level rise, urban heat islands, and air pollution.

- This segmentation allows for targeted climate resilience and adaptation strategies.

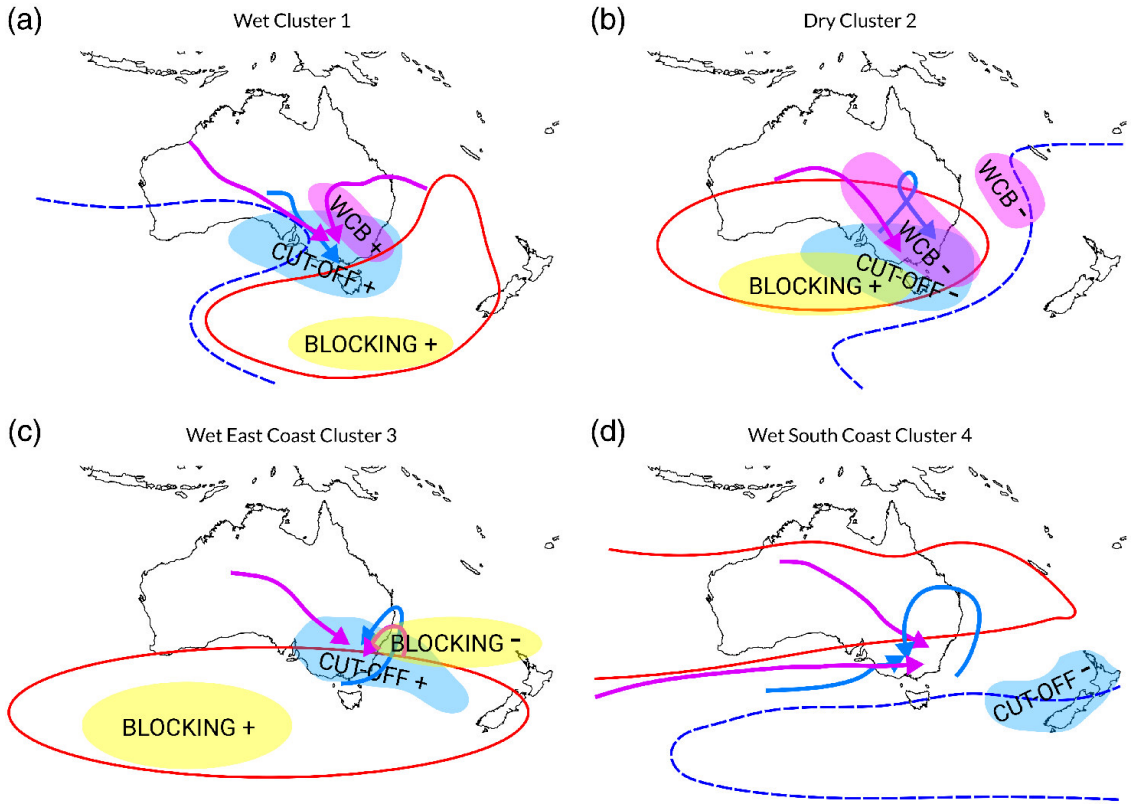
- e.g., coastal cities require different infrastructure and planning compared to cities at risk of heat waves.

Credit: kwest/Shutterstock.com

Clustering: Applications

- **Example 2: Categorising "weather systems" based on atmospheric conditions such as air pressure patterns, jet stream positions, and humidity levels.**

- identify distinct patterns, "weather system": high-pressure blocks, low-pressure systems, or specific jet stream configurations.
- These systems are closely linked to certain weather outcomes like clear skies, storms, or prolonged rain periods.
- Changes of weather system frequency determine the respective rainfall anomaly pattern.



A weather system perspective on winter-spring rainfall variability in southern Australia during El Nino, Hauser+, Quart J Royal Meteoro Soc, Volume: 146, Issue: 731, Pages: 2614-2633, First published: 28 April 2020, DOI: (10.1002/qj.3808)

Clustering

- Types:
  - Exclusive clustering: K-means
  - Overlapping clustering: Fuzzy C-means
  - Density-based clustering: DBSCAN
  - Hierarchical clustering: agglomerative clustering, divisive clustering
  - Probabilistic clustering: mixture of Gaussian models,
  - …

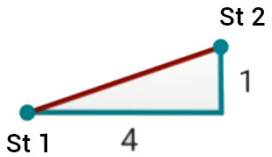
Clustering

How do we calculate the distance between two points?

- Example of distance metrics:
  - Euclidian distance
  - Manhattan distance
  - Maximum distance
  - Chebychev distance
  - Cosine similarity
  - Jaccard similarity
  - Minkowski metric, …

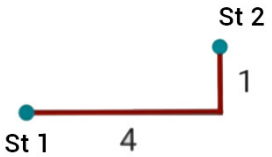
Euclidean distance

$$d = \sqrt{4^2 + 1^2} = 3{,}162$$



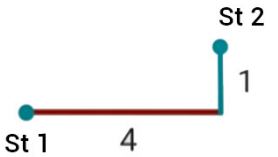
Manhattan distance

$$d = 4 + 1 = 5$$



Maximum distance

$$d = max(4\,,1) = 4$$

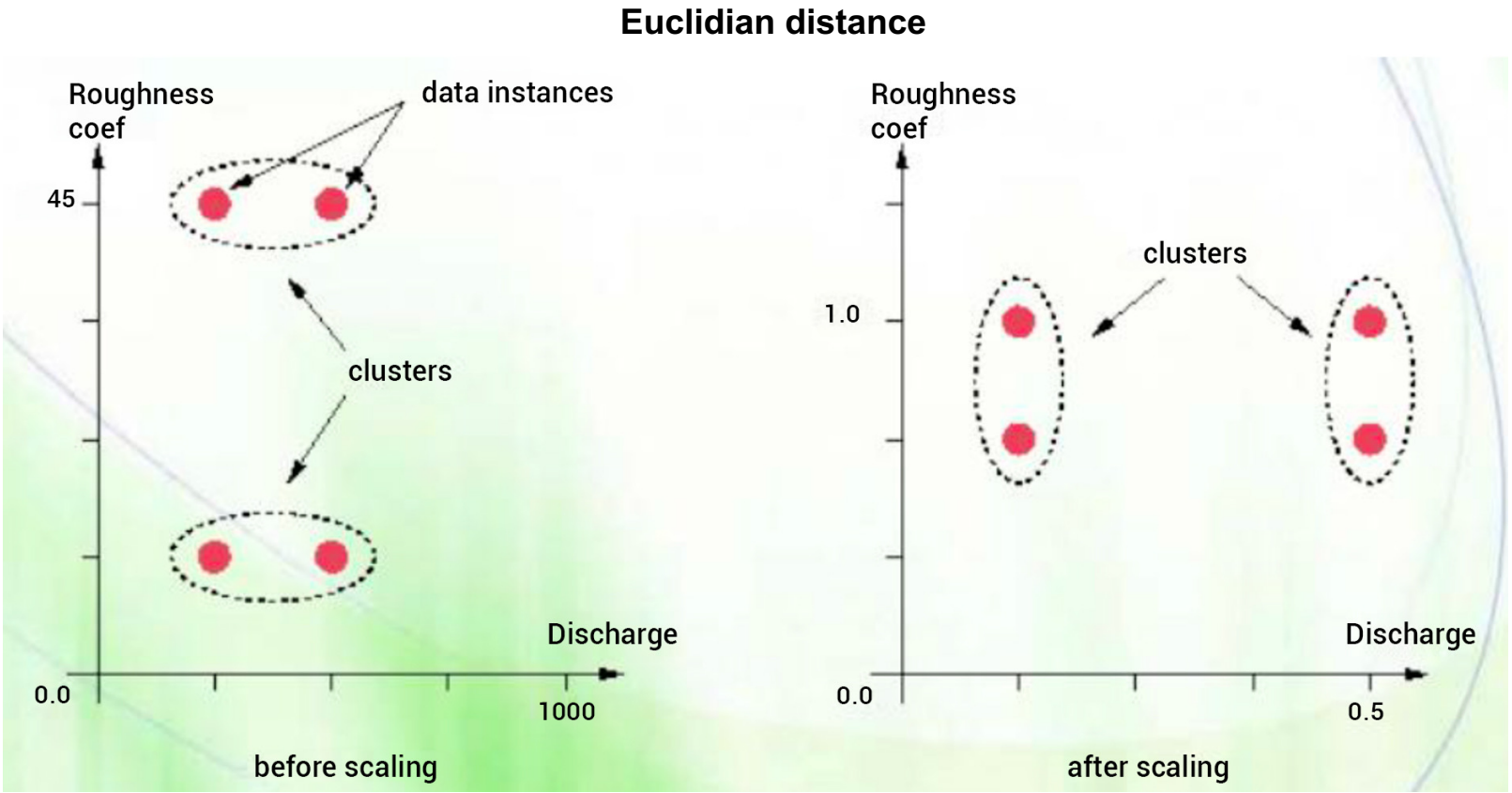


St = Observation station

- The choice of distance metric can significantly influence the clusters formed.

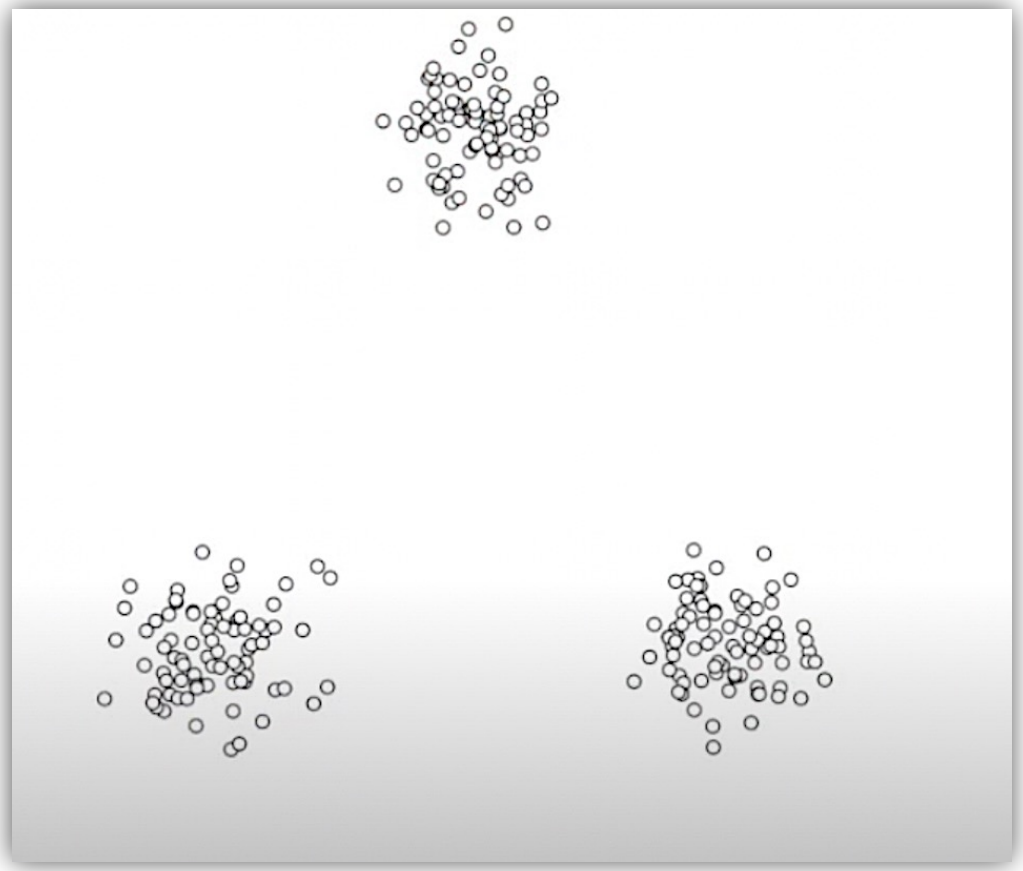
Clustering

- Data pre-processing (e.g. scaling) significantly impacts clustering results
- Features with larger numeric ranges dominate over those with smaller ranges, potentially skewing the clustering outcome

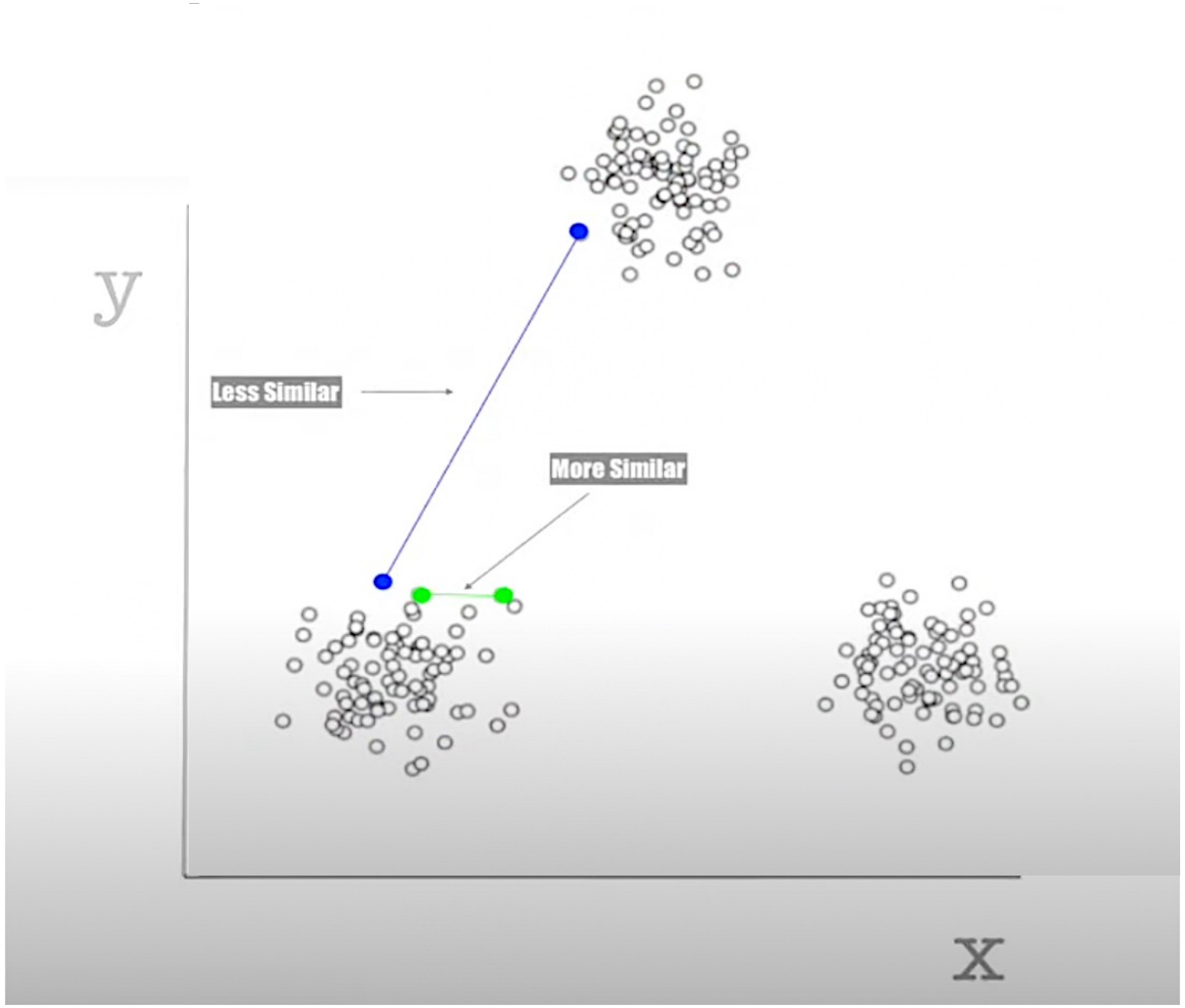


Clustering: K-means

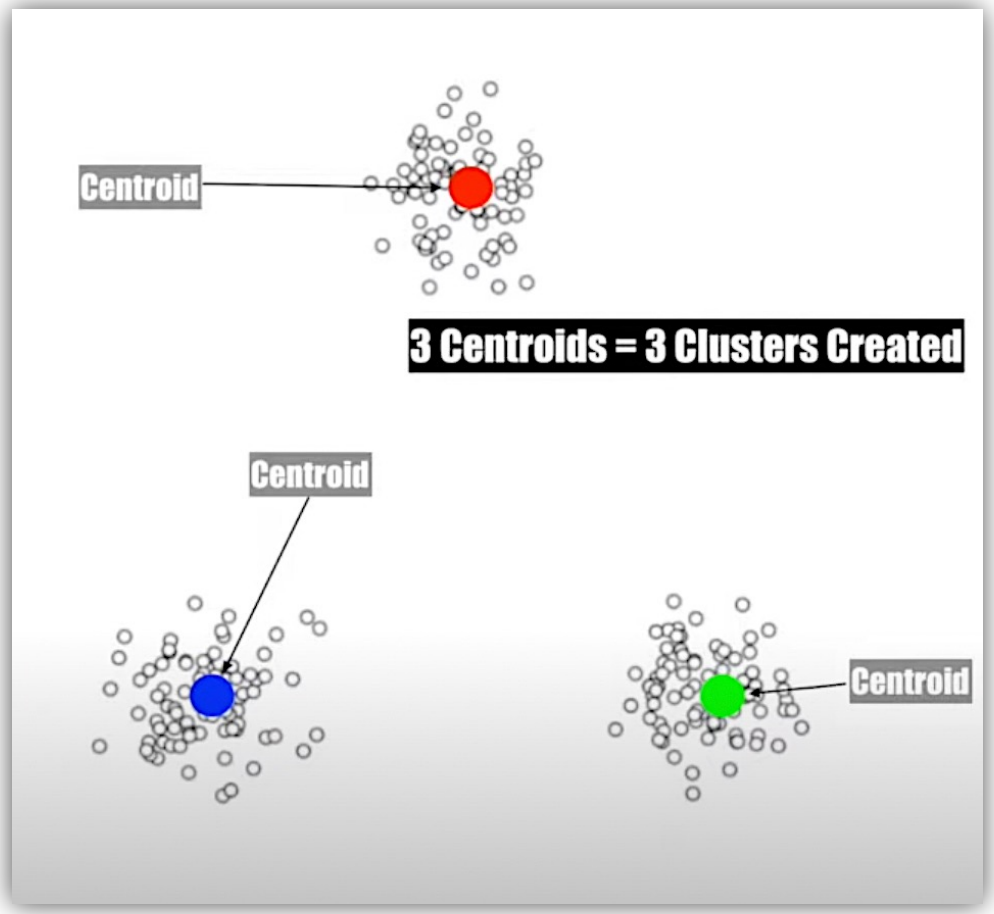
K-means uses Euclidean distance to form clusters of equal variance by minimizing the within-cluster sum of squares.



Clustering: K-means

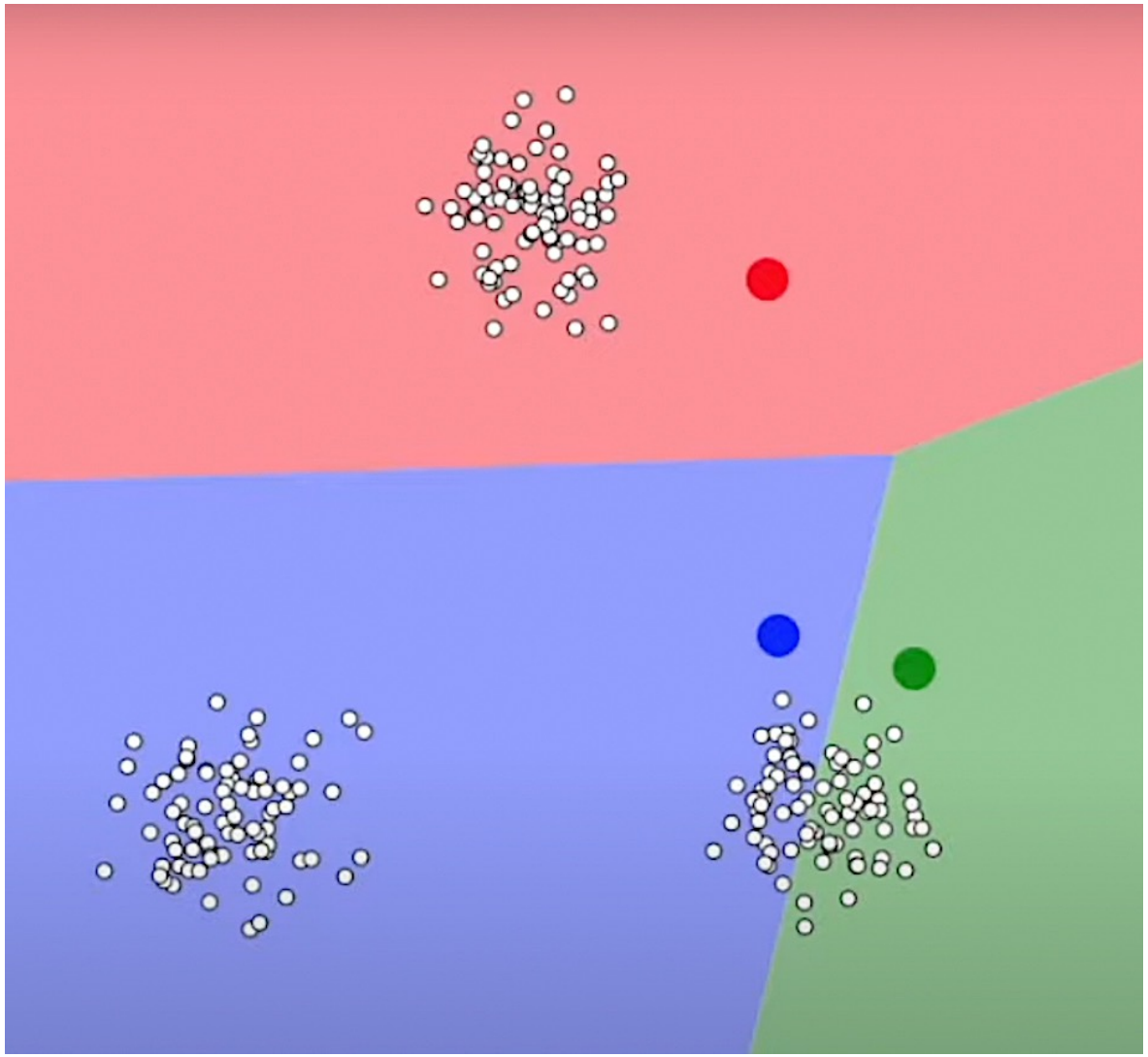


Clustering: K-means



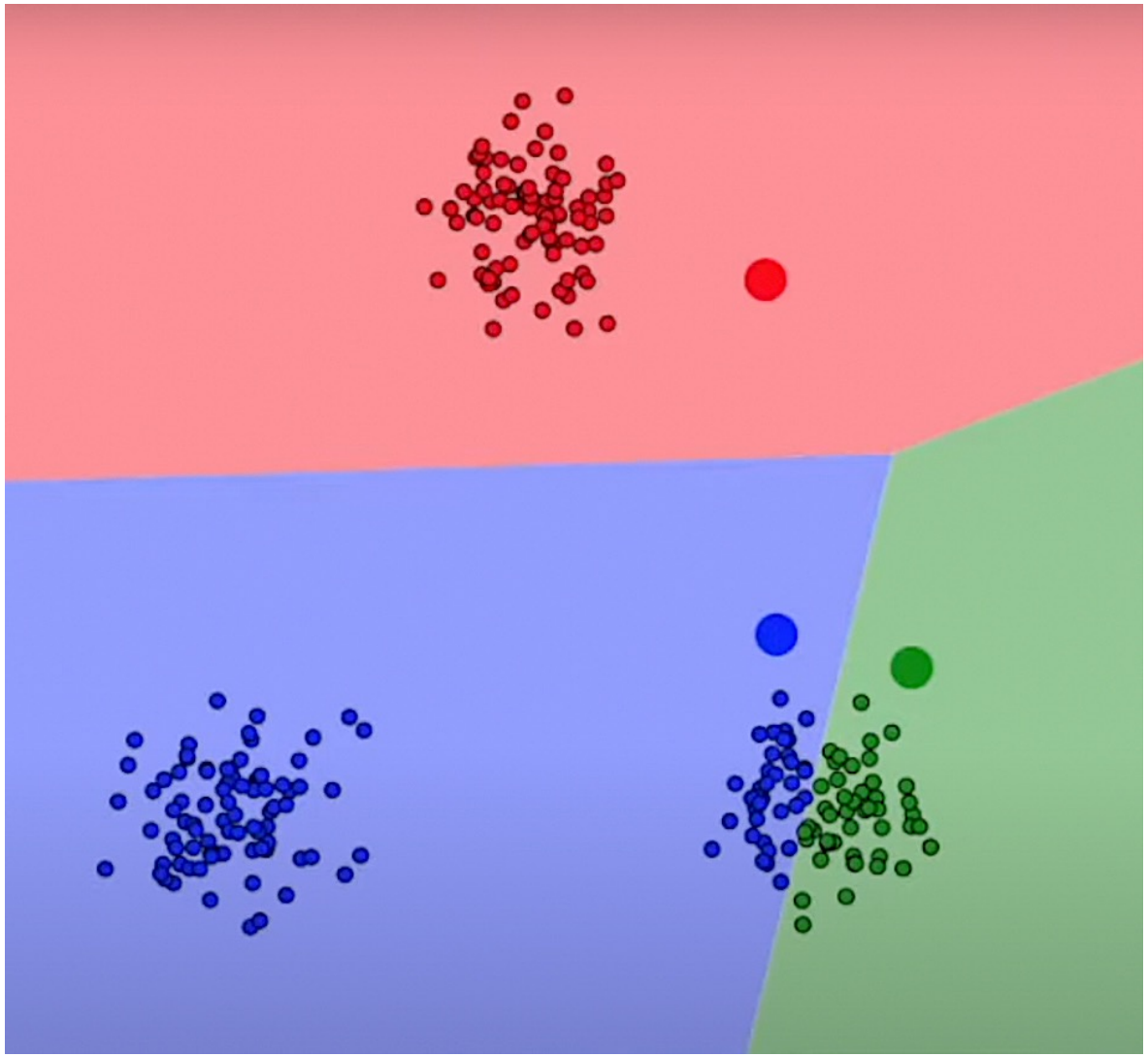
Clustering: K-means

1. Initialise the means according to K



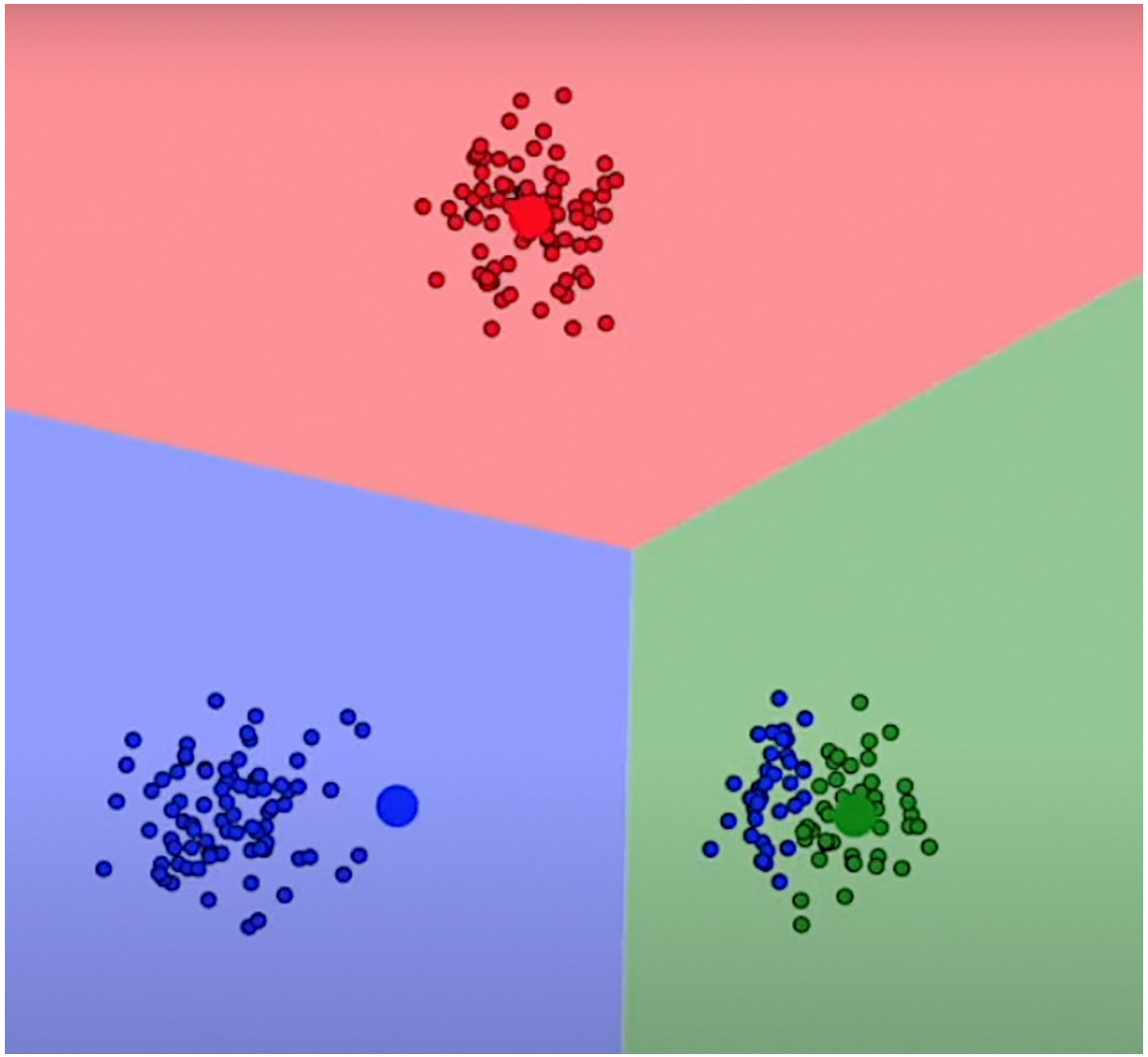
Clustering: K-means

2. Cluster n0 samples according to nearest mean



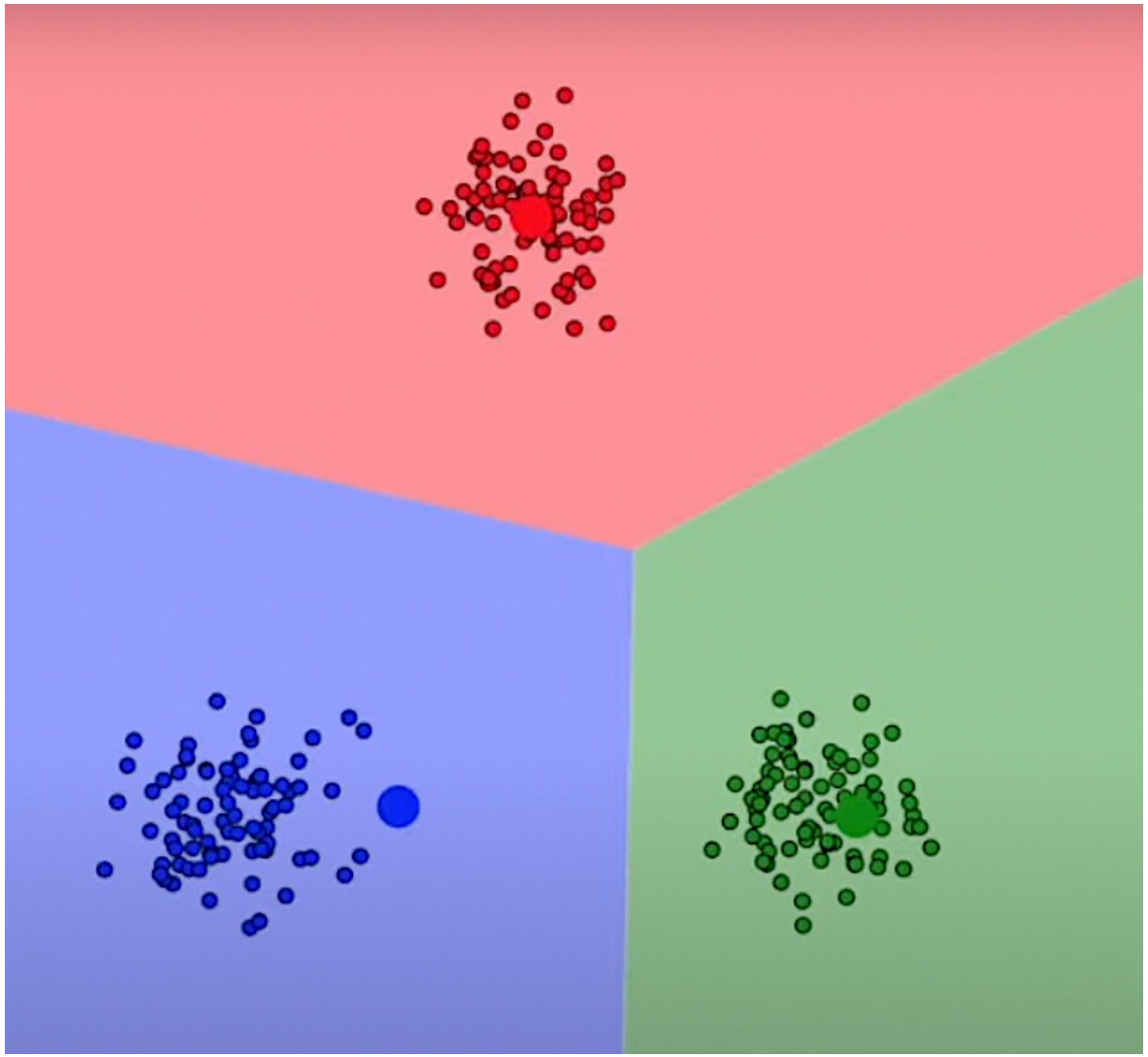
Clustering: K-means

1’. Recompute the mean



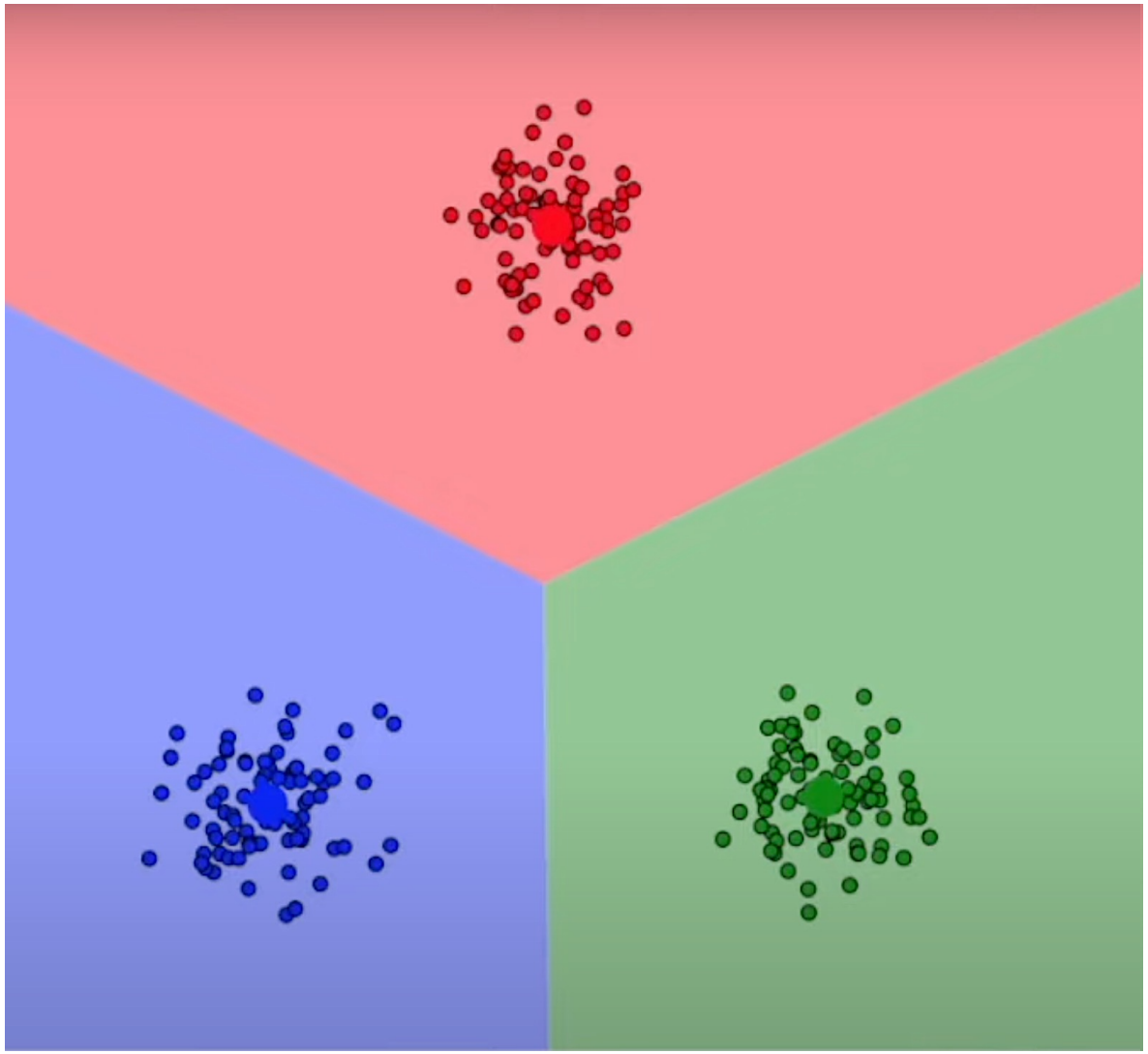
Clustering: K-means

2’. Cluster n1 samples according to nearest mean



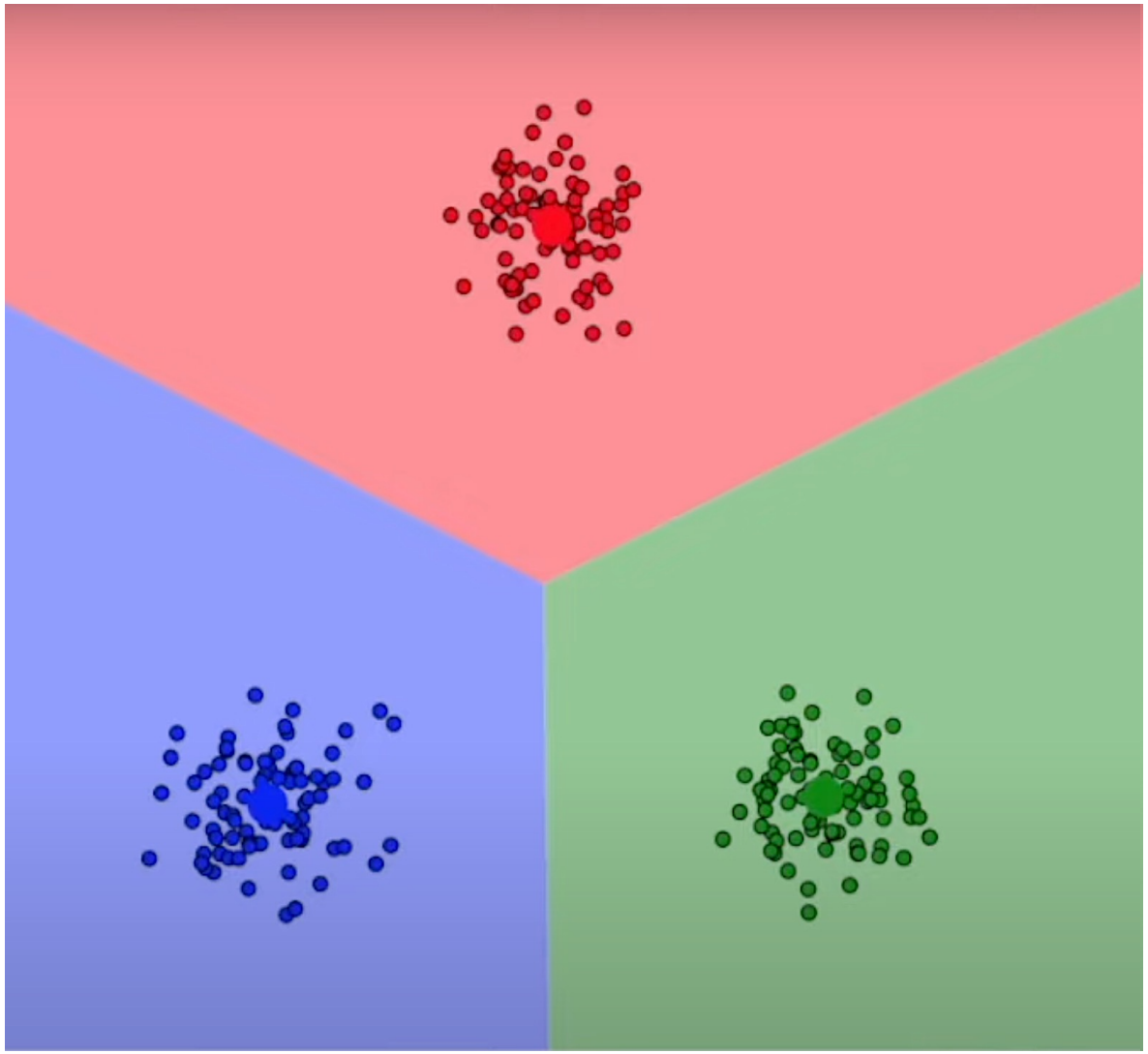
Clustering: K-means

1’’. Recompute the mean



Clustering: K-means

2’’. Cluster n2 samples according to nearest mean



Clustering: Image segmentation

- Clustering for image segmentation utilises spectral signatures to group pixels.
- Example: Crop classification
  - **Objective:** Precise mapping of different crop types (e.g. wheat, rice, maize) across vast areas.
  - **Method:** Pixels in satellite images are classified based on their reflectance properties in various electromagnetic spectrum bands.

Clustering: Image segmentation

- Clustering for image segmentation utilises spectral signatures to group pixels.
- Example: Crop classification
  - **Objective:** Precise mapping of different crop types (e.g. wheat, rice, maize) across vast areas.
  - **Method:** Pixels in satellite images are classified based on their reflectance properties in various electromagnetic spectrum bands.

Toy example

Original image

Clustering: Image segmentation

- Clustering for image segmentation utilises spectral signatures to group pixels.
- Example: Crop classification
  - **Objective:** Precise mapping of different crop types (e.g. wheat, rice, maize) across vast areas.
  - **Method:** Pixels in satellite images are classified based on their reflectance properties in various electromagnetic spectrum bands.

Toy example

Original image

RGB; K = 3

Clustering: Image segmentation

- Clustering for image segmentation utilises spectral signatures to group pixels.
- Example: Crop classification
  - **Objective:** Precise mapping of different crop types (e.g. wheat, rice, maize) across vast areas.
  - **Method:** Pixels in satellite images are classified based on their reflectance properties in various electromagnetic spectrum bands.

Toy example

Original image

Super pixels = group coherent regions locally

Red is localised

Clustering: K-means

Intra-cluster cohesion (compactness)

Inter-cluster separation (isolation)

- K?
  - **Elbow plot:** identifies the optimal number of clusters by finding the point where adding more clusters does not significantly decrease the within-cluster sum squares,
  - **Silhouette score:** evaluates cluster cohesion and separation; values closer to +1 indicate well-clustered data,
  - **Davies-Bouldin index:** evaluates cluster compactness and separation; lower values indicate better clustering,
  - **The gap statistic:** compares the within-cluster dispersion with that expected under a null reference distribution of the data, seeking the largest gap to determine the number of cluster,
  - **Domain knowledge:** subject-matter expertise to predefine the number of clusters based on known distinctions within the data, …

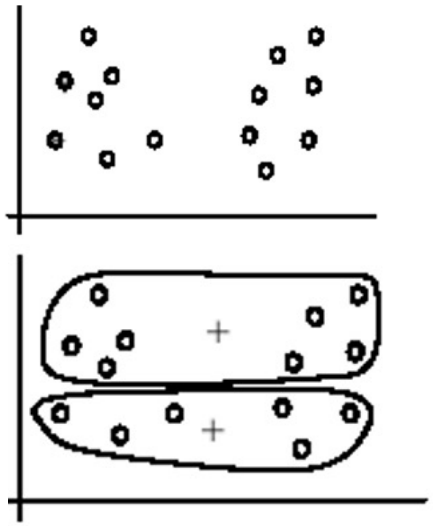
Clustering K-means

Pros

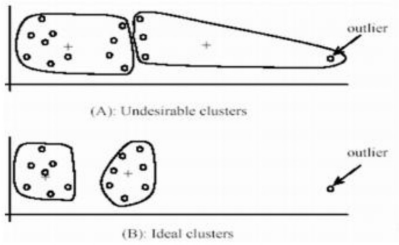
- Simple, fast to compute
- Easily interpretable
- converges to local minimum of within-cluster squared error

Cons

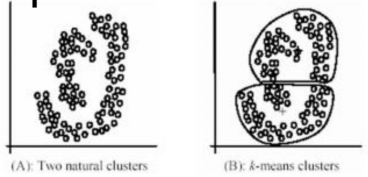
- Setting k?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters
- Assuming means can be computed



Outliers

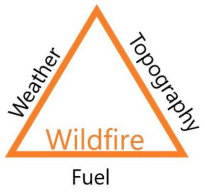


Spherical clusters



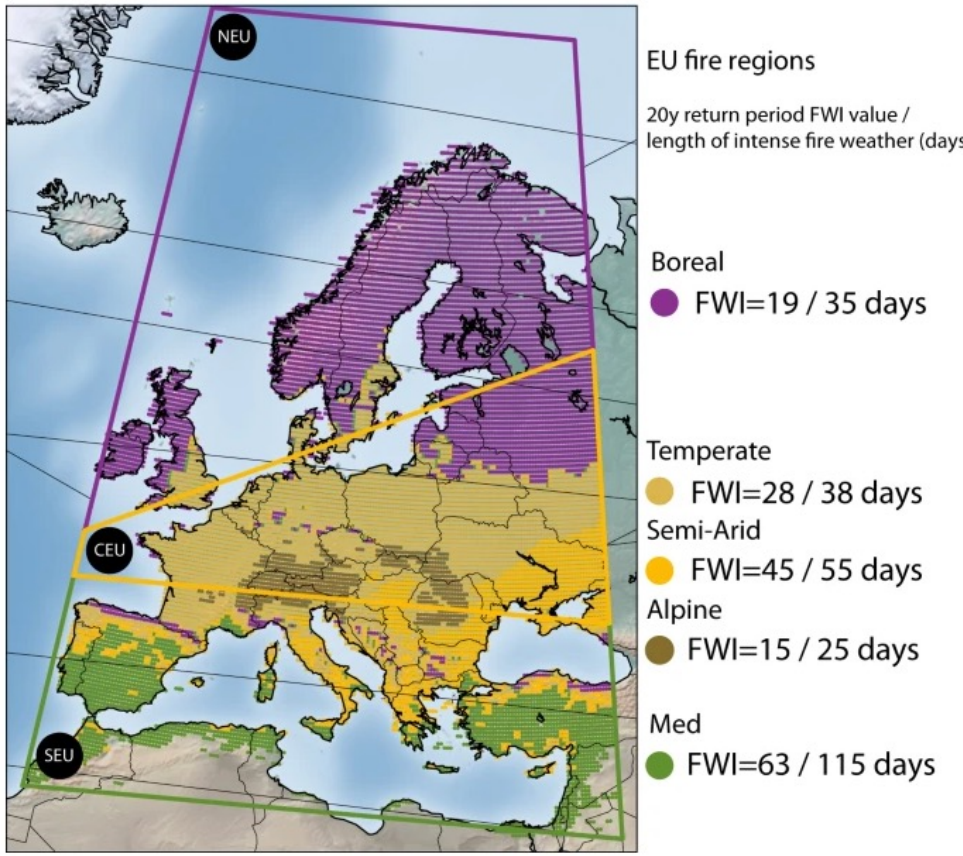
Clustering: K-Means

Europe faces up to tenfold increase in extreme fires in a warming climate, El Garroussi+, *npj climate and atmospheric science*, 2024 Doi:10.1038/s41612-024-00575-8



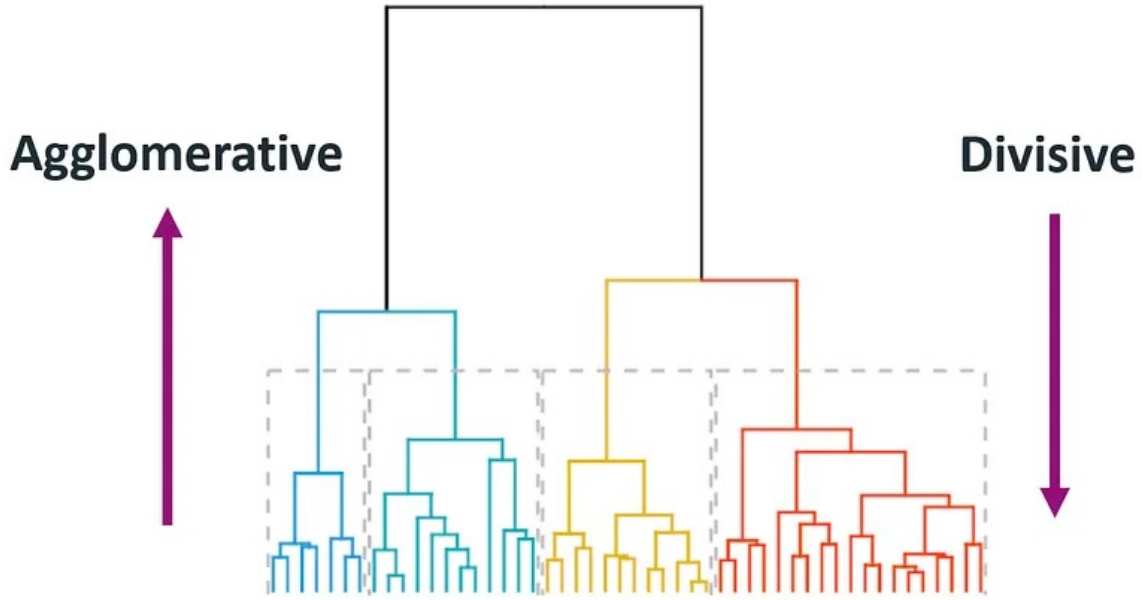
FWI
=
Dryness
+
Flammability

- K-Means was used to cluster different **fire regimes** based on Fire Weather Index (FWI) timeseries.



Clustering: Hierarchical clustering

- Produces a nested sequence of clusters, a tree, also called dendrogram.
- Agglomerative (bottom up) clustering: It builds the dendrogram from the bottom level.
- Divisive (top down) clustering: It starts with all data points in one cluster, the root.



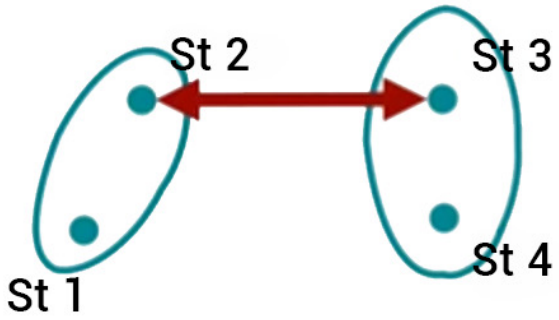
Hierarchical clustering

How do we determine the distance between two clusters?

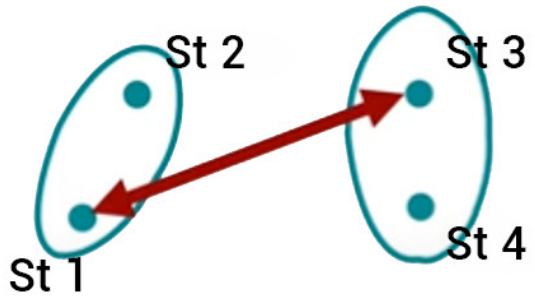
- Single-linkage: uses the distance between the closest elements in the cluster,
- Complete-linkage: uses the distance between the most distant elements of the cluster,
- And average-linkage: uses the average of all pairwise distances.

St = Observation station

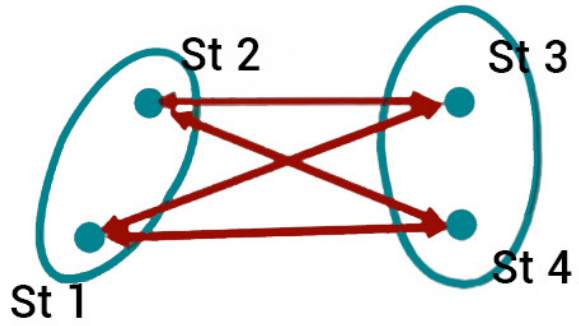
Single-linkage



Complete-linkage



Average-linkage



Hierarchical clustering

Pros

- Dendograms are great for visualisation
- Provides hierarchical relations between clusters
- Shown to be able to capture concentric clusters
- Stable

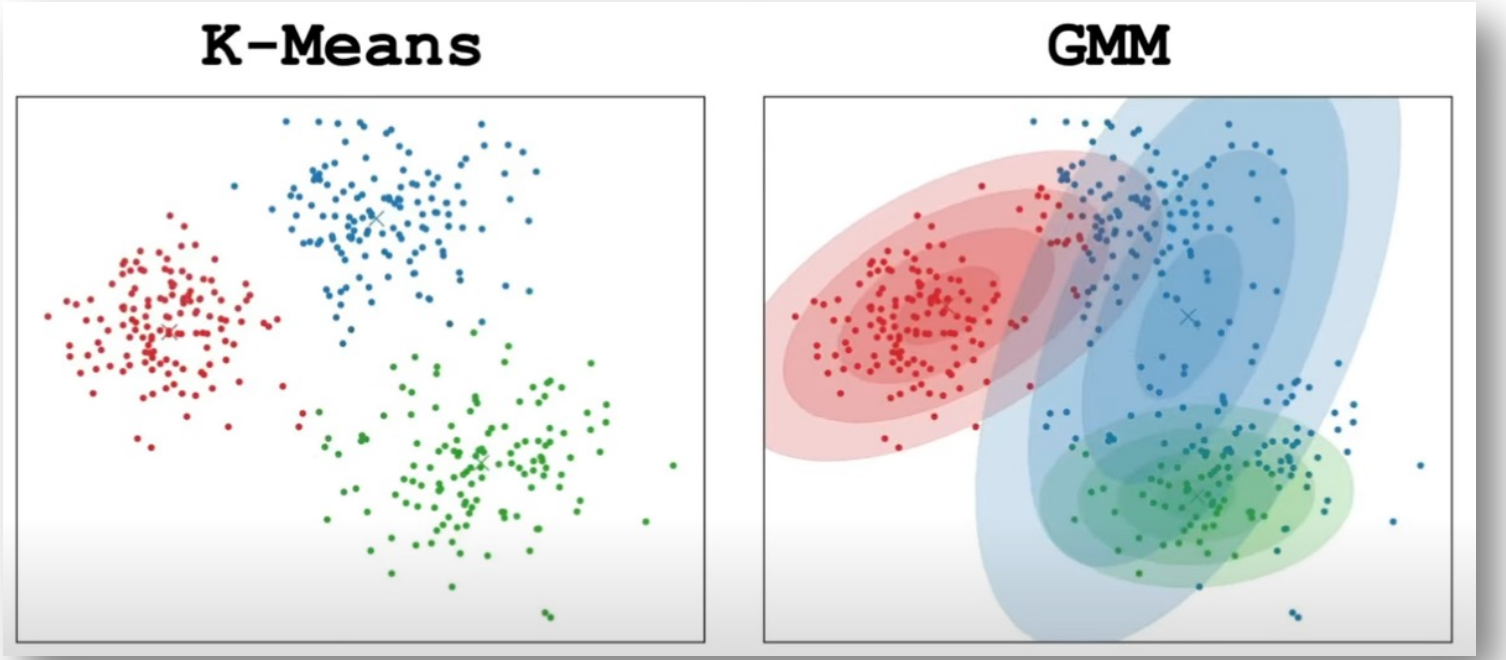
Cons

- Time complexity in large dataset
- Sensitive to outliers
- The choice of the distance metric (between two point and between two clusters) have a significant impact on the way clusters are generated
- Experiments showed that other clustering techniques outperform hierarchical clustering

Clustering: Gaussian mixture models

GMM: based on probability density estimation using gaussian mixture models and procedure called expectation maximisation (EM) to fit the model parameters.

- K-mean algorithm:
  - Assigned each example to exactly one cluster
  - What if clusters are overlapping?
  - What if cluster has a non-circular shape?
- Gaussian mixture models:
  - Clusters modelled as Gaussians
    - Not just by their means
  - Maximising the likelihood of observed data using EM procedure
  - Gives probability model of x; generative.



blue

Hard clustering

70% blue, 20% green,10% red

Soft clustering

Clustering: Mixture of Gaussians

- Start with parameters describing each cluster
- Mean μ_c, variance σ_c, "size" π_c
- Probability distribution: $p(x) = \sum_c \pi_c \mathcal{N}(x\ ;\ \mu_c, \sigma_c)$

Clustering: Mixture of Gaussians

- Start with parameters describing each cluster
- Mean μ_c, variance σ_c, "size" π_c
- Probability distribution: $p(x) = \sum_c \pi_c \mathcal{N}(x\ ;\ \mu_c, \sigma_c)$

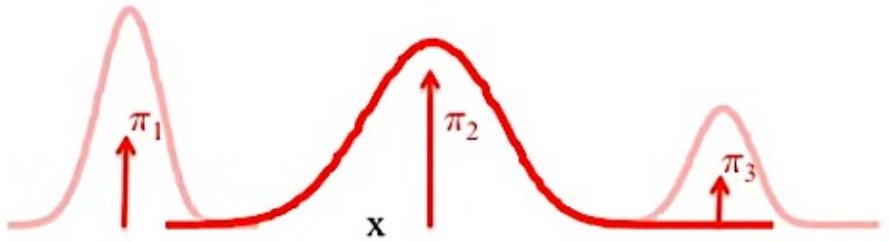
- Equivalent "latent variable" form:

$p(z = c) = \pi_c$ Select a mixture component with probability π

$p(x|z = c) = \mathcal{N}(x\ ;\ \mu_c, \sigma_c)$ Sample from that component's Gaussian

"Latent assignment" z:
we observe x, but z is hidden

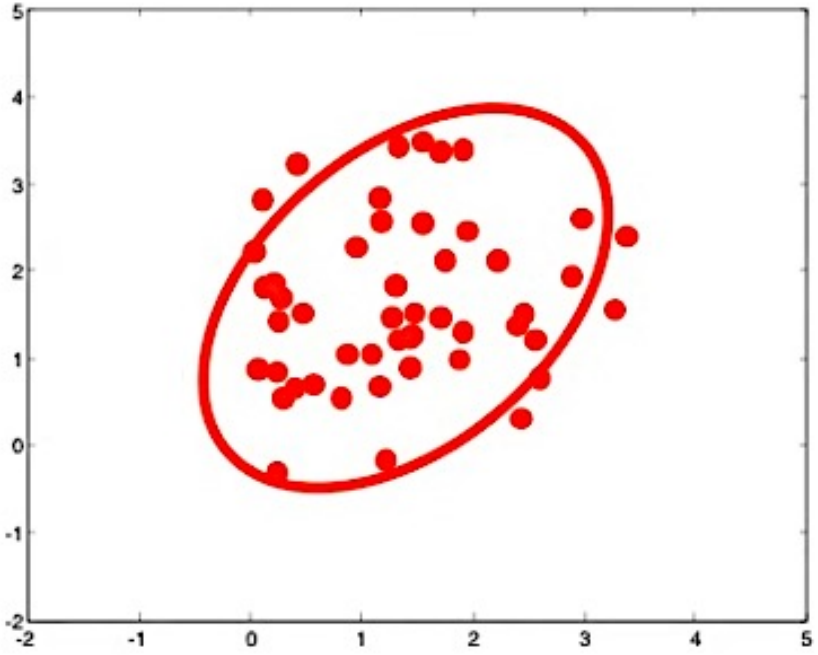
p(x) = marginal over x



Clustering: Gaussian mixture models

We use multivariate Gaussian models

$$\mathcal{N}(\underline{x}\ ;\ \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$



Maximum Likelihood estimates

$$\hat{\mu} = \frac{1}{m} \sum_i x^{(i)}$$

$$\hat{\Sigma} = \frac{1}{m} \sum_i (x^{(i)} - \hat{\mu})^T (x^{(i)} - \hat{\mu})$$

m: nb features

We'll model each cluster using one of these Gaussian "bells"…

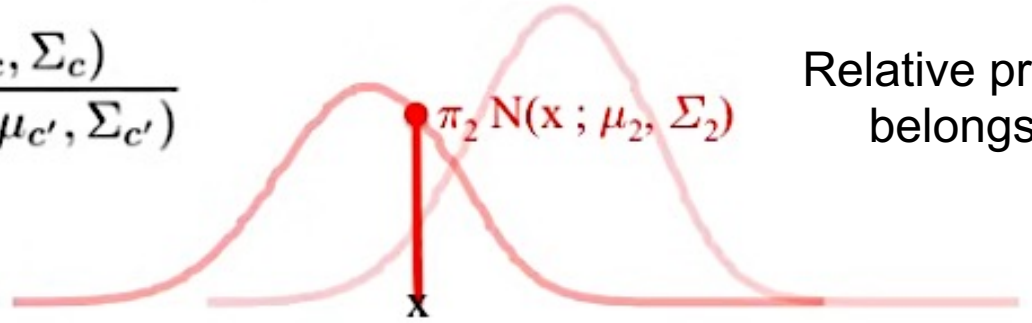
Clustering: Mixture of Gaussians

- Start with clusters: Mean μ_c, Covariance Σ_c, "size" π_c

- E-step ("Expectation")
  - For each datum (example) x_i,
  - Compute "r_{ic}", the probability that it belongs to cluster c
    - Compute its probability under model c
    - Normalize to sum to one (over clusters c)

EM Algorithm: E-step

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$



Relative probability that x_i belongs to cluster C

  - If x_i is very likely under the c^{th} Gaussian, it gets high weight
  - Denominator just makes r's sum to one

Clustering: Mixture of Gaussians

- Start with assignment probabilities r_{ic}
- Update parameters: mean μ_c, Covariance Σ_c, "size" π_c

- M-step ("Maximization")
  - For each cluster (Gaussian) z = c,
  - Update its parameters using the (weighted) data points

EM Algorithm: M-step

$$m_c = \sum_i r_{ic}$$ Total responsibility allocated to cluster c

$$\pi_c = \frac{m_c}{m}$$ Fraction of total assigned to cluster c

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)}$$ Weighted mean of assigned data

$$\Sigma_c = \frac{1}{m_c} \sum_i r_{ic} (x^{(i)} - \mu_c)^T (x^{(i)} - \mu_c)$$ Weighted covariance of assigned data (use new weighted means here)

Clustering: Mixture of Gaussians

- Each step increases the log-likelihood of our model

$$\log p(\underline{X}) = \sum_i \log \left[\sum_c \pi_c \, \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c) \right]$$

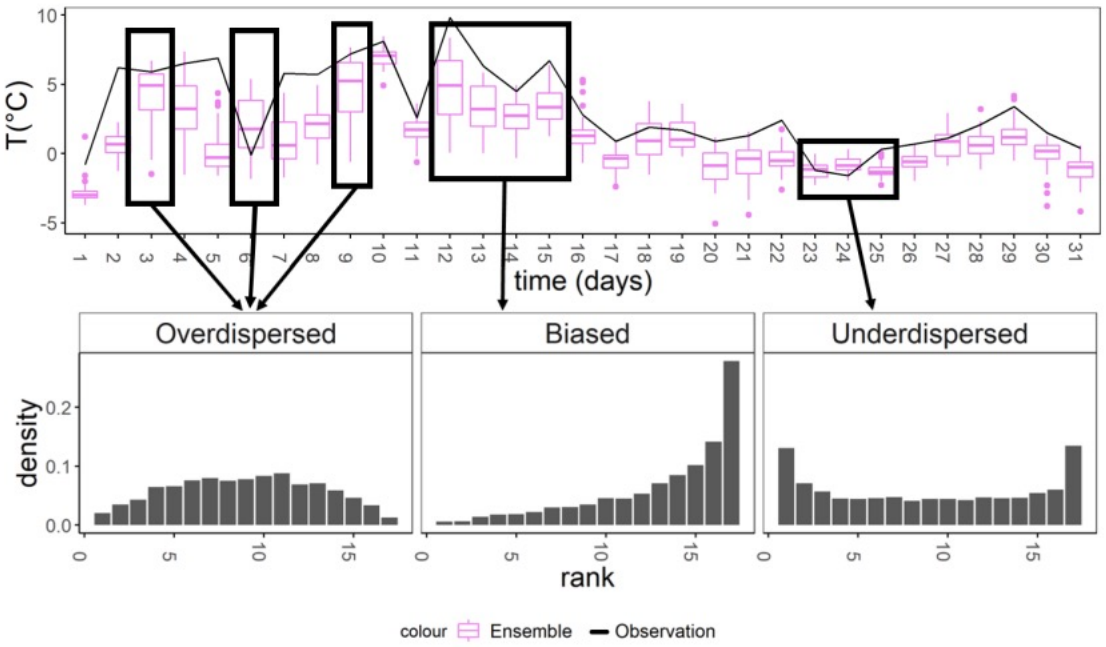
- Iterate until convergence
  - Convergence guaranteed
  - Local optima: initialization often important
- What should we do
  - If we want to choose a single cluster for an "answer"? Question
  - With new data we didn't see during training?
- Choosing the number of clusters
  - Can use penalized likelihood of training data
  - True probability model: can use log-likelihood of test data, log p(x')

Clustering: GMM

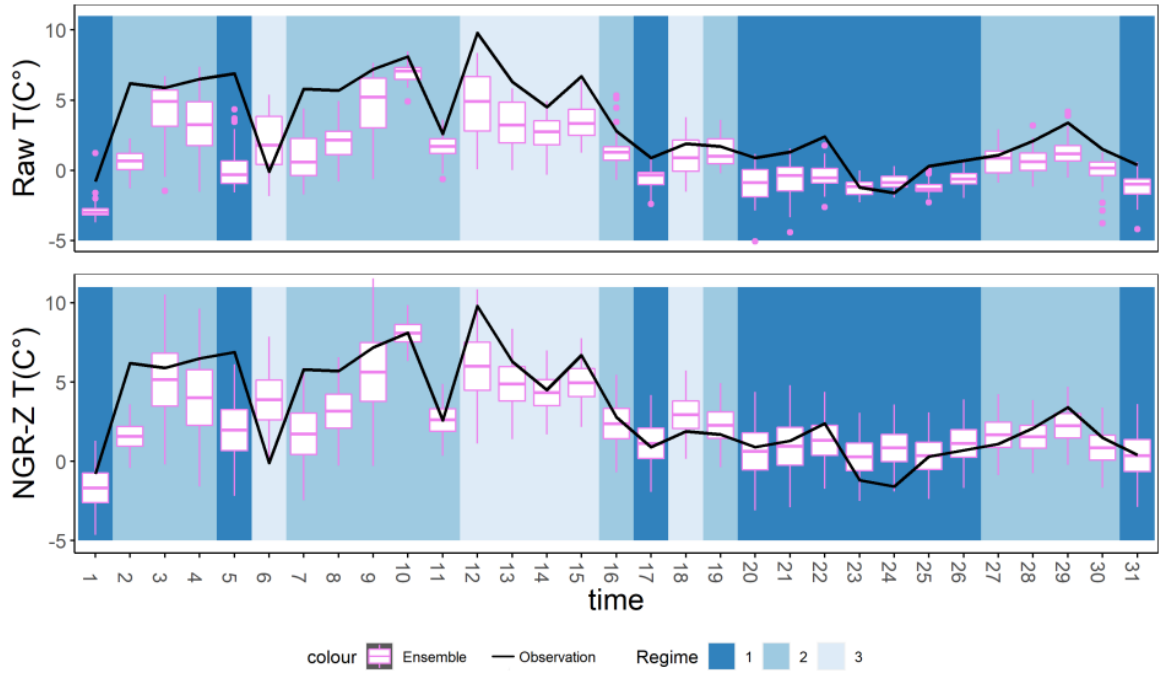
Gaussian mixture models for clustering and calibration of ensemble weather forecasts, Jouan+, *Discrete and Continuous Dynamical Systems – S*, Feb 2023
Doi: 10.3934/dcdss.2022037

- Ensemble forecasts may suffer from bias and under/over dispersion errors that need to be corrected.
- GMM were used to identify clusters which correspond to different types of distribution errors.
- A calibration method (Non-homogeneous Gaussian Regression) was then applied cluster by cluster to correct ensemble forecast distributions.

Temperature at Millau, January 2015, 6pm.



ECMWF ens, 3 days horizon



How to choose a clustering algorithm?

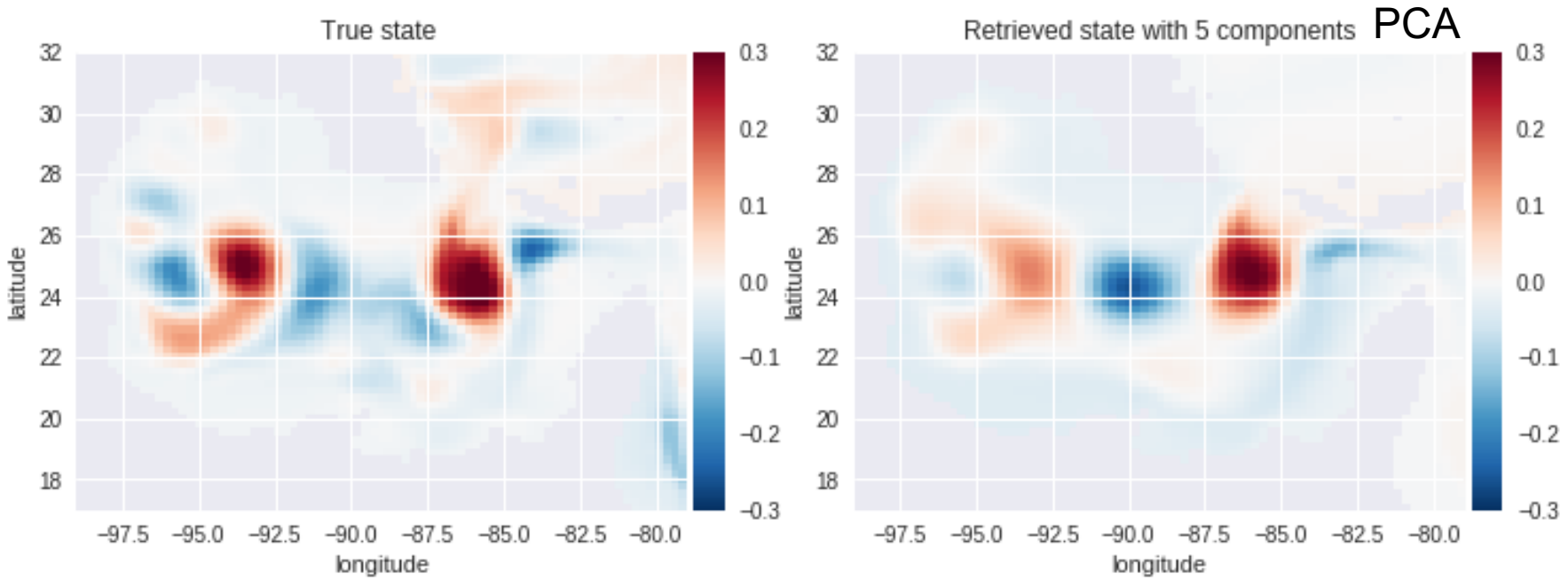
- **Choosing the "best" algorithm is a challenge**
- The common practice is to:
  - run several algorithms using different distance functions and parameter settings, and carefully analyse and compare the results.
- The interpretation of the results must be based on insight into the meaning of the original data together with knowledge of the algorithm used.
- Clustering is highly application dependent and to certain extent subjective.
- Cluster evaluation is a hard problem:
  - We don't know the correct clusters
  - Some methods are used like:
    - centroids and spreads analysis,
    - each class is a cluster -> confusion matrix is constructed -> entropy, purity, precision, recall and F-score.
  - In some applications, clustering is not the primary task, but used to help perform another task -> we can use the performance on the primary task to compare clustering methods.

Dimensionality reduction

Dimensionality reduction techniques are used to reduce the number of features in a dataset while preserving as much of the relevant information as possible. This is useful for reducing the computational complexity of models and for visualising high-dimensional data.

- It is a process of converting data set having vast dimensions into a dataset with lesser dimensions,
- It ensures that converted data set conveys similar information concisely.

Example: Reconstructing sea surface height (SSH) using 5 components (5D), single snapshot

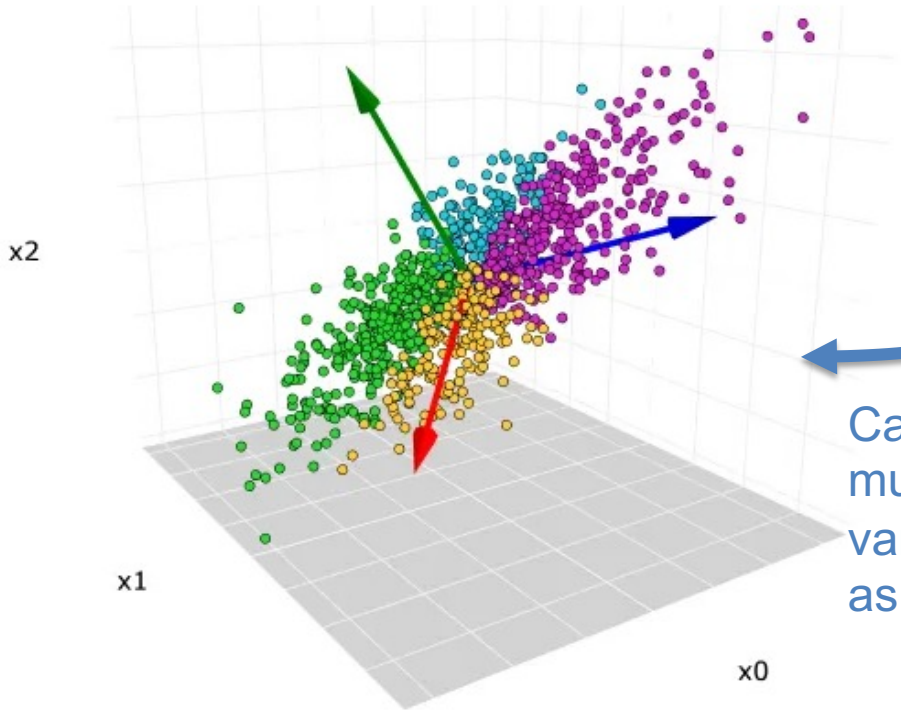


SSH (lat, lon, time)
time compression:
from 518 to 5

Dimensionality reduction: Principal component analysis

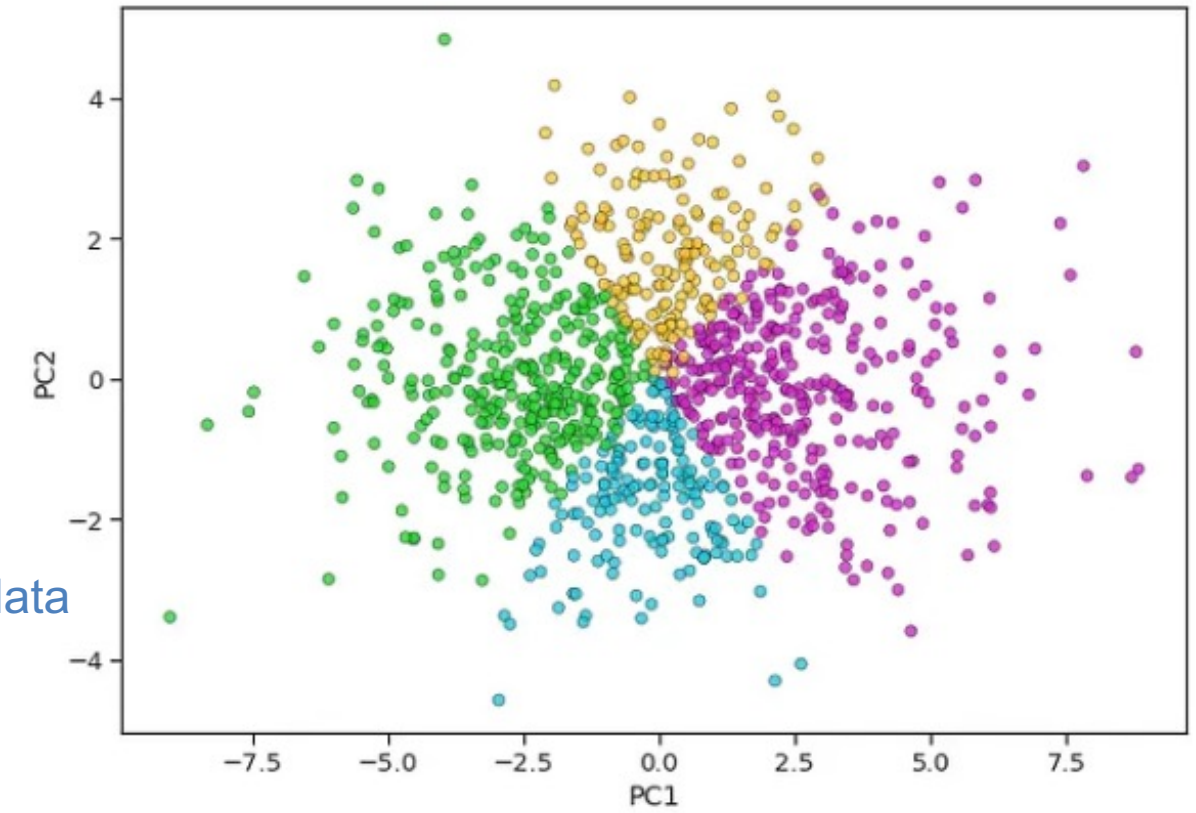
- Relies on the covariance between the features to determine the lower dimensional space.
- Transform large number of variables into a smaller number of uncorrelated variables called principal components (PCs).
- Fourier (frequency patterns) and wavelet (time and frequency) compression is similar.

3D



Capture as much of the variation in data as possible

2D



Dimensionality reduction: Principal component analysis

Algorithm: PCA

1. Normalize data
2. Calculate the covariance matrix : $C_y = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^T$
3. Solve : $\begin{cases} \lambda v = C_y v \\ \|v\|_{\mathbb{R}^n} = 1 \end{cases}$
4. Project on the k^{th} principal component : $y_{pc}^k = \langle v^k, y \rangle_{\mathbb{R}^n}$

(Scikit Learn (Pedregosa et al., 2011))

Dimensionality reduction: Principal component analysis

Eigenvector?

- Vectors **x** having same direction as A**x** are called *eigenvectors* of A (A is an n by n matrix).
- In the equation A**x**=λ**x**, λ is called an *eigenvalue* of A.

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} x \begin{pmatrix} 3 \\ 2 \end{pmatrix} \square \begin{pmatrix} 12 \\ 8 \end{pmatrix} \square 4x \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Dimensionality reduction: Principal component analysis

- $A\mathbf{x}=\lambda\mathbf{x} \Leftrightarrow (A-\lambda I)\mathbf{x}=0$
- How to calculate **x** and λ:
    1. Calculate $det(A-\lambda I)$, yields a polynomial (degree n)
    2. Determine roots to $det(A-\lambda I)=0$, roots are eigenvalues λ
    3. Solve $(A-\lambda I)\,\mathbf{x}=0$ for each λ to obtain eigenvectors **x**

Dimensionality reduction: Principal component analysis

Principal components

1. principal component (PC1)

   - The eigenvalue with the largest absolute value will indicate that the data have the largest variance along its eigenvector, the direction along which there is greatest variation

2. principal component (PC2)

   - the direction with maximum variation left in data, orthogonal to PC1.

   .

   .

   .

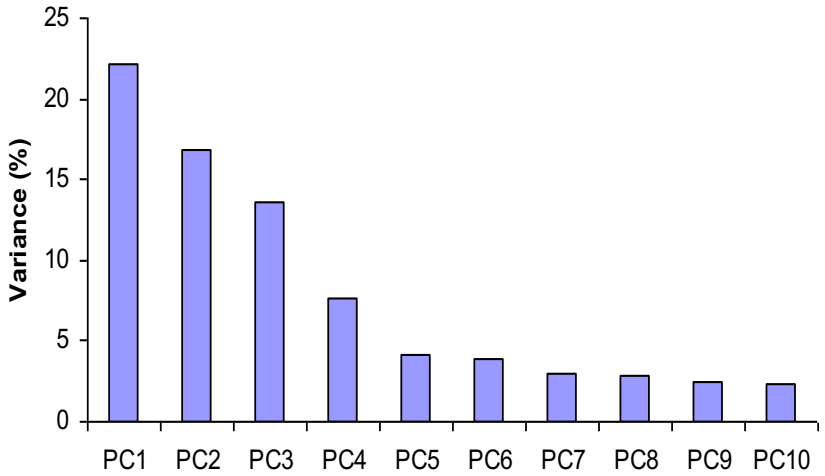
In general, only few directions manage to capture most of the variability in the data.

Dimensionality reduction: Principal component analysis

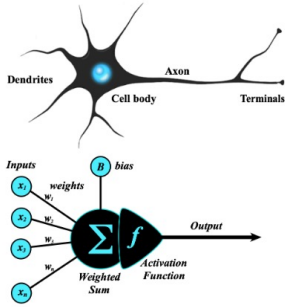
Eigenvalues

- Calculate eigenvalues λ and eigenvectors **x** for covariance matrix:
  - Eigenvalues λ_j are used for calculation of [% of total variance] (V_j) for each component *j*:

$$V_j = 100 \cdot \frac{\lambda_j}{\sum_{x=1}^{n} \lambda_x} \qquad \sum_{x=1}^{n} \lambda_x = n$$

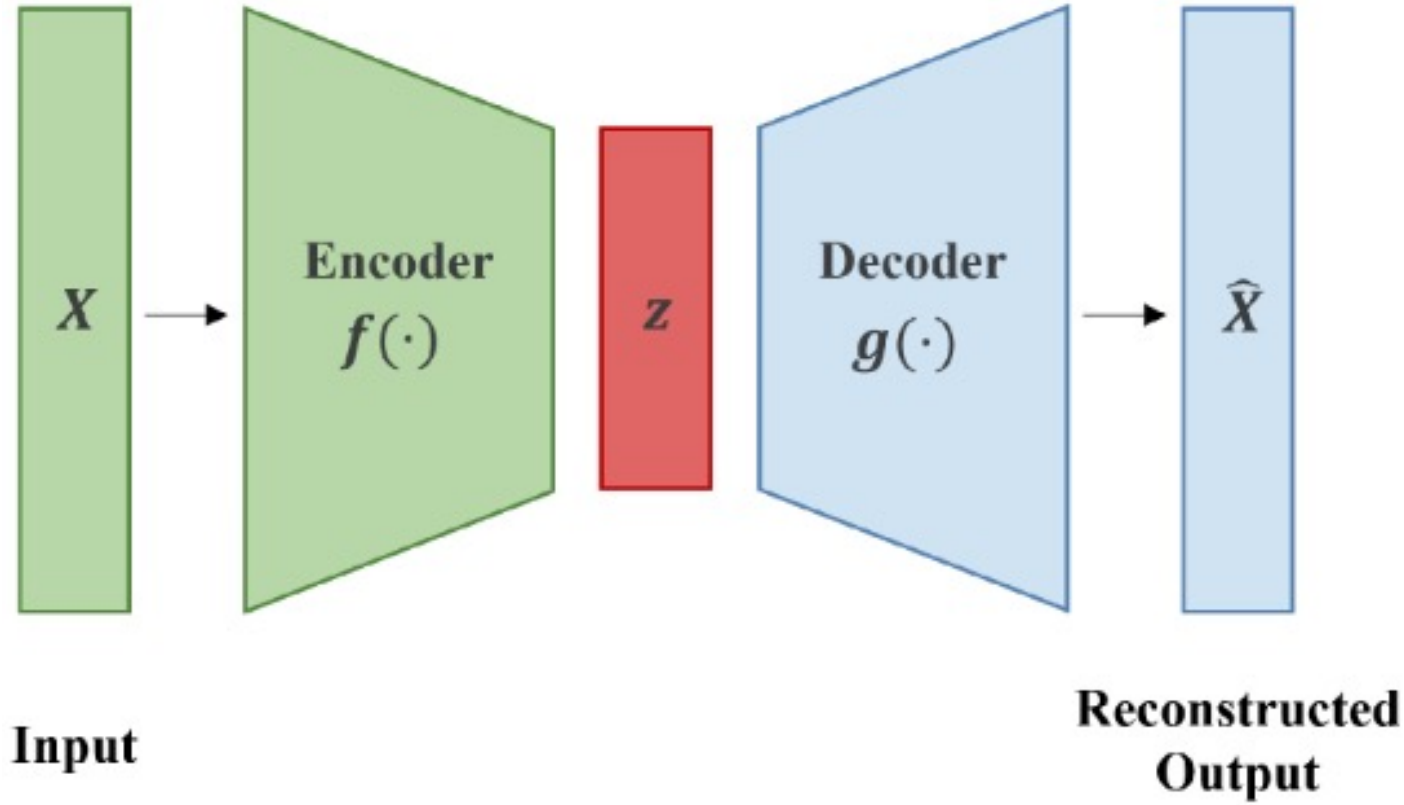


Dimensionality reduction: Autoencoder



Further details in tomorrow's lecture

- Autoencoder (AE) learns to compress (encode) the input data into a lower-dimensional representation and then reconstruct (decode) it back to its original form.
- This process forces the autoencoder to capture the most important features in the compressed representation -> effective for feature extraction and data compression.
- Adding non-linearity (such as nonlinear activation functions and more hidden layers) -> powerful representations of X with less information loss.



Undercomplete autoencoder: dim(Z) < dim(X)

Variational autoencoder

- Variational autoencoders (VAE) are generative models that learn compressed representations of their training data as **probability distributions**.

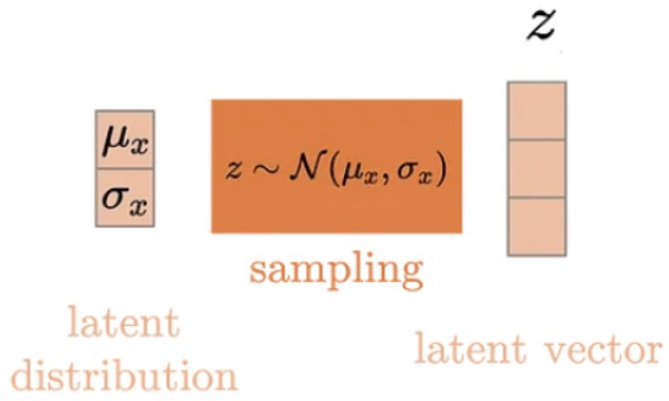
-> Generative AI: VAE generate new sample data by creating variations of those learned representations.

AE

discrete latent space
Z: single encoding vector
deterministic encoding

VAE

continuous latent space
μ vector of means + σ vector of standard deviations
stochastic encoding



Variational autoencoder

Loss function

$$\text{reconstruction loss} = \|x - \hat{x}\|_2$$

$$\textit{similarity loss} = \textit{KL Divergence} = D_{KL}(\mathcal{N}(\mu_x, \sigma_x) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I}))$$

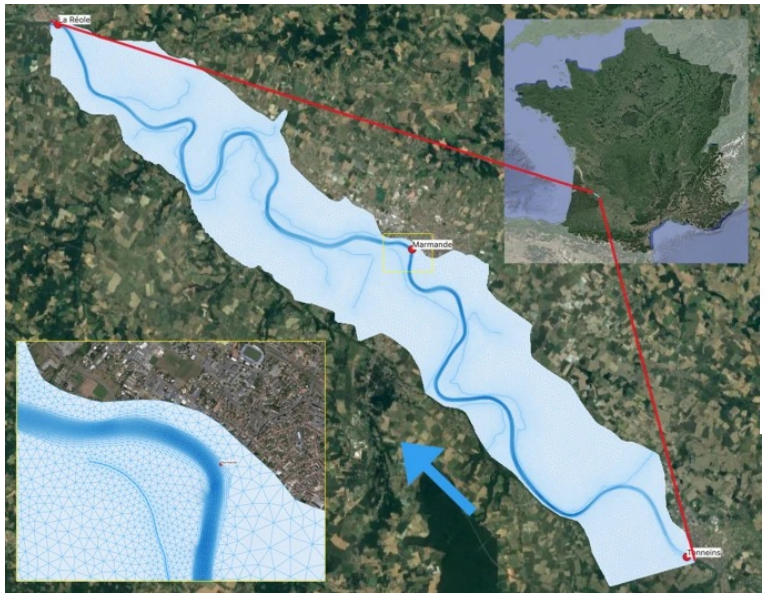
$$loss = reconstruction\ loss + similarity\ loss$$

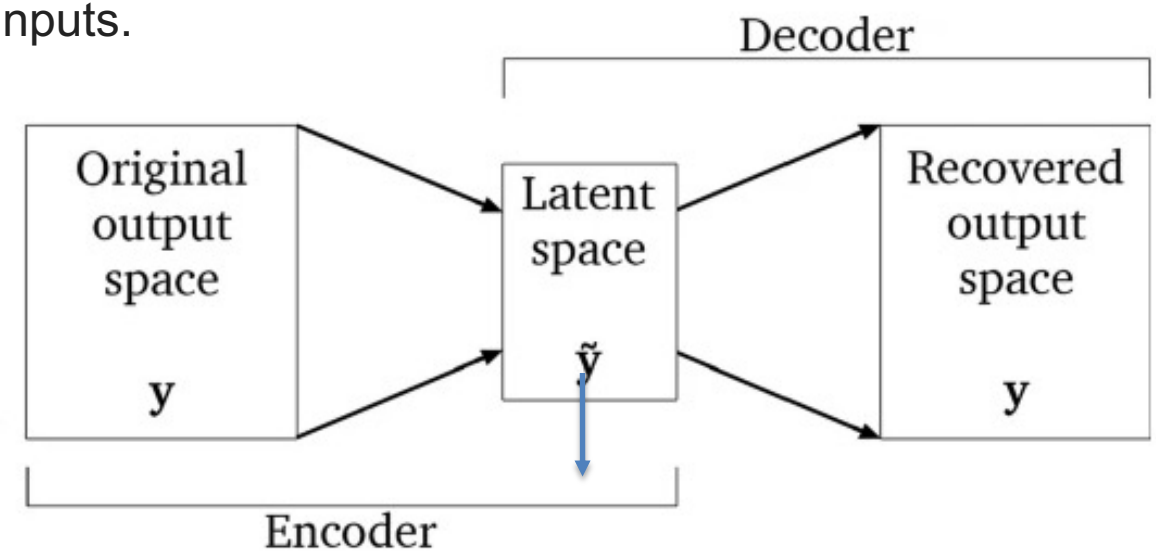
The loss function is defined by the VAE objectives. VAE has two objectives:
1. Reconstruct the input
2. Latent space should be normally distributed

Unsupervised learning as an input for supervised learning: Flood forecast

Tackling random fields non-linearities with unsupervised clustering of polynomial chaos expansion in latent space: application to global sensitivity analysis of river flooding, El Garroussi+, *SERRA*, 2021

- Predict the evolution of a flood event:
  - AE used to reduce the multi-dimensional hydraulic model output (water level),
  - Automatically partition the input space into clusters that are not affected by non-linearities and support accurate regression models.
  - Divide-and-conquer principle: a mixture of local regression models is proposed to deal with non-linearity in the hydraulic state with respect to hydraulic inputs.





- Clustering using GMM,
- A supervised classification model is then built within each cluster,
- A regression model is built inside each class.

Challenges in unsupervised learning

- Determining the number of clusters in a dataset,
- Dealing with high-dimensional data,
- Interpreting the results without predefined labels.
- The risk of finding patterns that don't generalize well to new data can lead to overfitting.
- The absence of ground truth makes it difficult to evaluate the model's performance accurately.

Future of unsupervised learning

- Advancements in algorithm efficiency, interpretability, and integration with supervised and reinforcement learning for more comprehensive models.
- Innovations in deep learning, such as improved autoencoders and generative adversial networks, will unlock new capabilities in data generation, anomaly detection, and complex pattern recognition across various domains.

Thank you for your attention

Unsupervised learning for data exploration

Navigating data's hidden patterns

Siham El Garroussi

ECMWF

siham.garroussi@ecmwf.int

ECMWF