Data Science for Smart Cities

CE88

Prof: Alexei Pozdnukhov
GSI: Madeleine Sheehan

115 McLaughlin Hall

alexeip@berkeley.edu
m.sheehan@berkeley.edu
CE88 in title

Today

Data exploration: clustering

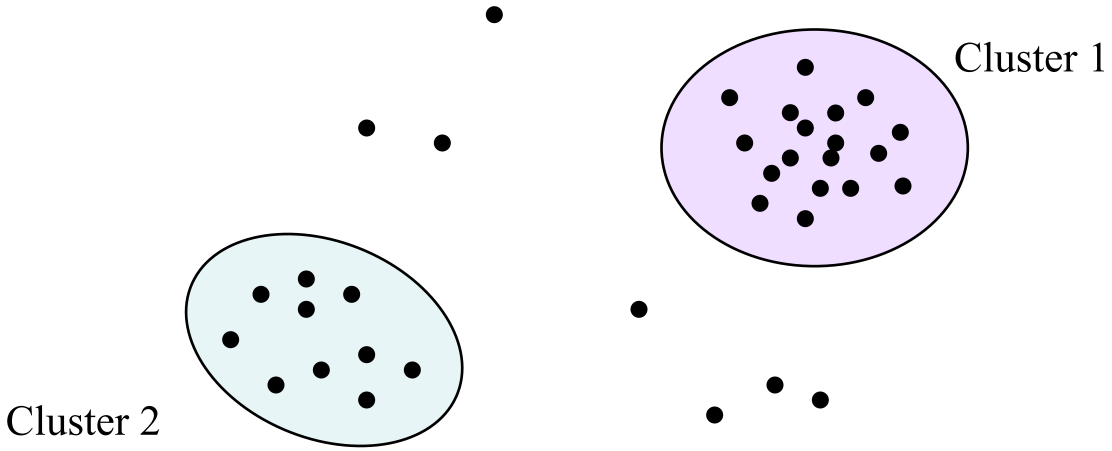
Minilab 12

Towards the final project (optional)

What is Cluster Analysis?

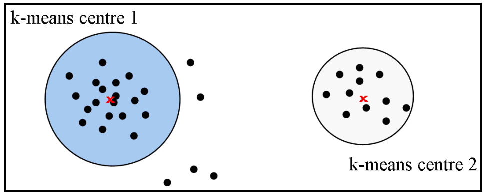
- Cluster: a collection of data objects
  - **Similar** to the objects in the same cluster (intraclass similarity)
  - **Dissimilar** to the objects in other clusters (interclass dissimilarity)
- Cluster analysis
  - Statistical/geometrical method for grouping a set of data objects into clusters
  - A good clustering method produces high quality clusters with high intraclass similarity and low interclass similarity
- Clustering is unsupervised classification
- Can be a stand-alone tool or as a preprocessing step for other algorithms

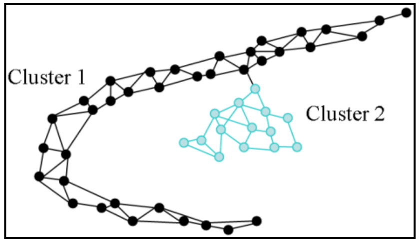
What is Cluster Analysis?



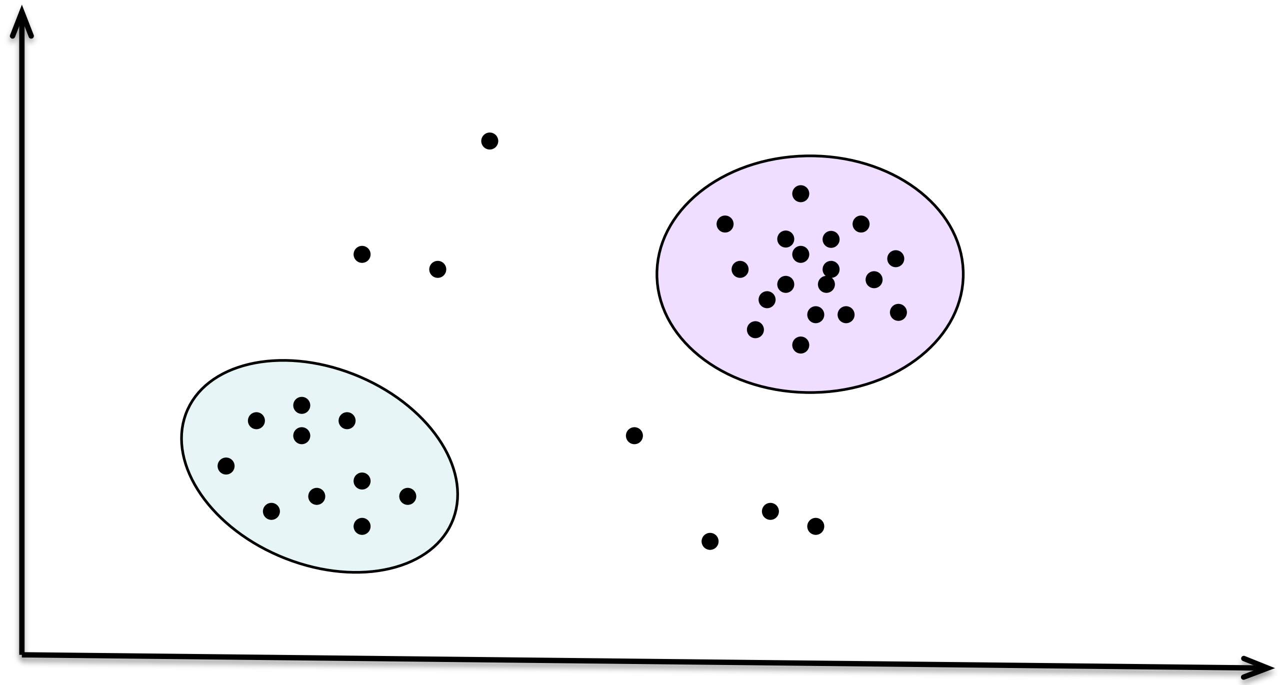
Requirements for Clustering

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal domain knowledge required to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to order of input records
- Robustness wrt high dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability



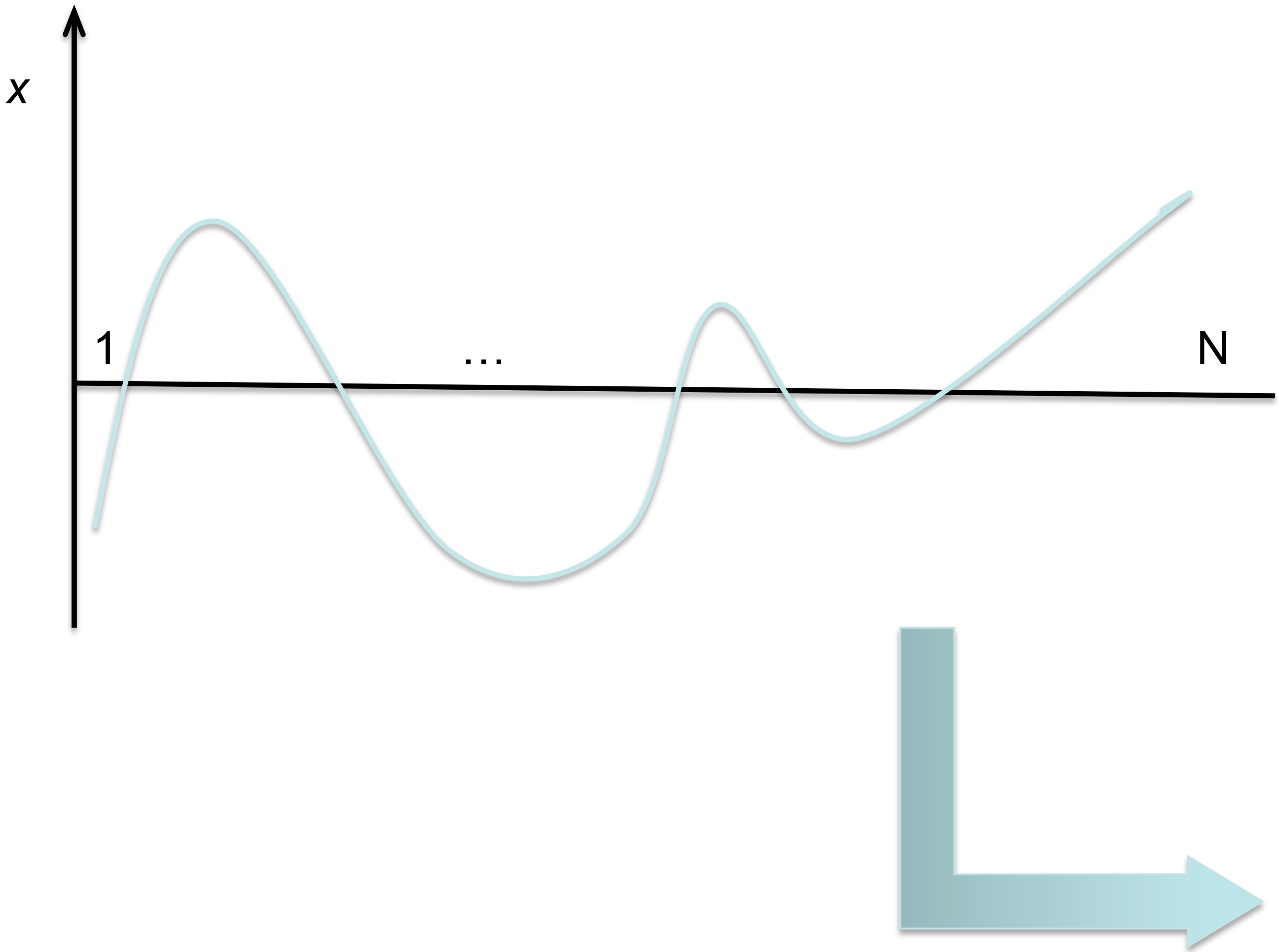


Data representation



$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

Data representation



$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

Data representation

[[x_1 x_2 . . .   ]

[x_{32} x_{33} . . .]]

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

Similarity measures

- Euclidean Distance

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{n=1}^{N} (x_n - y_n)^2}$$

Similarity measures

- Cosine similarity

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \quad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$C_{\text{cosine}}(\vec{x}, \vec{y}) = \frac{\frac{1}{N}\sum_{i=1}^{N} x_i \times y_i}{\|\vec{x}\| \times \|\vec{y}\|}$$

$\vec{x} = \vec{y}$ +1 ≥ Cosine Correlation ≥ – 1 $\vec{x} = -\vec{y}$

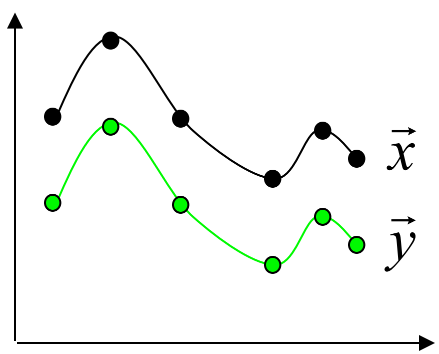
Similarity measures

- Pearson Correlation

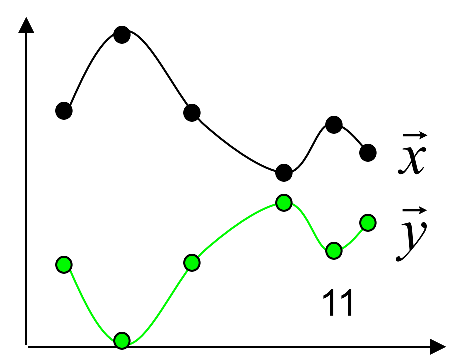
$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$C_{pearson}(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{N} (x_i - m_x)(y_i - m_y)}{\sqrt{[\sum_{i=1}^{N} (x_i - m_x)^2][\sum_{i=1}^{N} (y_i - m_y)^2]}}$$

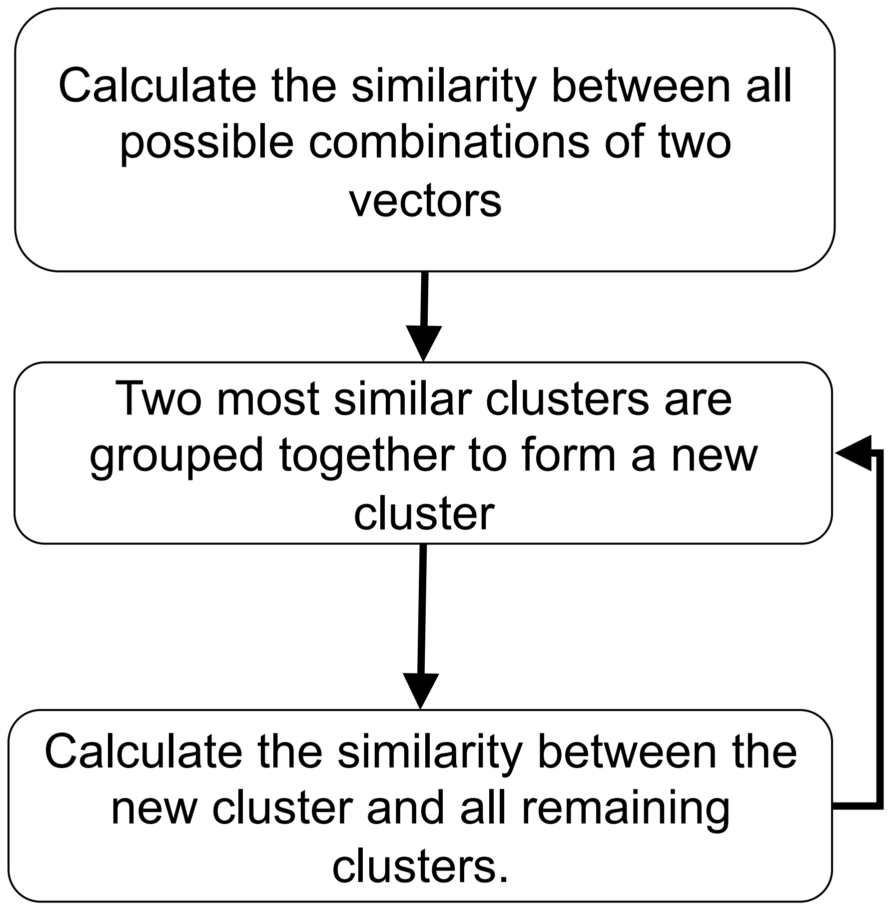
$$\boxed{\begin{aligned} m_x &= \frac{1}{N}\sum_{n=1}^{N} x_n \\ m_y &= \frac{1}{N}\sum_{n=1}^{N} y_n \end{aligned}}$$



$+1 \geq$ Pearson Correlation ≥ -1



Hierarchical Clustering



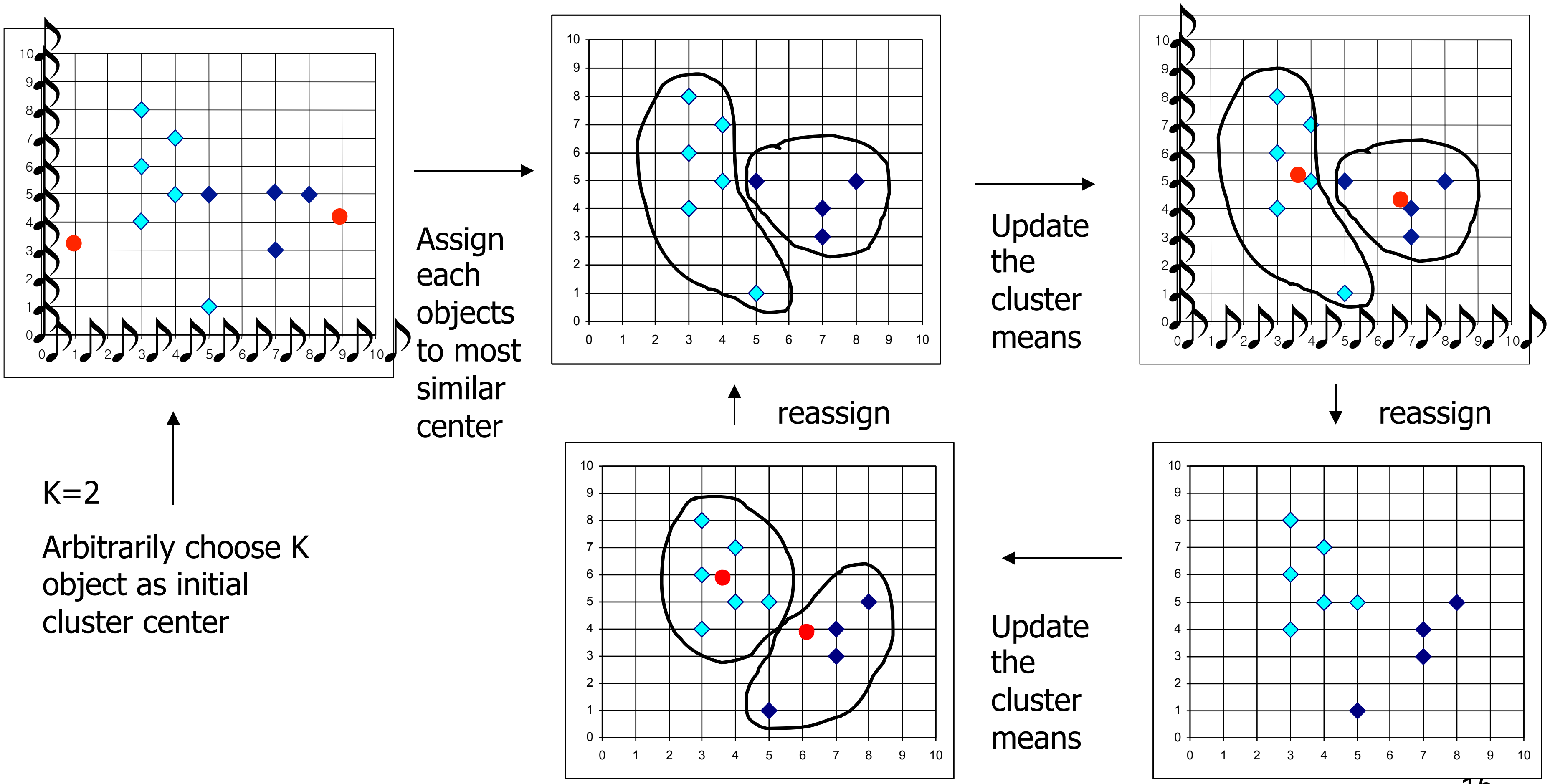
K-Means clustering

- The meaning of ‘K-means’
  - Why it is called ‘K-means’ clustering: K points are used to represent the clustering result; each point corresponds to the centre (geometric mean) of a cluster
- Each point is assigned to the cluster with the closest center point
- The number K must be specified
- Basic algorithm

K-Means clustering

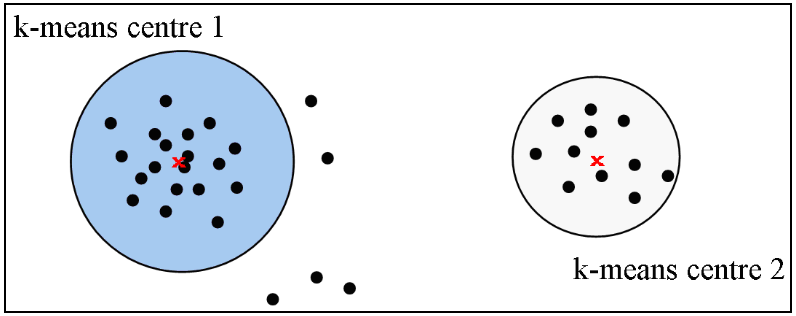
- Given *k*, the *k-means* algorithm is implemented in 4 steps:
  - Partition objects into *k* non-empty subsets
  - Arbitrarily choose *k* points as initial centers (centroids)
  - Assign each object to the cluster with the nearest center
  - Calculate the mean of the cluster and update the center point
  - Go back to Step 3, stop when no more new assignment

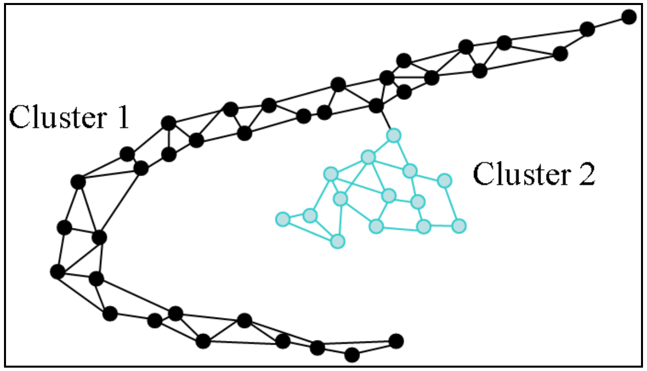
K-Means clustering



Clustering

- Data exploration method
- Can be interpreted as a purely geometrical approach of grouping similar data samples together
- Requires data representation and the definition of similarity
- K-means (and other algorithms)
- Involves parameters choice (number of clusters, etc)





Implementation of clustering methods

Scikit-learn

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large n_samples, medium n_clusters	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Spectral clustering	number of clusters	Medium n_samples, small n_clusters	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Hierarchical clustering	number of clusters	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints	Distances between points
DBSCAN	neighborhood size	Very large n_samples, medium n_clusters	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers

Minilab

- How to choose parameters: “toy” problem
- Clustering EV owners charging patterns
- Interpretation of clustering results