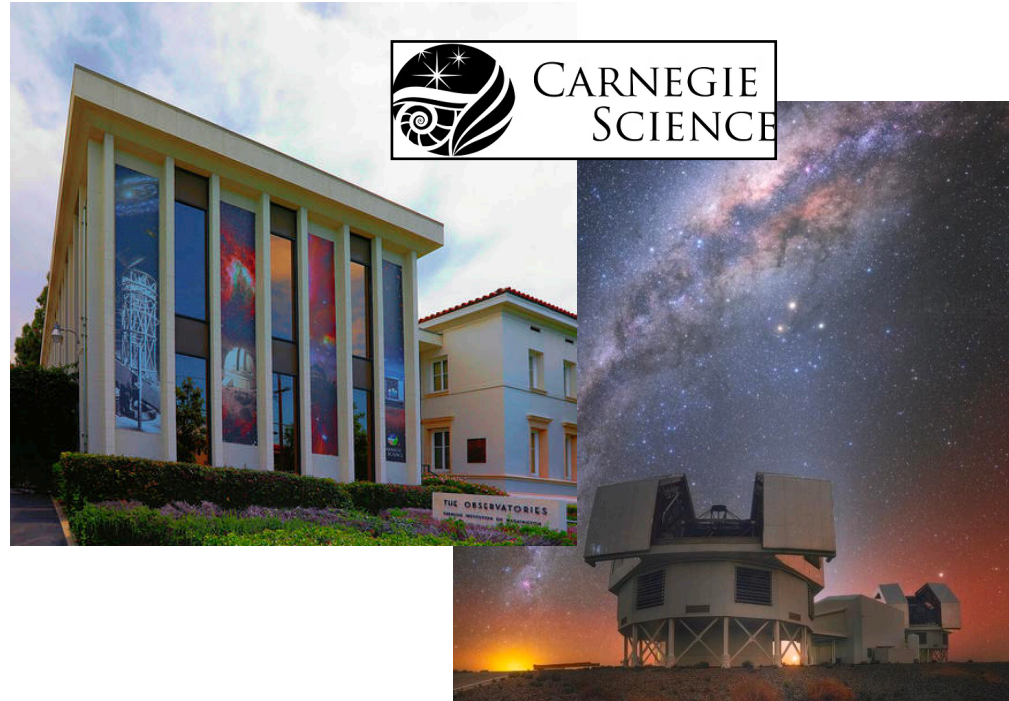

Applications of unsupervised machine learning techniques for data exploration and discovery in ISM science

Dalya Baron
Stanford University

International summer school on the ISM of galaxies (GISM3)

July 2025

Nice to meet you all!



This 5-yo is Ori



I really like running in the nature!

Data science drives astronomical research

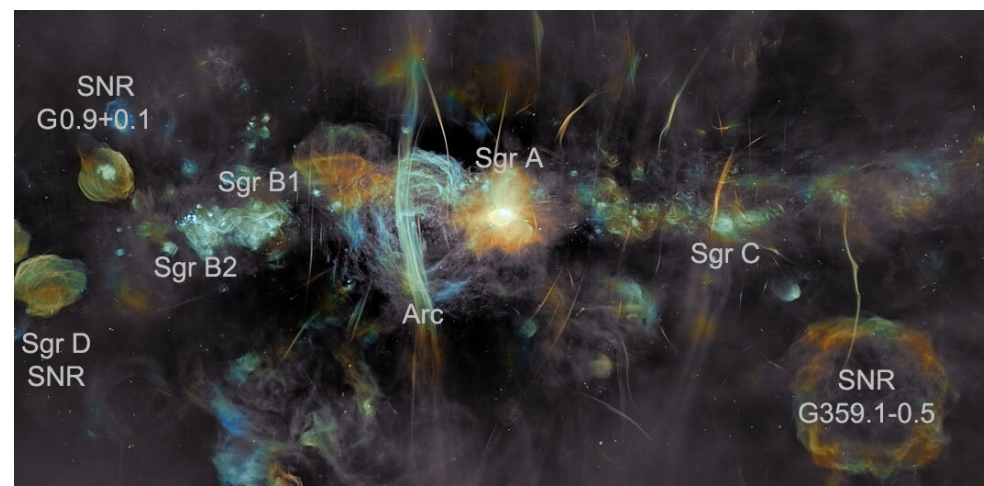
- ❖ Astrophysicists cannot build controlled experiments in the laboratory.
- ❖ We observe billions of astronomical sources, each is an ongoing experiment.

Data science drives astronomical research

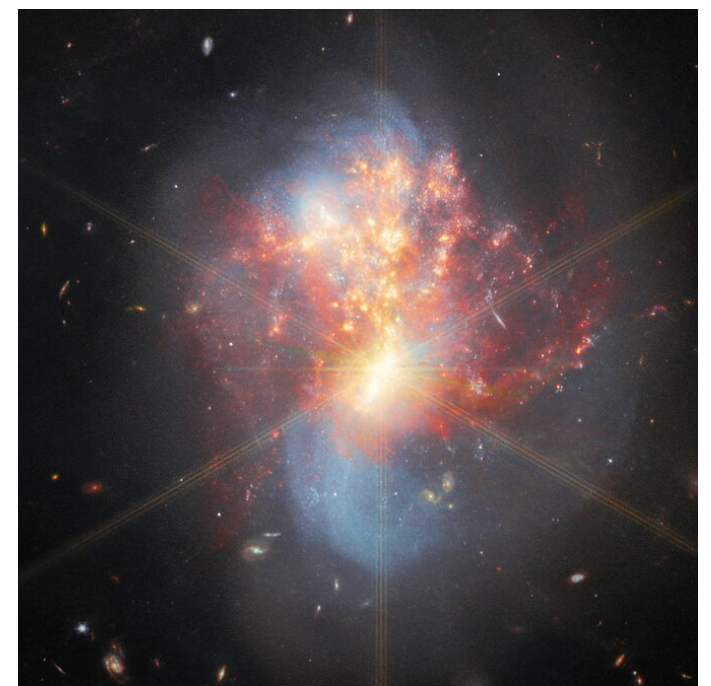
- ❖ Astrophysicists cannot build controlled experiments in the laboratory.
- ❖ We observe billions of astronomical sources, each is an ongoing experiment.
- ❖ The systems we observe are highly complex, and can be described by very different models with many unknown parameters.



The Pillars of Creation.
Credits: J. DePasquale, A.
M. Koekemoer, A. Pagan
(STScI).



The galactic center in radio.
Credits: I. Heywood & J. C. Munoz-
Mateos



A galaxy merger.
Credits: L. Armus & A.
Evans

Data science drives astronomical research

- ❖ To constrain the large space of possible physical models, we search for:
 - ❖ **Universal functions:** e.g., the initial mass function.
 - ❖ **Scaling relations and trends** between different properties: e.g., Tully-Fisher (L vs. v_{rot}), Kennicutt-Schmidt (M_{H_2} vs. SFR), and $M_{BH} - \sigma$ relations.
 - ❖ **Classes/clusters:** e.g., early vs. late type galaxies, stellar classes.
 - ❖ **Outliers:** e.g., quasars, super luminous supernovae, fast radio bursts.

Data science drives astronomical research

- ❖ To constrain the large space of possible physical models, we search for:
 - ❖ **Universal functions:** e.g., the initial mass function.
 - ❖ **Scaling relations and trends** between different properties: e.g., Tully-Fisher (L vs. v_{rot}), Kennicutt-Schmidt (M_{H_2} vs. SFR), and $M_{BH} - \sigma$ relations.
 - ❖ **Classes/clusters:** e.g., early vs. late type galaxies, stellar classes.
 - ❖ **Outliers:** e.g., quasars, super luminous supernovae, fast radio bursts.

In the emerging field of data science, a central goal is to uncover these patterns in complex data. In astronomy, this has been a driving force of progress for decades.

New data horizons

- ❖ We make discoveries when we observe the Universe in a new way.
- ❖ **The blessing of dimensionality:** multiple large surveys mapping astronomical sources in wavelength and time.

New data horizons

- ❖ We make discoveries when we observe the Universe in a new way.
- ❖ **The blessing of dimensionality**: multiple large surveys mapping astronomical sources in wavelength and time.
- ❖ Discovery opportunities now or in the near future (my biased POV!):
 - ❖ Milky Way: **Local Volume Mapper** + 3D dust maps + multi-band 2D maps tracing gas, dust, and molecules + HI cubes, and more!
 - ❖ ISM and nearby galaxies: **PHANGS** + DGIS + MANGA.
 - ❖ Stars: **Milky Way Mapper** + Gaia + TESS + 4MOST.
 - ❖ Galaxies and cosmology: **Euclid** + **DESI** + **Rubin** + Roman.

New data horizons

- ❖ We make discoveries when we observe the Universe in a new way.
- ❖ **The blessing of dimensionality**: multiple large surveys mapping astronomical sources in wavelength and time.
- ❖ Discovery opportunities now or in the near future (my biased POV!):
 - ❖ Milky Way: **Local Volume Mapper** + 3D dust maps + multi-band 2D maps tracing gas, dust, and molecules + HI cubes, and more!
 - ❖ ISM and nearby galaxies: **PHANGS** + DGIS + MANGA.
 - ❖ Stars: **Milky Way Mapper** + Gaia + TESS + 4MOST.
 - ❖ Galaxies and cosmology: **Euclid** + **DESI** + **Rubin** + Roman.

The combination of data from the different surveys will facilitate the next discoveries in astronomy.

This lecture: applications of unsupervised machine learning algorithms for data exploration and discovery in ISM science

- ❖ Unsupervised learning: a family of algorithms that do not require “ground truth” labels or target variables for training.

This lecture: applications of unsupervised machine learning algorithms for data exploration and discovery in ISM science

- ❖ Unsupervised learning: a family of algorithms that do not require “ground truth” labels or target variables for training.
- ❖ Operating directly on the data, they are used for:
 - ❖ Clustering: association of objects in the sample to a (typically small) number of groups.

This lecture: applications of unsupervised machine learning algorithms for data exploration and discovery in ISM science

- ❖ Unsupervised learning: a family of algorithms that do not require “ground truth” labels or target variables for training.
- ❖ Operating directly on the data, they are used for:
 - ❖ Clustering: association of objects in the sample to a (typically small) number of groups.
 - ❖ Dimensionality reduction: representing the high-dimensional dataset in a low dimensional space. Embedding into 2D or 3D enables visualization.


This lecture: applications of unsupervised machine learning algorithms for data exploration and discovery in ISM science

- ❖ Unsupervised learning: a family of algorithms that do not require “ground truth” labels or target variables for training.
- ❖ Operating directly on the data, they are used for:
 - ❖ Clustering: association of objects in the sample to a (typically small) number of groups.
 - ❖ Dimensionality reduction: representing the high-dimensional dataset in a low dimensional space. Embedding into 2D or 3D enables visualization.
 - ❖ Outlier detection: identification of rare or anomalous objects in the sample.

Anatomy of unsupervised algorithms

$$f(\overrightarrow{X}, \{a_1, a_2, \dots\}) = \overrightarrow{y}$$


Anatomy of unsupervised algorithms

$$f(\overrightarrow{X}, \{a_1, a_2, \dots\}) = \overrightarrow{y}$$


Input dataset:

- Raw data (spectra, images, light-curves).
- Extracted features.
- Measured relations between different objects (distances, correlations).

Anatomy of unsupervised algorithms

$$f(\overrightarrow{X}, \{a_1, a_2, \dots\}) = \overrightarrow{y}$$
A red arrow points from the vector \overrightarrow{X} to the 'Input dataset' box. A green arrow points from the set of parameters $\{a_1, a_2, \dots\}$ to the 'Hyperparameters' box.

Input dataset:

- Raw data (spectra, images, light-curves).
- Extracted features.
- Measured relations between different objects (distances, correlations).

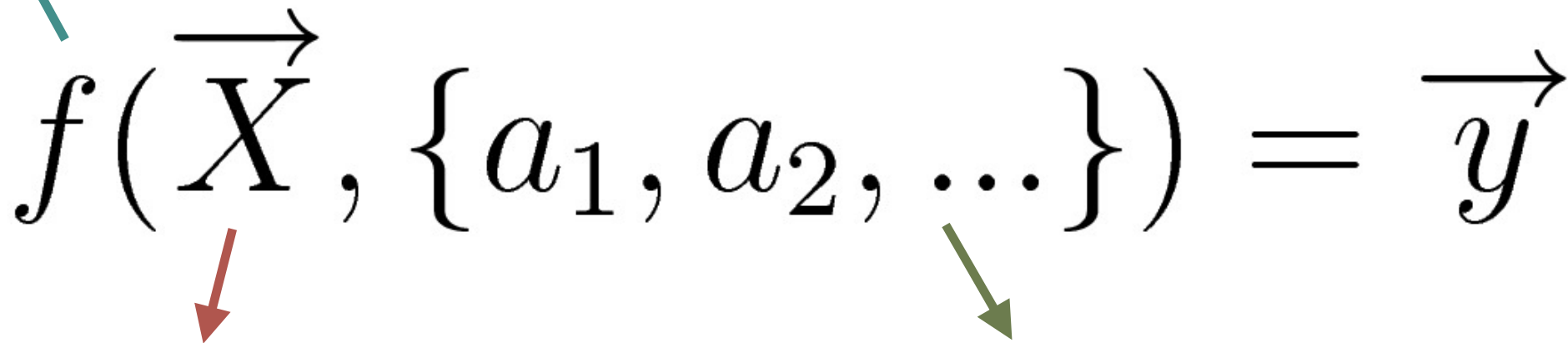
Hyperparameters:

- Tuning parameters of the algorithm.
- Can strongly affect the result.
- Traditionally, cannot be optimized for.

Anatomy of unsupervised algorithms

Internal choices / cost function:

- Usually, we cannot control these.
- Strongly affect the result, and define the range of possible outputs.



The diagram illustrates the components of an unsupervised algorithm. A central equation is shown: $f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$. A teal arrow points from the 'Internal choices / cost function' box to the set of parameters $\{a_1, a_2, \dots\}$. A red arrow points from the 'Input dataset' box to the input vector \vec{X} . A green arrow points from the 'Hyperparameters' box to the function f .

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

Input dataset:

- Raw data (spectra, images, light-curves).
- Extracted features.
- Measured relations between different objects (distances, correlations).

Hyperparameters:

- Tuning parameters of the algorithm.
- Can strongly affect the result.
- Traditionally, cannot be optimized for.

Anatomy of unsupervised algorithms

Internal choices / cost function:

- Usually, we cannot control these.
- Strongly affect the result, and define the range of possible outputs.

Algorithm output:

- Density distribution.
- Clusters.
- Embedding in low-D space.
- Outliers.

The diagram illustrates the anatomy of unsupervised algorithms. At the center is the equation $f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$. Four arrows point from this equation to surrounding boxes: a teal arrow points from \vec{X} to the 'Input dataset' box; a red arrow points from $\{a_1, a_2, \dots\}$ to the 'Internal choices / cost function' box; a green arrow points from f to the 'Hyperparameters' box; and a brown arrow points from \vec{y} to the 'Algorithm output' box.

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

Input dataset:

- Raw data (spectra, images, light-curves).
- Extracted features.
- Measured relations between different objects (distances, correlations).

Hyperparameters:

- Tuning parameters of the algorithm.
- Can strongly affect the result.
- Traditionally, cannot be optimized for.

Good Practices

- ❖ Start simple:

- ❖ Simulate simple low-dimensional dataset, without noise, where the output can be anticipated.
- ❖ Compare the output of the algorithm for different data representations and different choices of hyper-parameters.

- ❖ Gradually complicate the model:

- ❖ Add more dimensions (some of them should be uninformative).
- ❖ Add noise.
- ❖ Compare the output for different representations and hyper-parameters.

- ❖ Physically-motivated model:

- ❖ Simulate a physically-motivated dataset.
- ❖ Experiment with different noise properties, different representations, and hyper-parameters.

- ❖ Try to break the algorithm!

List of topics

- ❖ Input data sets and distance measures.
- ❖ Clustering algorithms.
- ❖ Dimensionality reduction algorithms.
- ❖ Outlier detection algorithms.

Input data and distance measures

Types of input data

The algorithm takes as an input a list of objects with N measured properties. By default, each object is considered as a point in an N -dimensional **Euclidean** space.

Types of input data

The algorithm takes as an input a list of objects with N measured properties. By default, each object is considered as a point in an N -dimensional **Euclidean** space.

- ❖ **Raw data** - data obtained directly from the telescope after “minimal” processing:
 - ❖ Astronomical images in different bands.
 - ❖ Spectra.
 - ❖ Time-series data (can also be in multiple bands).

Types of input data

The algorithm takes as an input a list of objects with N measured properties. By default, each object is considered as a point in an N -dimensional **Euclidean** space.

- ❖ **Raw data - data obtained directly from the telescope after “minimal” processing:**
 - ❖ Astronomical images in different bands.
 - ❖ Spectra.
 - ❖ Time-series data (can also be in multiple bands).
- ❖ **Features extracted from the raw data.**
 - ❖ For stellar spectra: effective temperature, bolometric luminosity, metallicity, mass, etc.
 - ❖ For galaxy images: effective radius, Sersic index, morphological class (spiral or elliptical), etc.

Types of input data

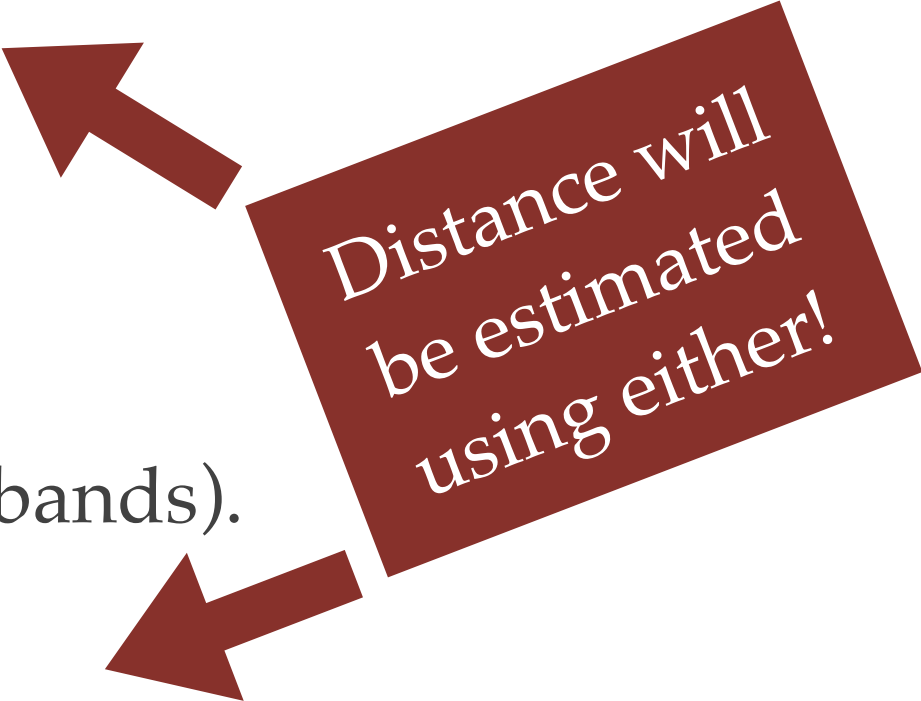
The algorithm takes as an input a list of objects with N measured properties. By default, each object is considered as a point in an N -dimensional **Euclidean** space.

- ❖ **Raw data - data obtained directly from the telescope after “minimal” processing:**
 - ❖ Astronomical images in different bands.
 - ❖ Spectra.
 - ❖ Time-series data (can also be in multiple bands).
- ❖ **Features extracted from the raw data.**
 - ❖ For stellar spectra: effective temperature, bolometric luminosity, metallicity, mass, etc.
 - ❖ For galaxy images: effective radius, Sersic index, morphological class (spiral or elliptical), etc.
- ❖ **Relations between the objects: correlation or distance matrix.**

Types of input data

The algorithm takes as an input a list of objects with N measured properties. By default, each object is considered as a point in an N -dimensional **Euclidean** space.

- ❖ **Raw data - data obtained directly from the telescope after “minimal” processing:**
 - ❖ Astronomical images in different bands.
 - ❖ Spectra.
 - ❖ Time-series data (can also be in multiple bands).
- ❖ **Features extracted from the raw data.**
 - ❖ For stellar spectra: effective temperature, bolometric luminosity, metallicity, mass, etc.
 - ❖ For galaxy images: effective radius, Sersic index, morphological class (spiral or elliptical), etc.
- ❖ **Relations between the objects: correlation or distance matrix.**



Distance will be estimated using either!

Types of input data

The algorithm takes as an input a list of objects with N measured properties. By default, each object is considered as a point in an N -dimensional **Euclidean** space.

- ❖ **Raw data - data obtained directly from the telescope after “minimal” processing:**
 - ❖ Astronomical images in different bands.
 - ❖ Spectra.
 - ❖ Time-series data (can also be in multiple bands).
- ❖ **Features extracted from the raw data.**
 - ❖ For stellar spectra: effective temperature, bolometric luminosity, metallicity, mass, etc.
 - ❖ For galaxy images: effective radius, Sersic index, morphological class (spiral or elliptical), etc.
- ❖ **Relations between the objects: correlation or distance matrix.**

Less
processing

More
processing

Types of input data

The algorithm takes as an input a list of objects with N measured properties. By default, each object is considered as a point in an N -dimensional **Euclidean** space.

- ❖ **Raw data - data obtained directly from the telescope after “minimal” processing:**
 - ❖ Astronomical images in different bands.
 - ❖ Spectra.
 - ❖ Time-series data (can also be in multiple bands).
- ❖ **Features extracted from the raw data.**
 - ❖ For stellar spectra: effective temperature, bolometric luminosity, metallicity, mass, etc.
 - ❖ For galaxy images: effective radius, Sersic index, morphological class (spiral or elliptical), etc.
- ❖ **Relations between the objects: correlation or distance matrix.**

Less
knowledge

More
knowledge

Raw data vs. derived features

- ❖ **Data quality:** Deriving features often involves cleaning the data, improving its quality.

Raw data vs. derived features

- ❖ **Data quality:** Deriving features often involves cleaning the data, improving its quality.
- ❖ **Simplicity:** Derived features reduce dimensionality, making the task easier and more scalable. Using prior knowledge simplifies learning, rather than relying on algorithms to rediscover it.

Raw data vs. derived features

- ❖ **Data quality:** Deriving features often involves cleaning the data, improving its quality.
- ❖ **Simplicity:** Derived features reduce dimensionality, making the task easier and more scalable. Using prior knowledge simplifies learning, rather than relying on algorithms to rediscover it.
- ❖ **Interpretability:** Outputs are easier to understand when based on known, derived features. Even if we apply ML to the raw data, interpreting the result will require derived features!

Raw data vs. derived features

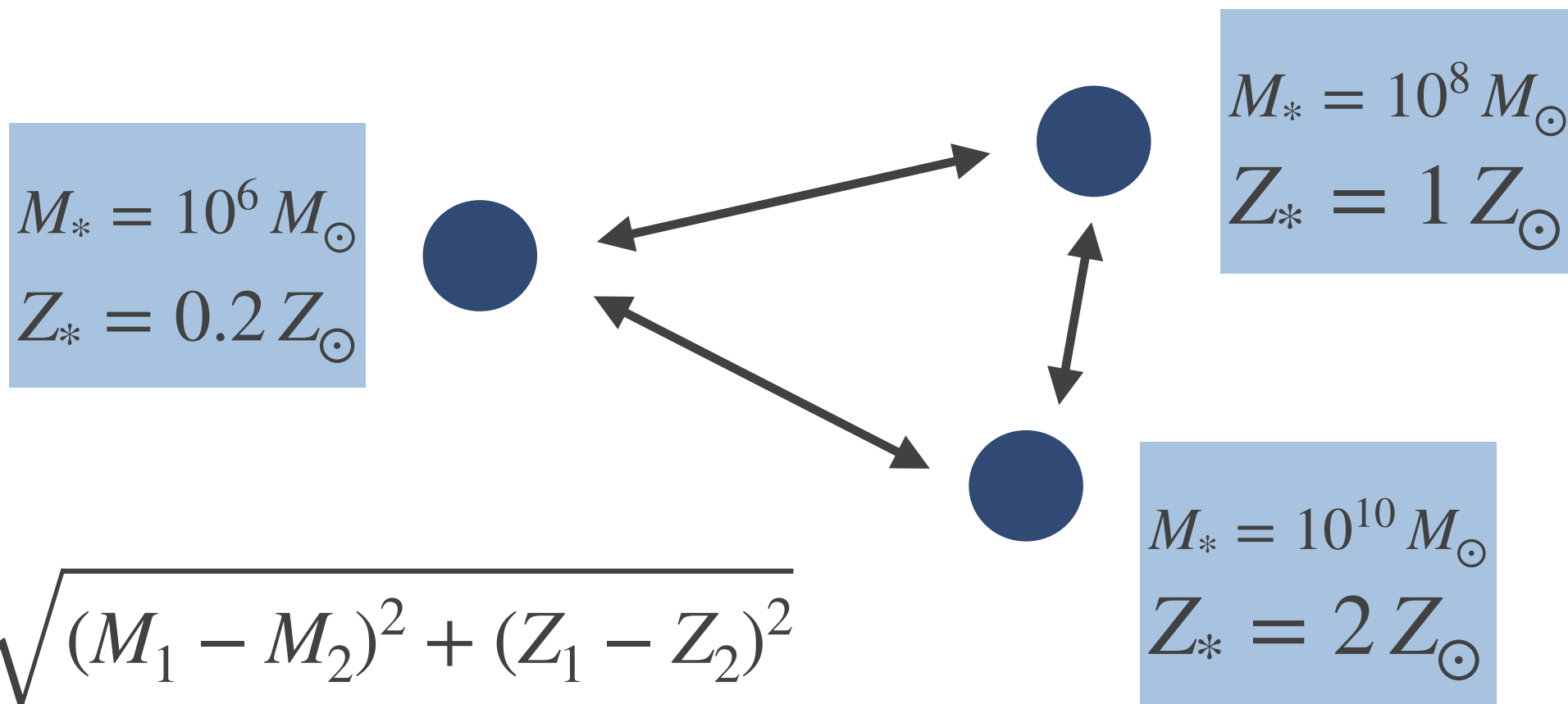
- ❖ **Data quality:** Deriving features often involves cleaning the data, improving its quality.
- ❖ **Simplicity:** Derived features reduce dimensionality, making the task easier and more scalable. Using prior knowledge simplifies learning, rather than relying on algorithms to rediscover it.
- ❖ **Interpretability:** Outputs are easier to understand when based on known, derived features. Even if we apply ML to the raw data, interpreting the result will require derived features!
- ❖ **Upper limits and non-detections:** cannot be trivially incorporated when using derived features.

Raw data vs. derived features

- ❖ **Data quality:** Deriving features often involves cleaning the data, improving its quality.
- ❖ **Simplicity:** Derived features reduce dimensionality, making the task easier and more scalable. Using prior knowledge simplifies learning, rather than relying on algorithms to rediscover it.
- ❖ **Interpretability:** Outputs are easier to understand when based on known, derived features. Even if we apply ML to the raw data, interpreting the result will require derived features!
- ❖ **Upper limits and non-detections:** cannot be trivially incorporated when using derived features.
- ❖ **Potential for new discoveries:** Focusing only on known features limits the chance of finding unknown patterns.

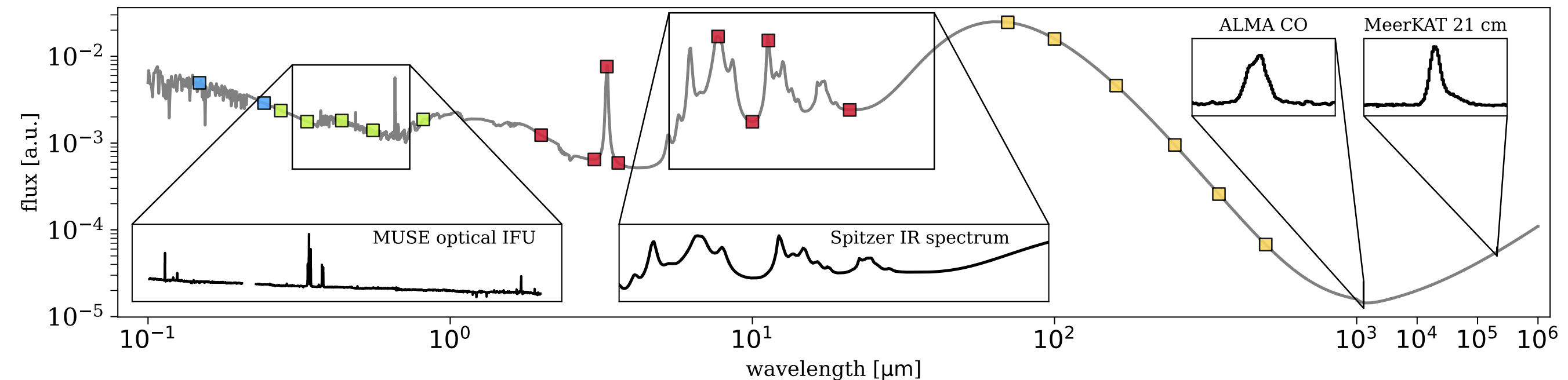
Data considerations: scaling & normalization

- ❖ Astronomical observations have physical units, and might have different dynamical scales. Features with larger variance will dominate the summed Euclidean distance between individual objects.



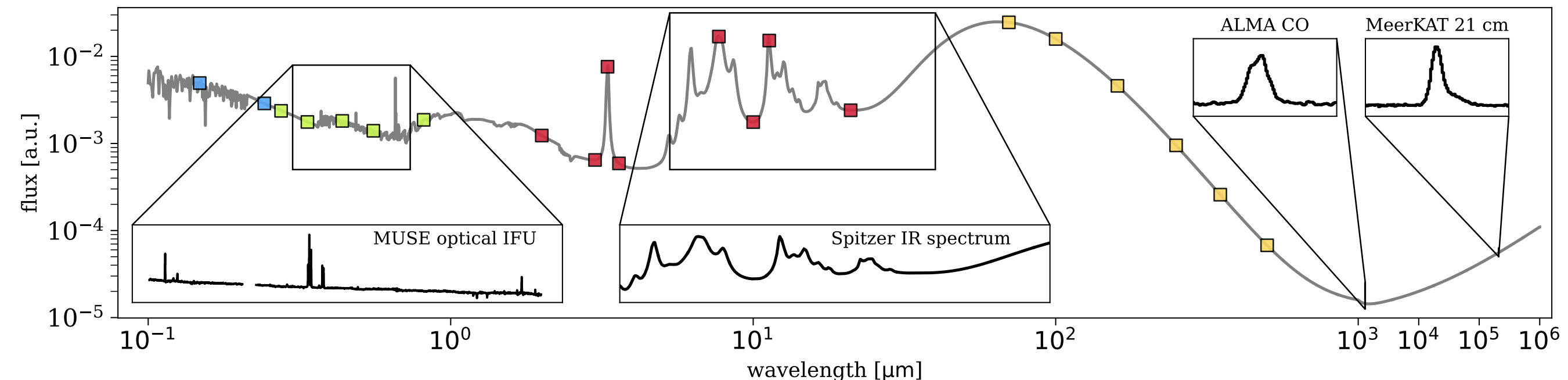
Data considerations: scaling & normalization

- ❖ Astronomical observations have physical units, and might have different dynamical scales. Features with larger variance will dominate the summed Euclidean distance between individual objects.



Data considerations: scaling & normalization

- ❖ Astronomical observations have physical units, and might have different dynamical scales. Features with larger variance will dominate the summed Euclidean distance between individual objects.



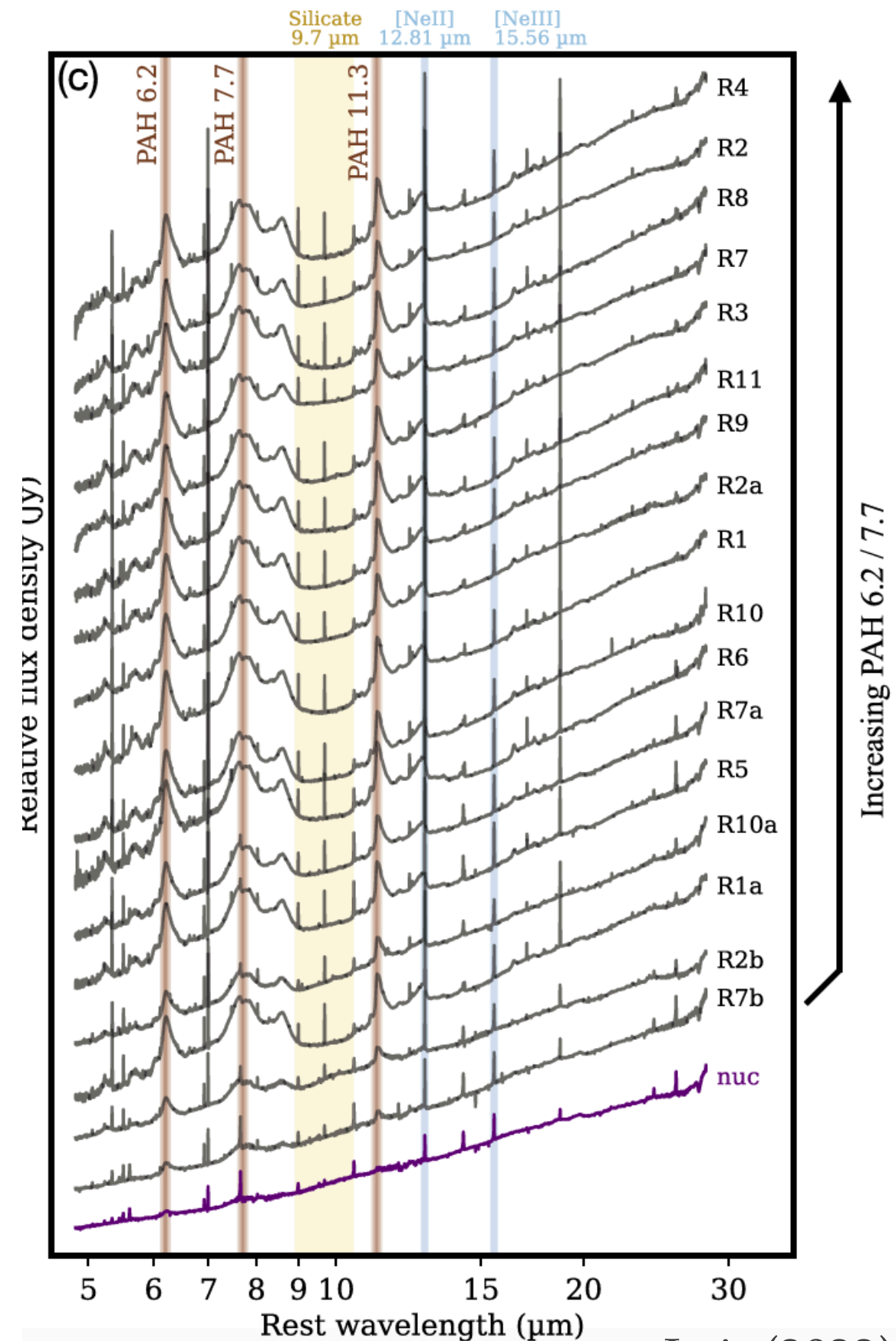
- ❖ **What to do?** Apply rescaling and normalization to all the features. Use the logarithm of the feature as the new feature, and/or normalize using the mean and standard deviation of the distribution: $f_{norm} = (f - \mu_f) / \sigma_f$.

Data considerations: outliers

- ❖ Some algorithms are based on the variance within each feature and are highly-sensitive to the presence of outliers in the data.
- ❖ **What to do?** Use histograms to inspect each feature separately and identify and/or remove outliers.

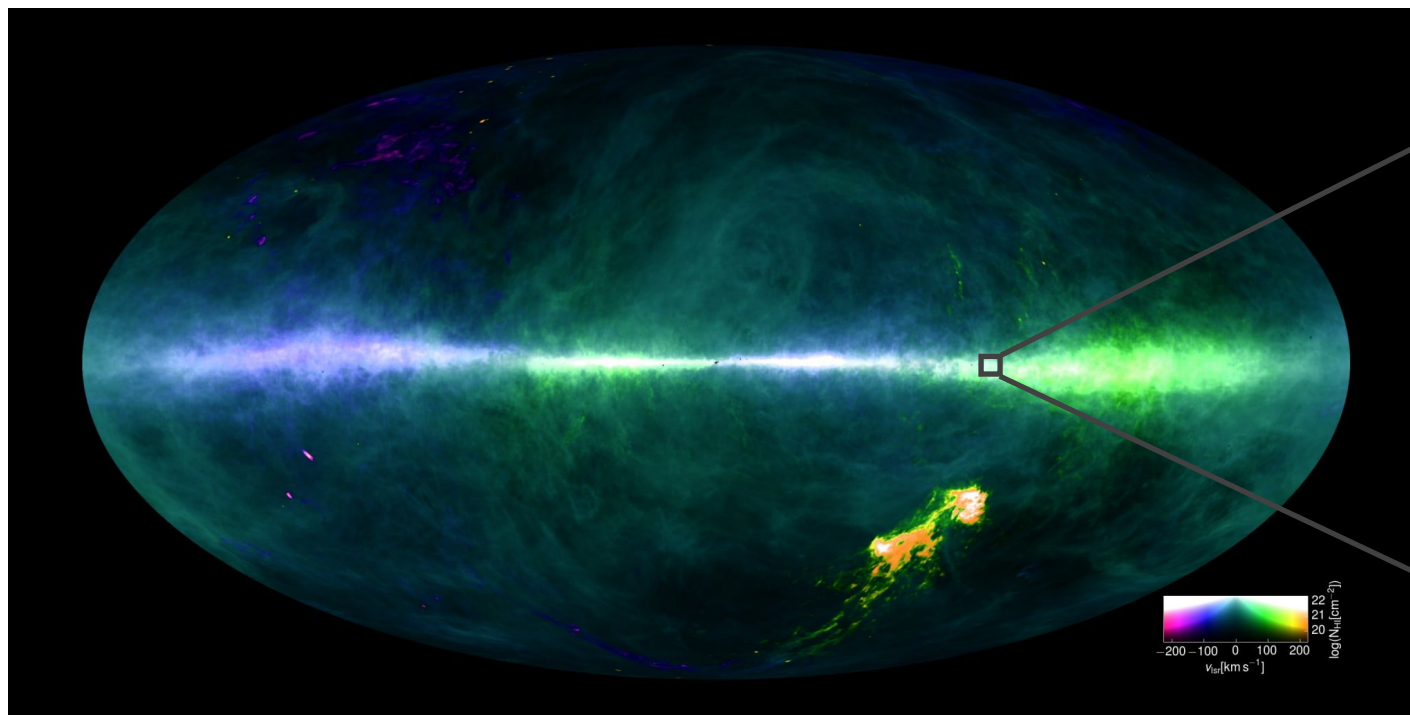
Data considerations: correlated features

- ❖ A set of highly-correlated features in the dataset will result in a higher weight in the summed Euclidean distance, which may wash-out other structures in the dataset.
- ❖ **What to do?**
 - ❖ Use PCA-derived features.
 - ❖ Compute feature ratios or deviations from scaling relations.
 - ❖ Use NN-based dimensionality reduction (e.g., auto-encoder).

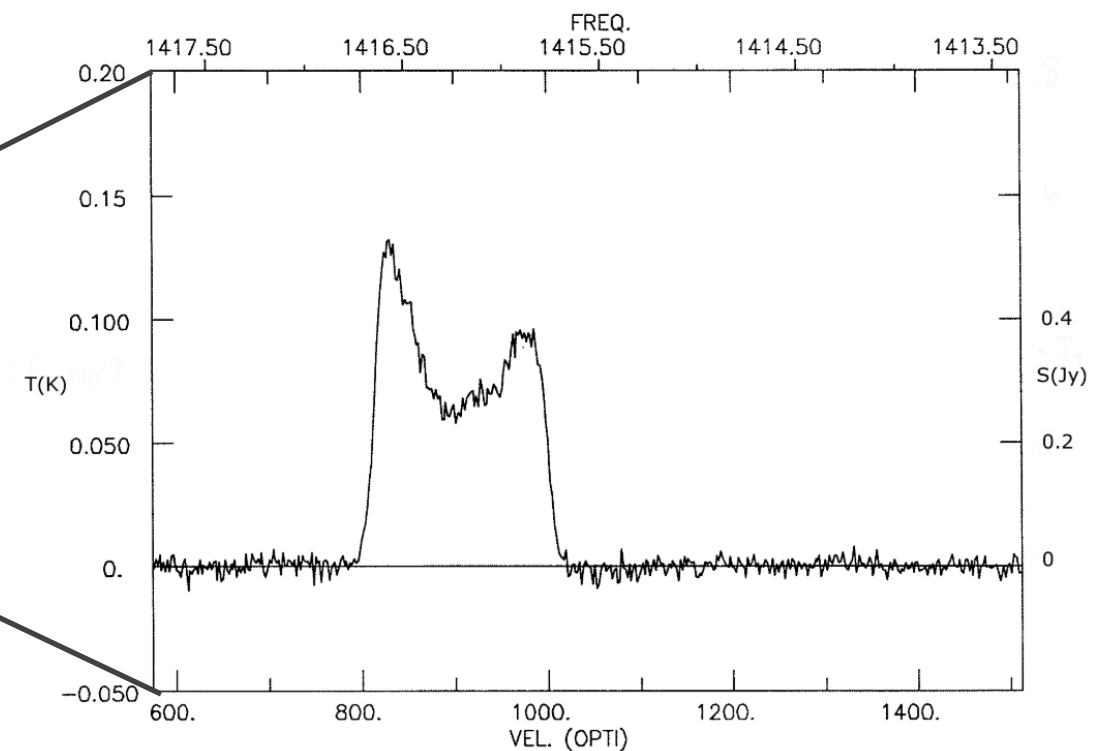


Data considerations: feature importance

- ❖ Not all features are equally important:
 - ❖ Some features of the data will not contain important information.



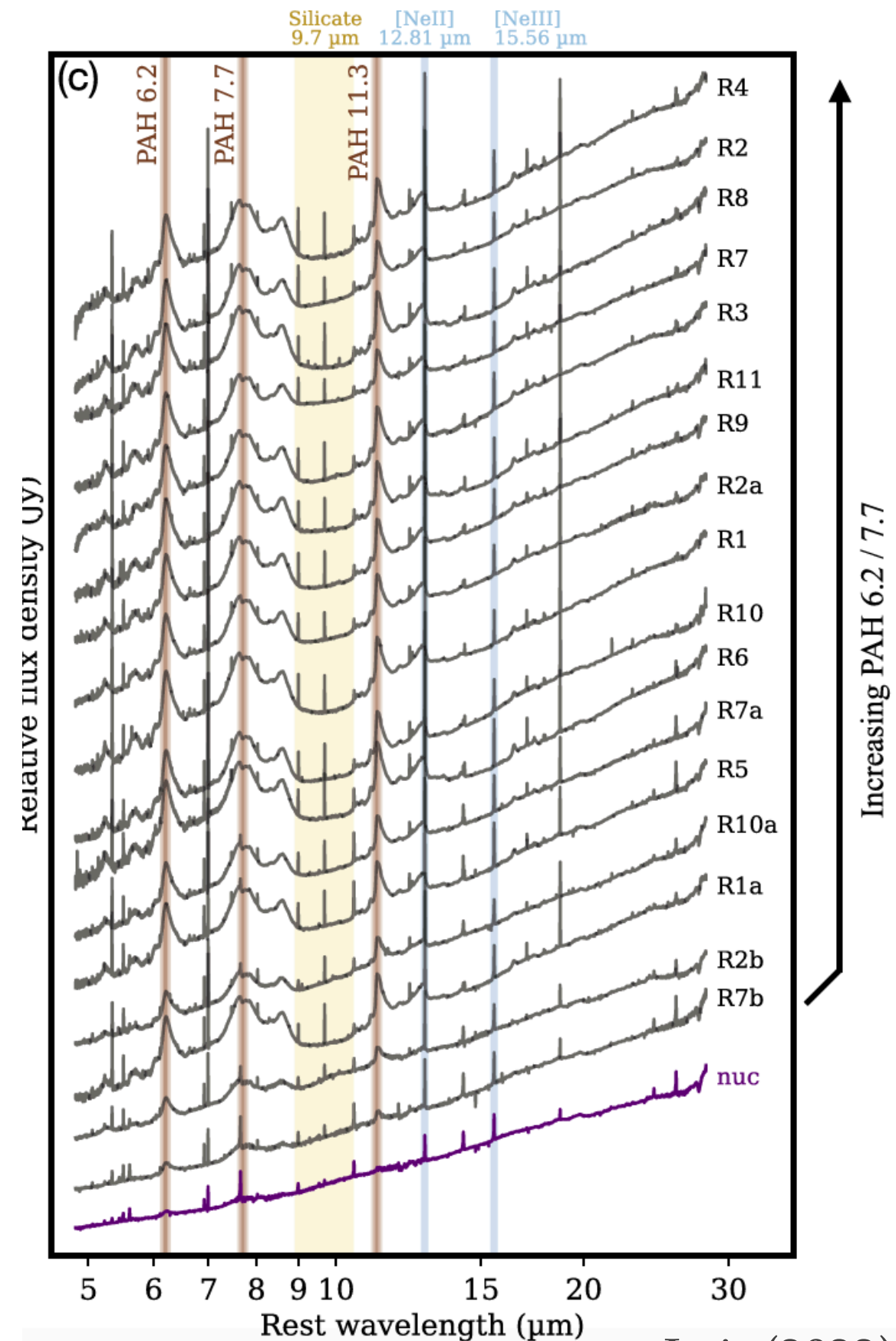
HI4PI map



HI spectrum
(taken from Haynes+1998)

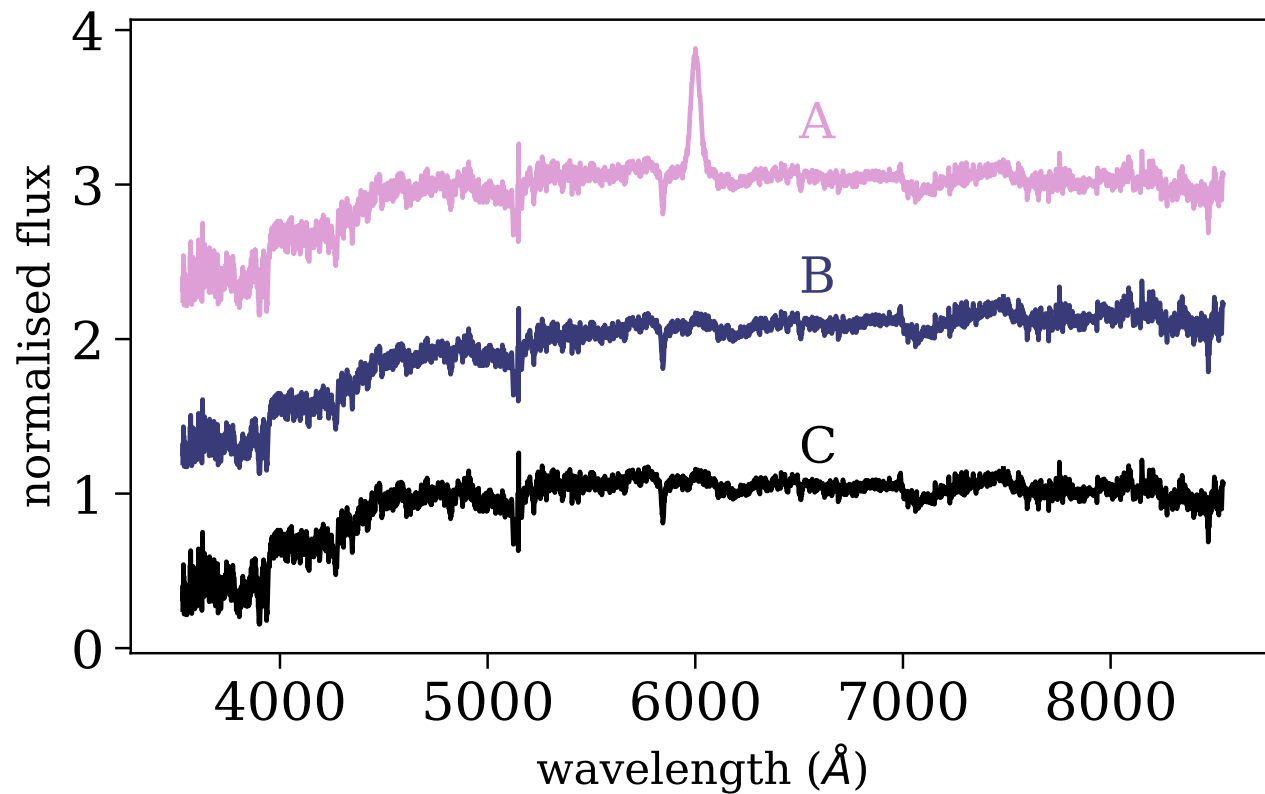
Data considerations: feature importance

- ❖ Not all features are equally important:
- ❖ Important features may carry a small weight in terms of distance or variance.

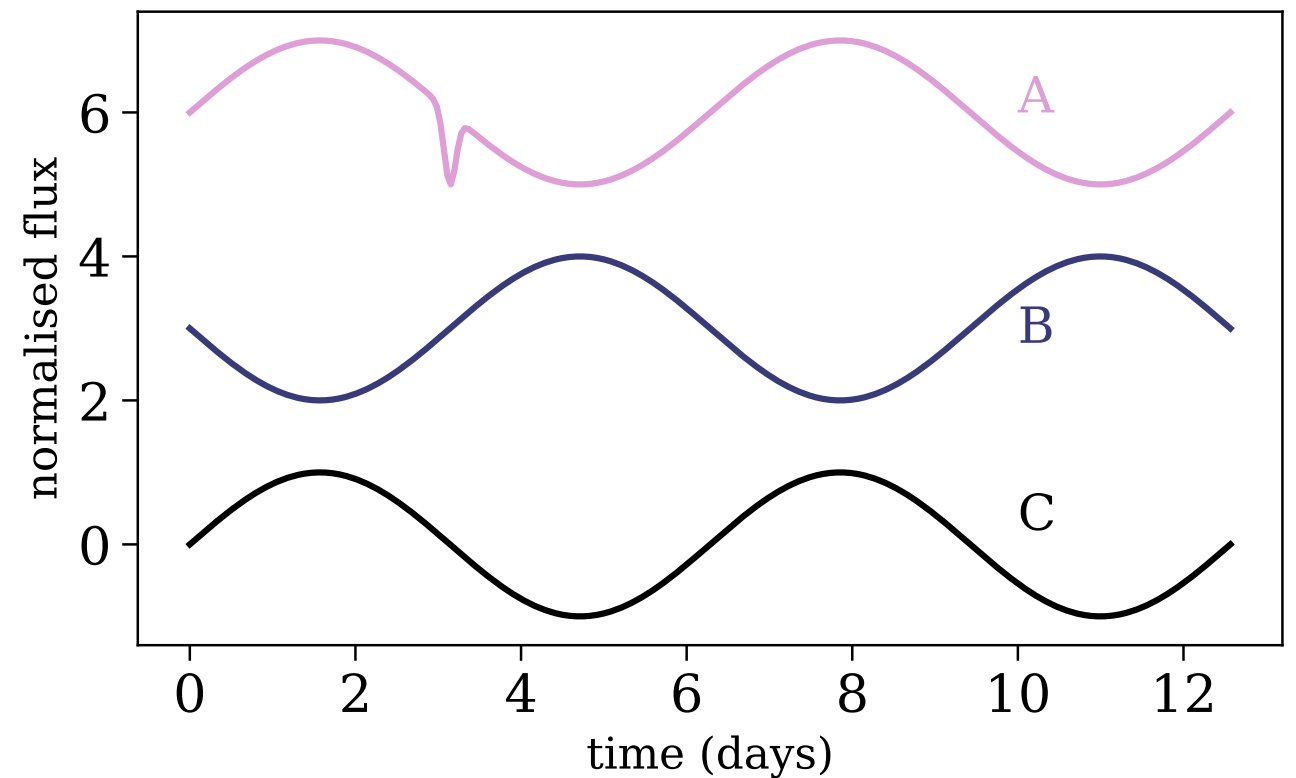


Raw data: aspects to consider

Example of spectra:



Example of time-series:



- ❖ May be beneficial to transform the data into a different space.
For example:
 - ❖ Frequency domain for time series data.
 - ❖ Wavelet transform for imaging data.
 - ❖ Representation using “eigenvectors” of spectral information.

Distance measures

Regardless of whether we want to perform clustering, dimensionality reduction, or outlier detection, the large majority of algorithms start by estimating the pairwise distance between objects in the N-dimensional space.

Distance measures

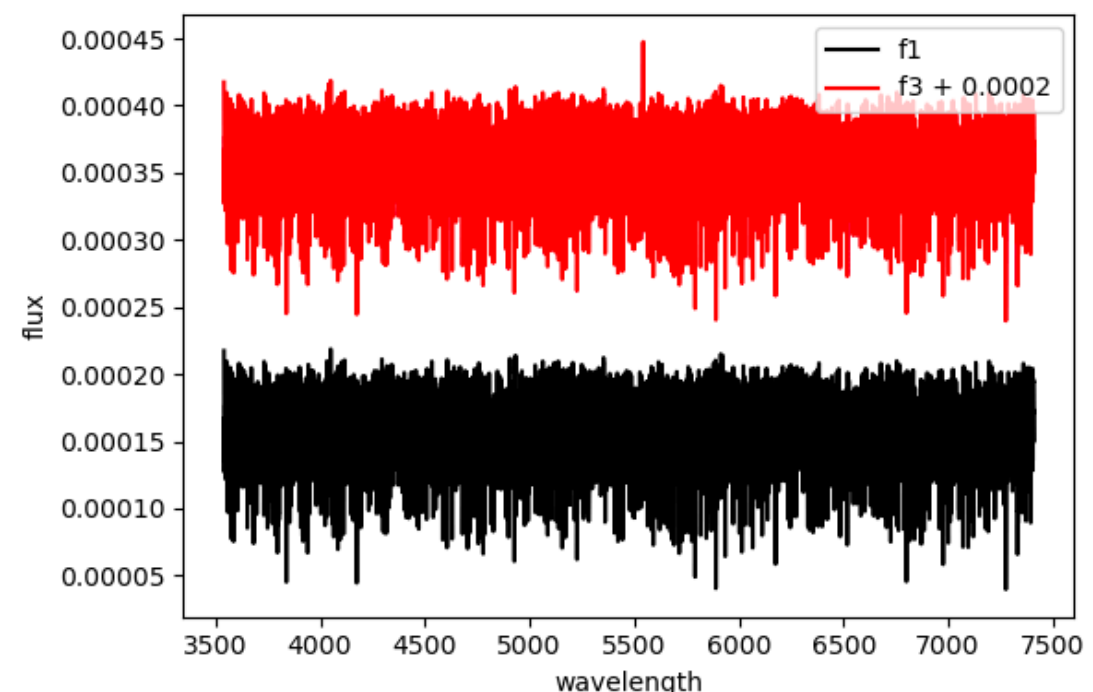
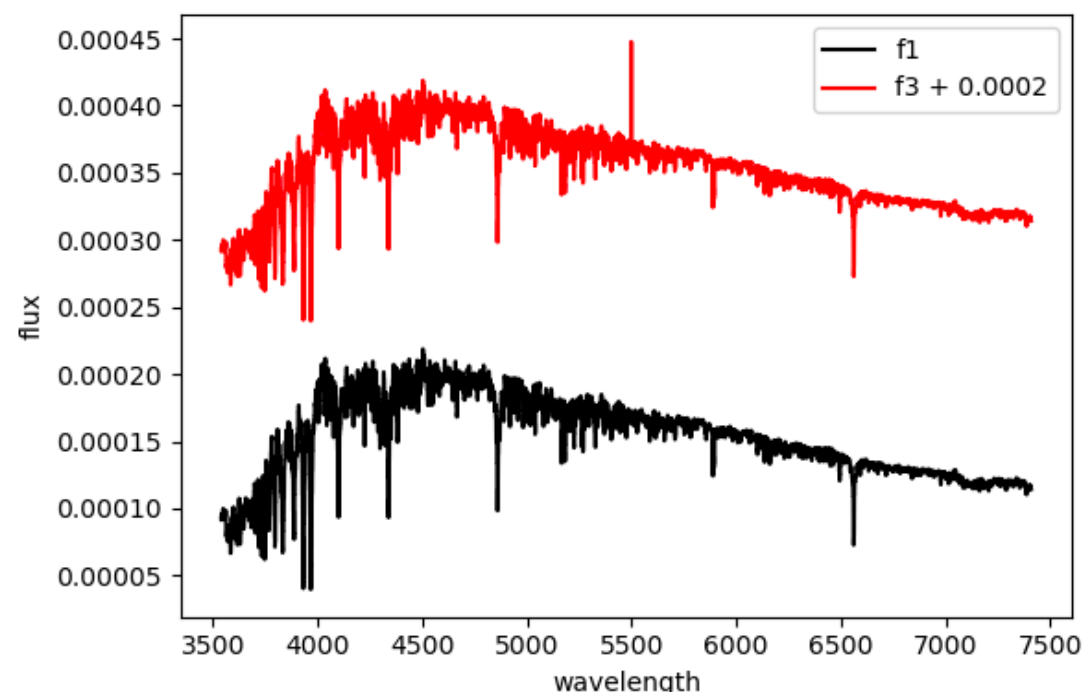
Regardless of whether we want to perform clustering, dimensionality reduction, or outlier detection, the large majority of algorithms start by estimating the pairwise distance between objects in the N-dimensional space.

❖ Euclidean Distance:

❖ The default distance metric assumed in most cases.

❖ All features are equally important: $D_{ij}^2 = \sum_{f:\text{features}} (x_{if} - x_{jf})^2$

❖ The relative order between the different features does not matter!



Distance measures

Regardless of whether we want to perform clustering, dimensionality reduction, or outlier detection, the large majority of algorithms start by estimating the pairwise distance between objects in the N-dimensional space.

- ❖ Euclidean Distance:

- ❖ The default distance metric assumed in most cases.

- ❖ All features are equally important: $D_{ij}^2 = \sum_{f: \text{features}} (x_{if} - x_{jf})^2$

- ❖ The relative order between the different features does not matter!

- ❖ Other metrics:

- ❖ Pearson / Spearman correlation coefficient.

- ❖ KL-divergence.

- ❖ Earth mover's distance or energy distance: the relative order of the features matters!!.

- ❖ A list of popular metrics can be found here.

Dimensionality reduction algorithms

PCA: Principal Component Analysis

ICA: Independent Component Analysis

NNMF: Non-negative Matrix Factorization

SOM: Self Organizing Maps

tSNE: t-distributed Stochastic Neighbor Embedding

UMAP: Uniform Manifold Approximation and Projection

What is dimensionality reduction?

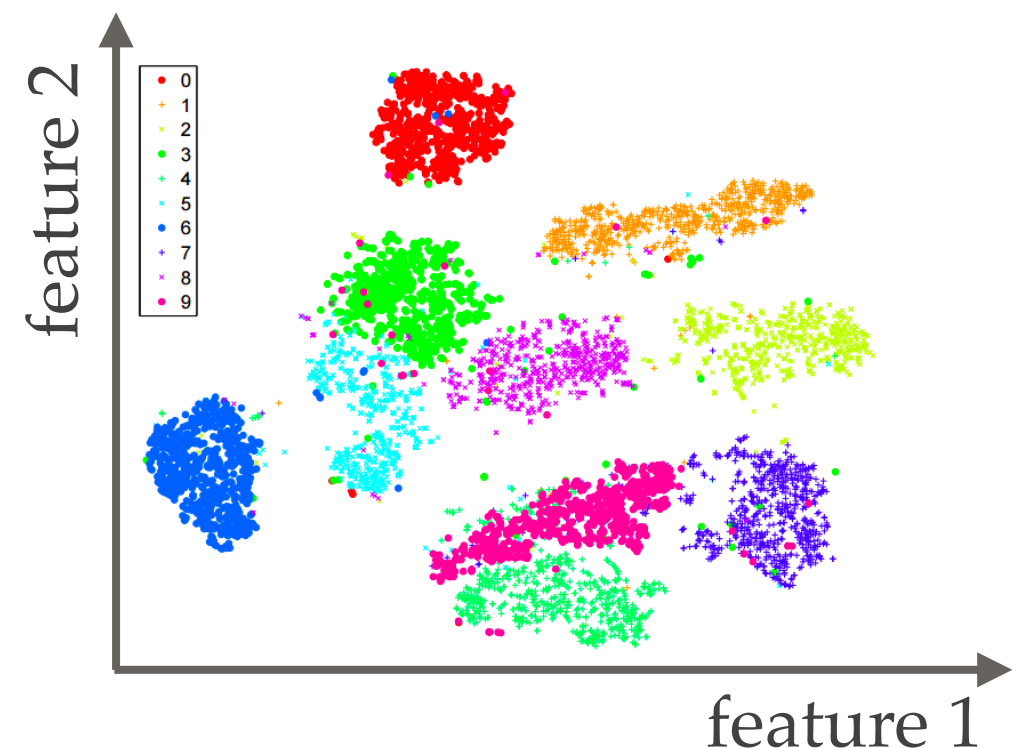
- ❖ Dimensionality reduction is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation *retains some meaningful* properties of the original data.

28 x 28 features per object



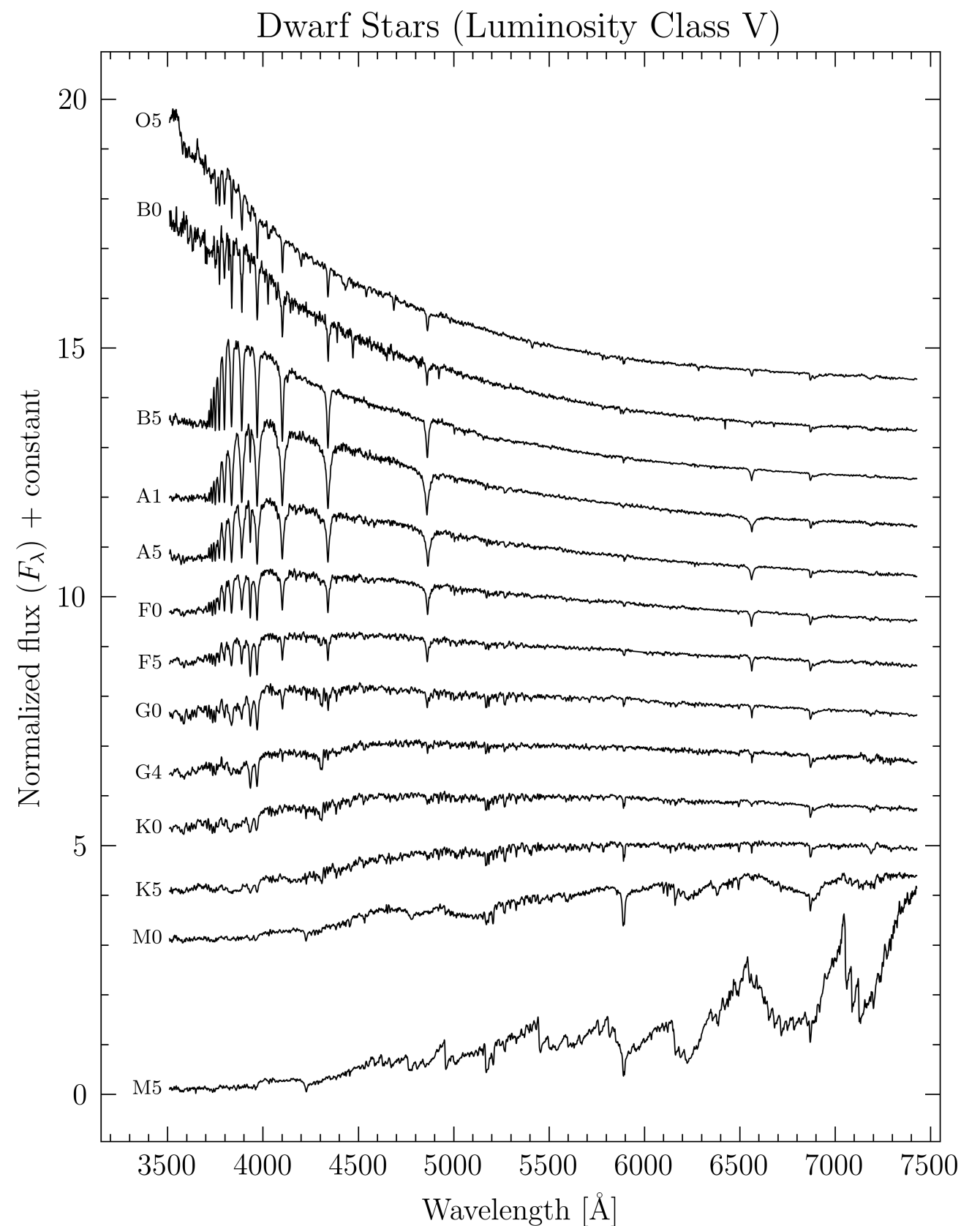
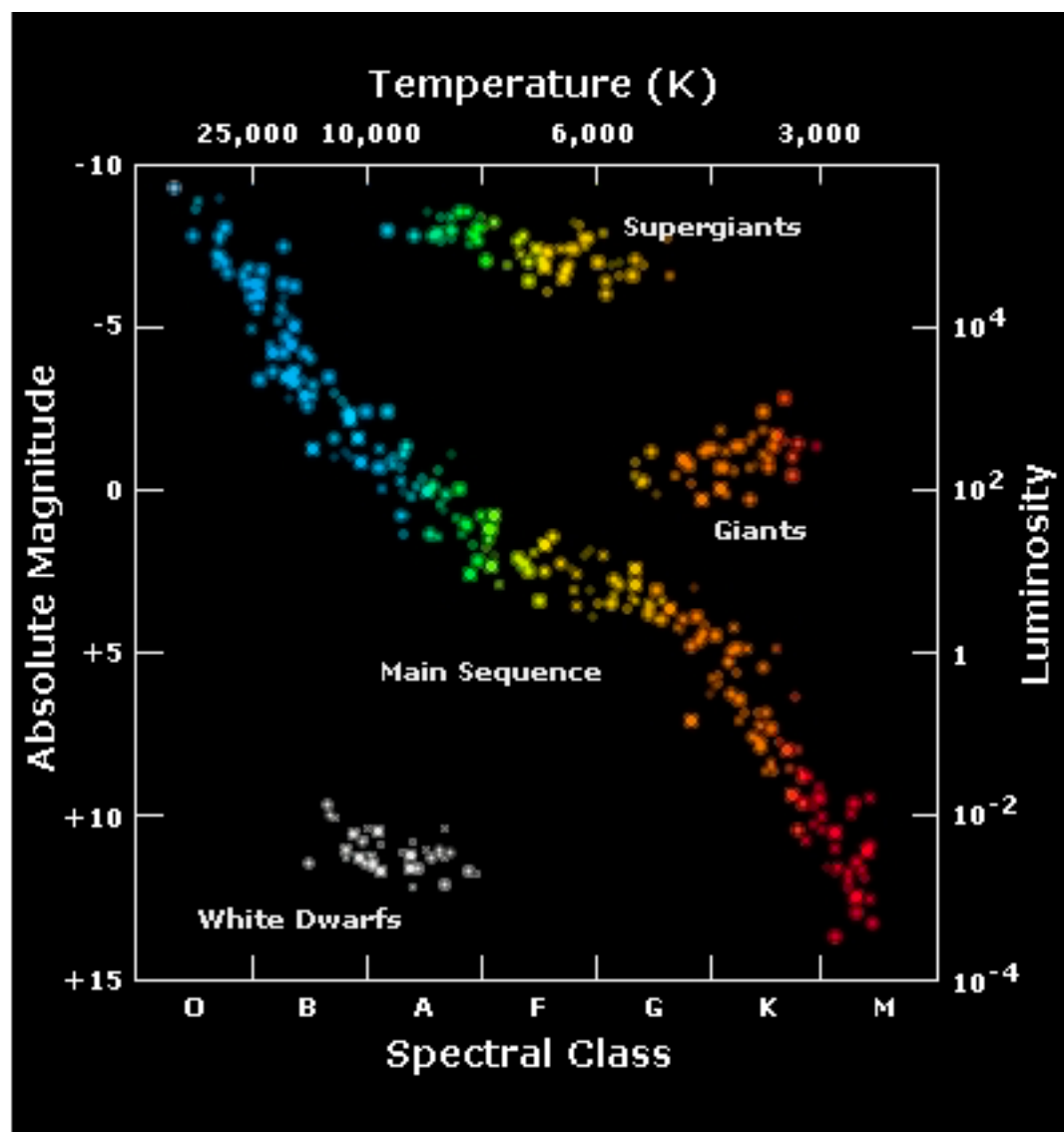
dimensionality
reduction
algorithm

2 features per object



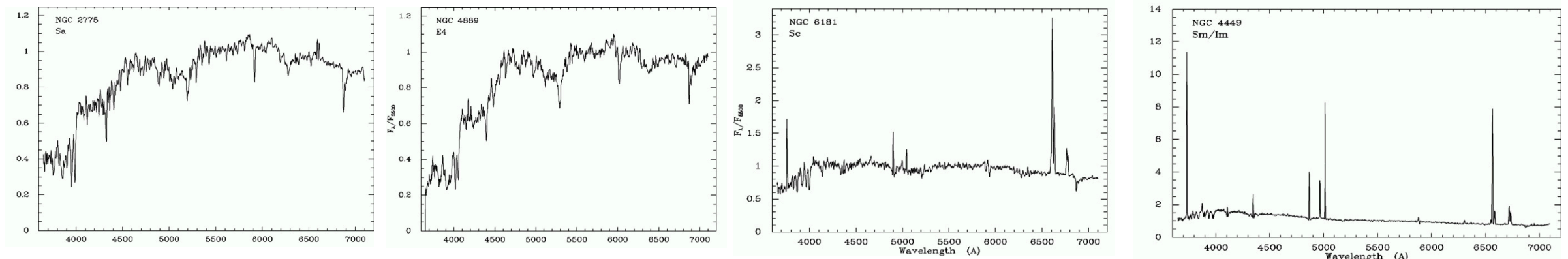
“Traditional” dimensionality reduction in astronomy

❖ The stellar sequence:

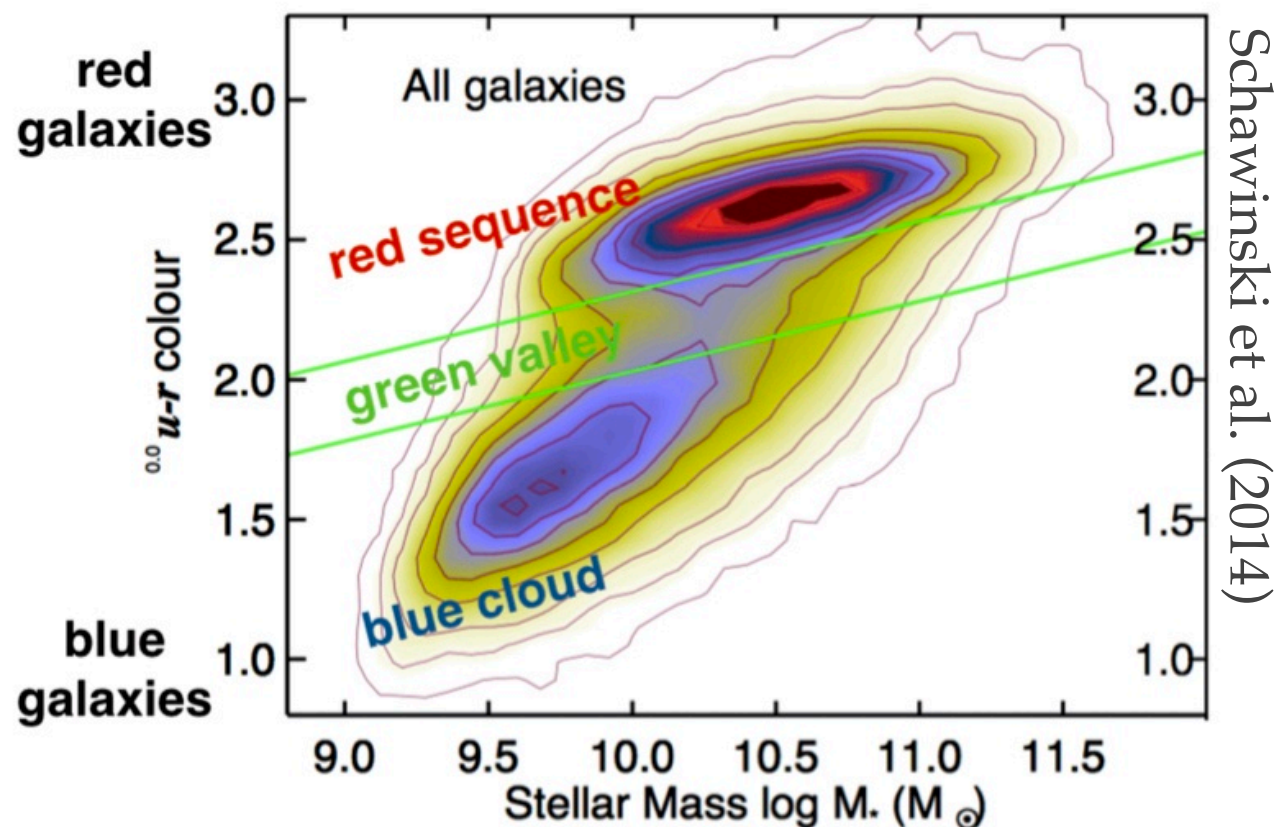


“Traditional” dimensionality reduction in astronomy

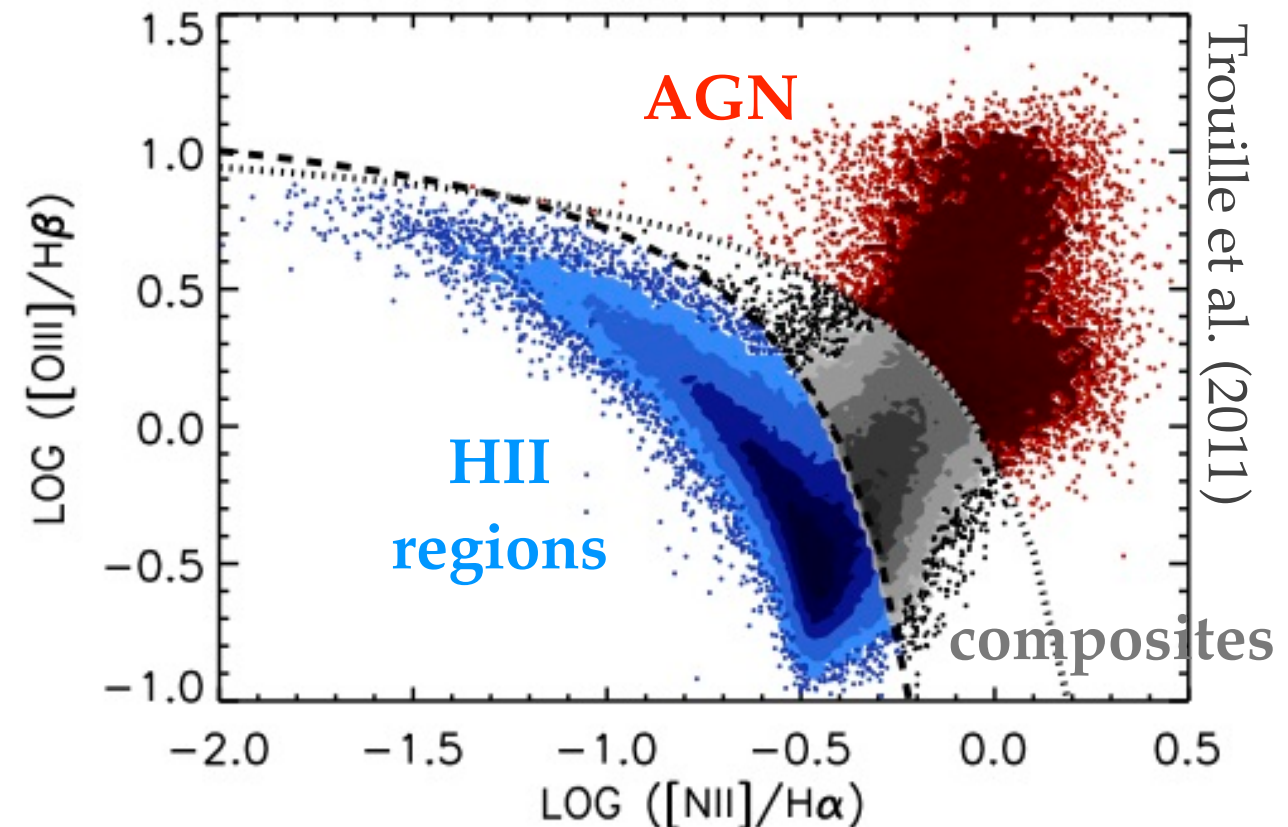
- ❖ Galaxy spectra: color-magnitude diagram and the BPT diagram



Color-magnitude diagram



BPT diagram



Different use cases

- ❖ Uncover new trends / correlations.
- ❖ Data visualization and interpretation.
- ❖ Look for outliers or interesting objects.

Different use cases

- ❖ Uncover new trends / correlations.
- ❖ Data visualization and interpretation.
- ❖ Look for outliers or interesting objects.
- ❖ Improve performance of supervised machine learning:
 - ❖ Original features can be correlated and redundant.
 - ❖ Most traditional algorithms cannot handle thousands of features.

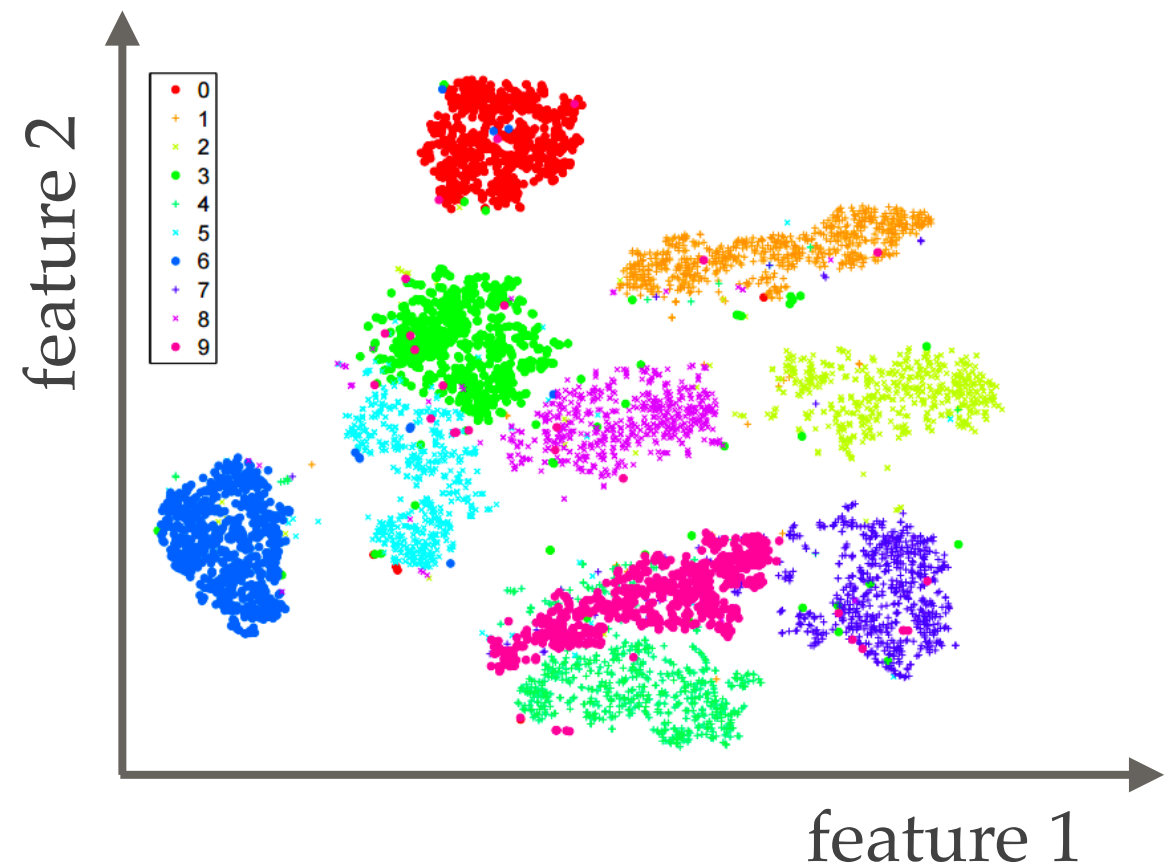
Different use cases

- ❖ Uncover new trends / correlations.
- ❖ Data visualization and interpretation.
- ❖ Look for outliers or interesting objects.
- ❖ Improve performance of supervised machine learning:
 - ❖ Original features can be correlated and redundant.
 - ❖ Most traditional algorithms cannot handle thousands of features.
- ❖ Compressing data (e.g., the Square Kilometre Array; SKA).

Two types of outputs

1. Low-dimensional embedding of our dataset.

This is a common output of all dimensionality reduction algorithms: PCA, ICA, NNMF, LLE, SOM, tSNE, UMAP, etc.



2. Prototypes or Eigen-vectors.

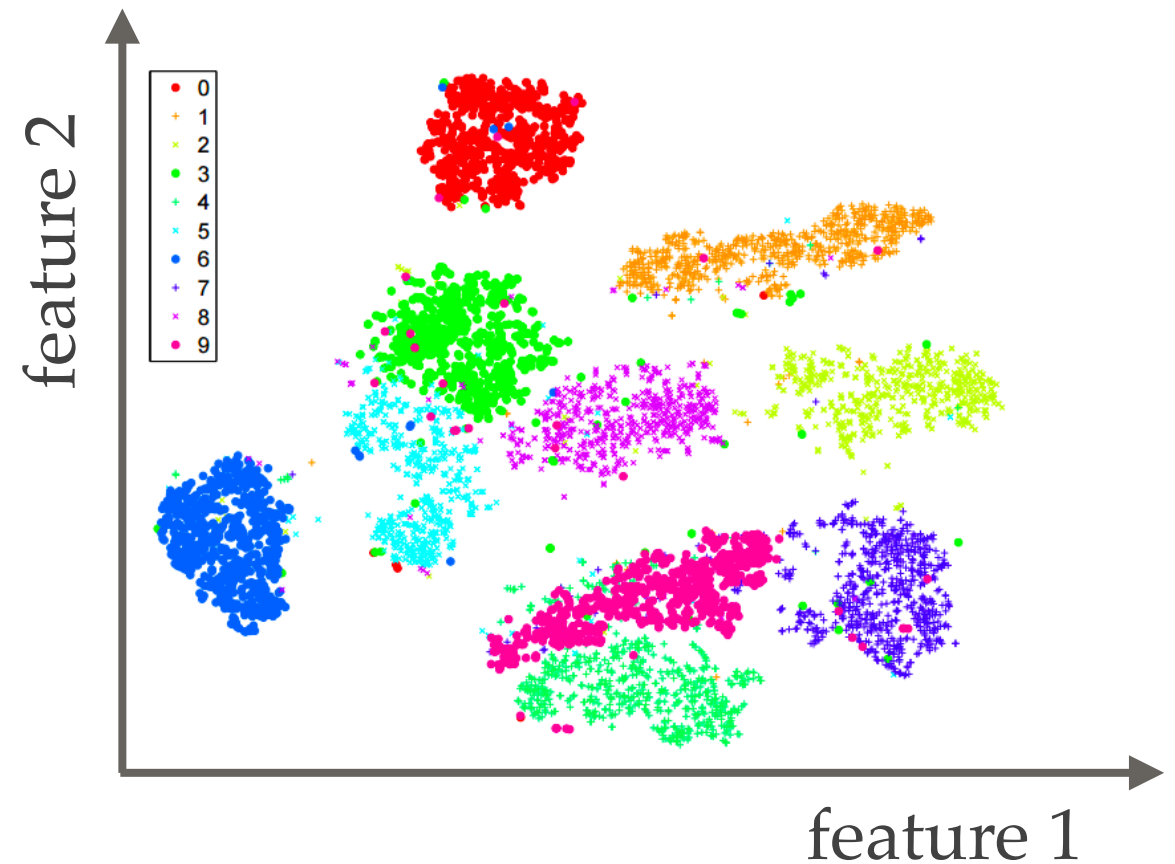
Along with the low-dimensional embedding, this is the output of: PCA, ICA, NNMF, and SOM.



Two types of outputs

1. Low-dimensional embedding of our dataset.

This is a common output of all dimensionality reduction algorithms: PCA, ICA, NNMF, LLE, SOM, tSNE, UMAP, etc.



2. Prototypes or Eigen-vectors.

Along with the low-dimensional embedding, this is the output of: PCA, ICA, NNMF, and SOM.

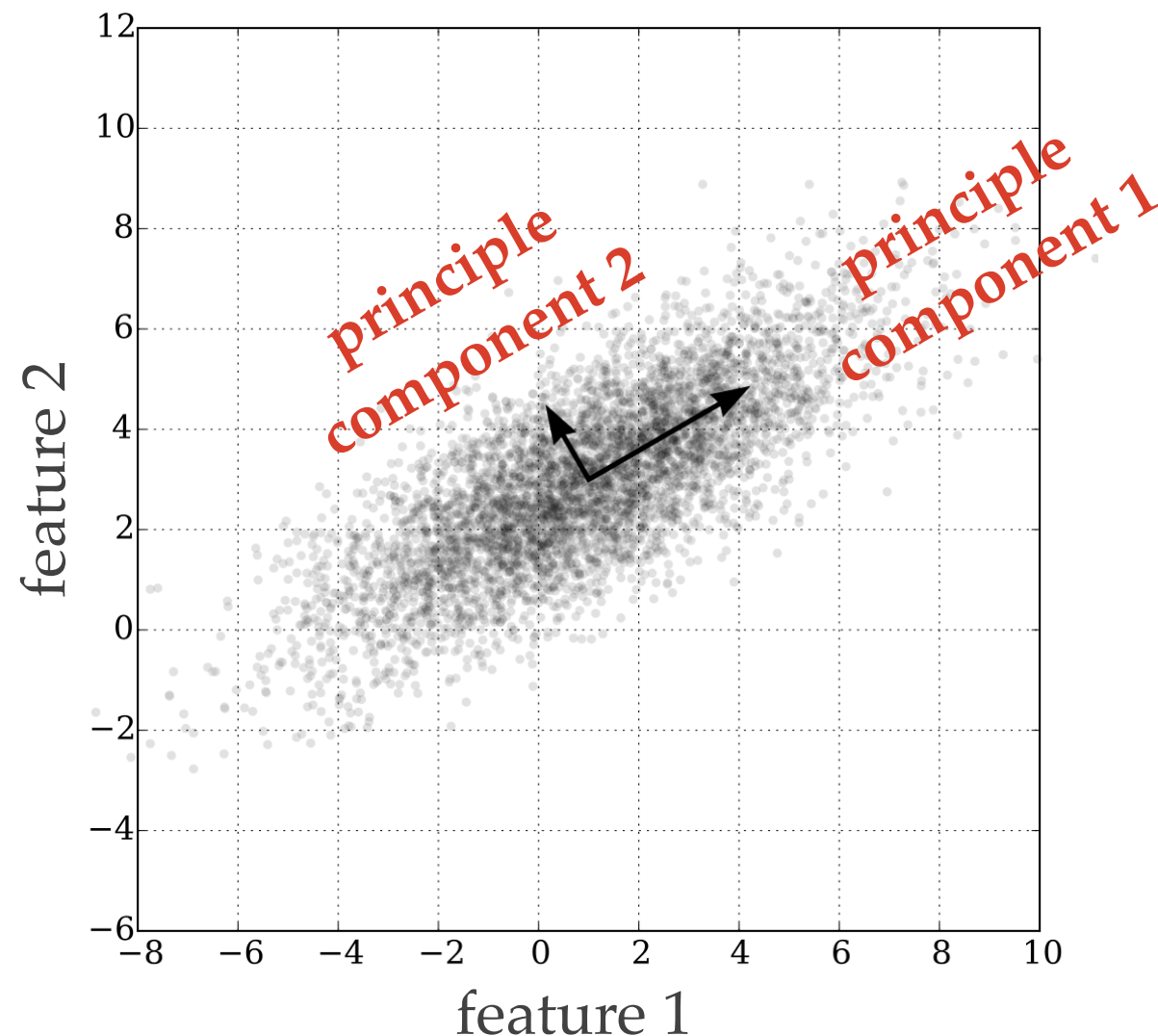


Can be useful when trying to interpret the resulting embedding!

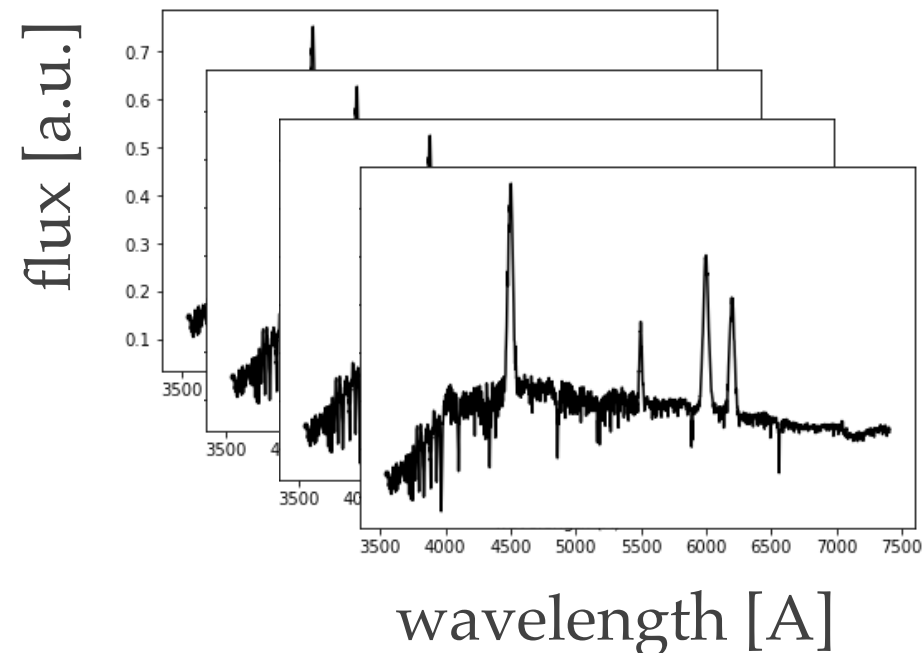


Principal Component Analysis (PCA)

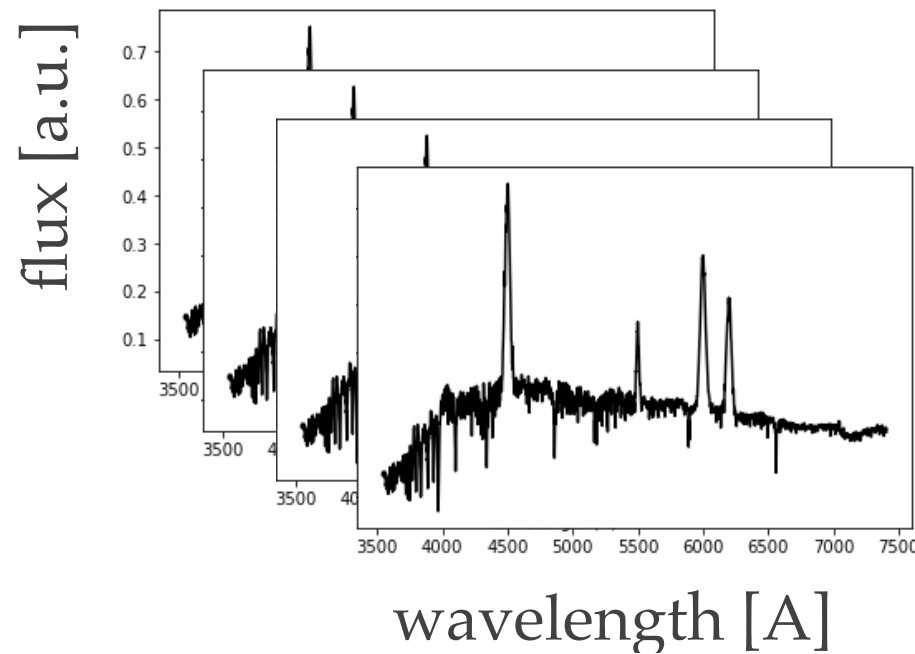
- ❖ Linear transformation of the input data into a new coordinate system, defined by the *principle components* where the axes (principal components) capture the directions of maximum variance. By keeping only the top components, PCA reduces dimensionality while preserving as much of the data's variability as possible.



Principal Component Analysis (PCA)



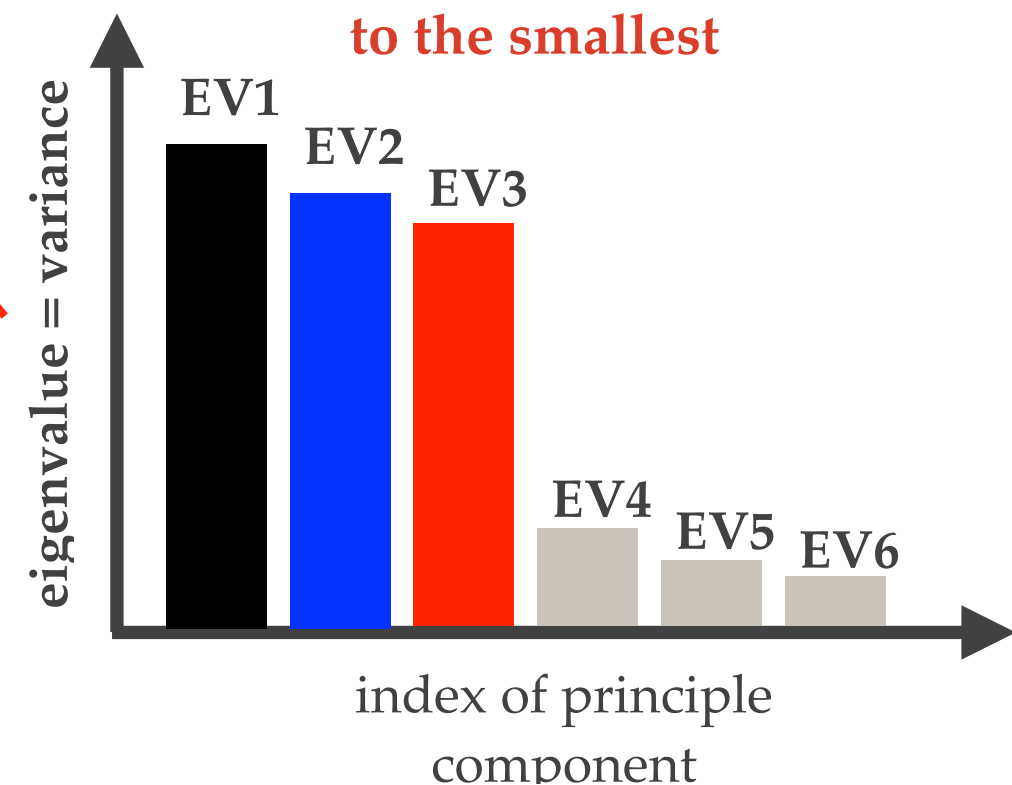
Principal Component Analysis (PCA)



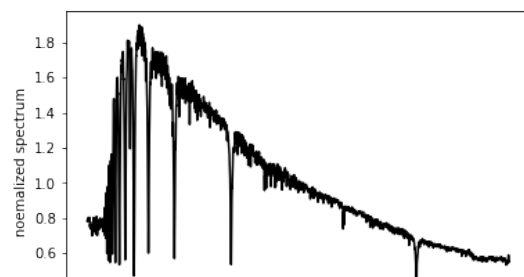
The matrix of the
eigenvectors

$$Z^T Z = P D P^{-1}$$

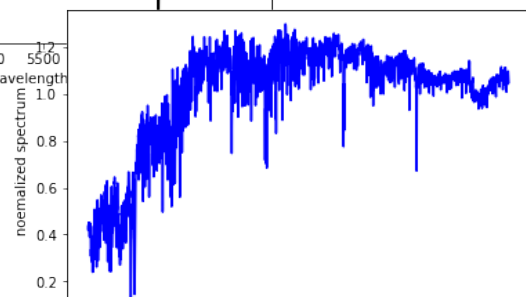
Diagonal matrix containing
eigenvalues ordered from the largest
to the smallest



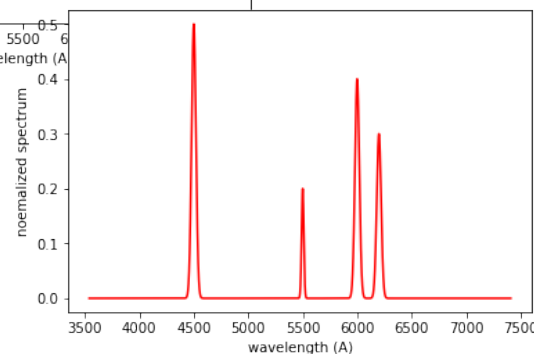
PC1



PC2

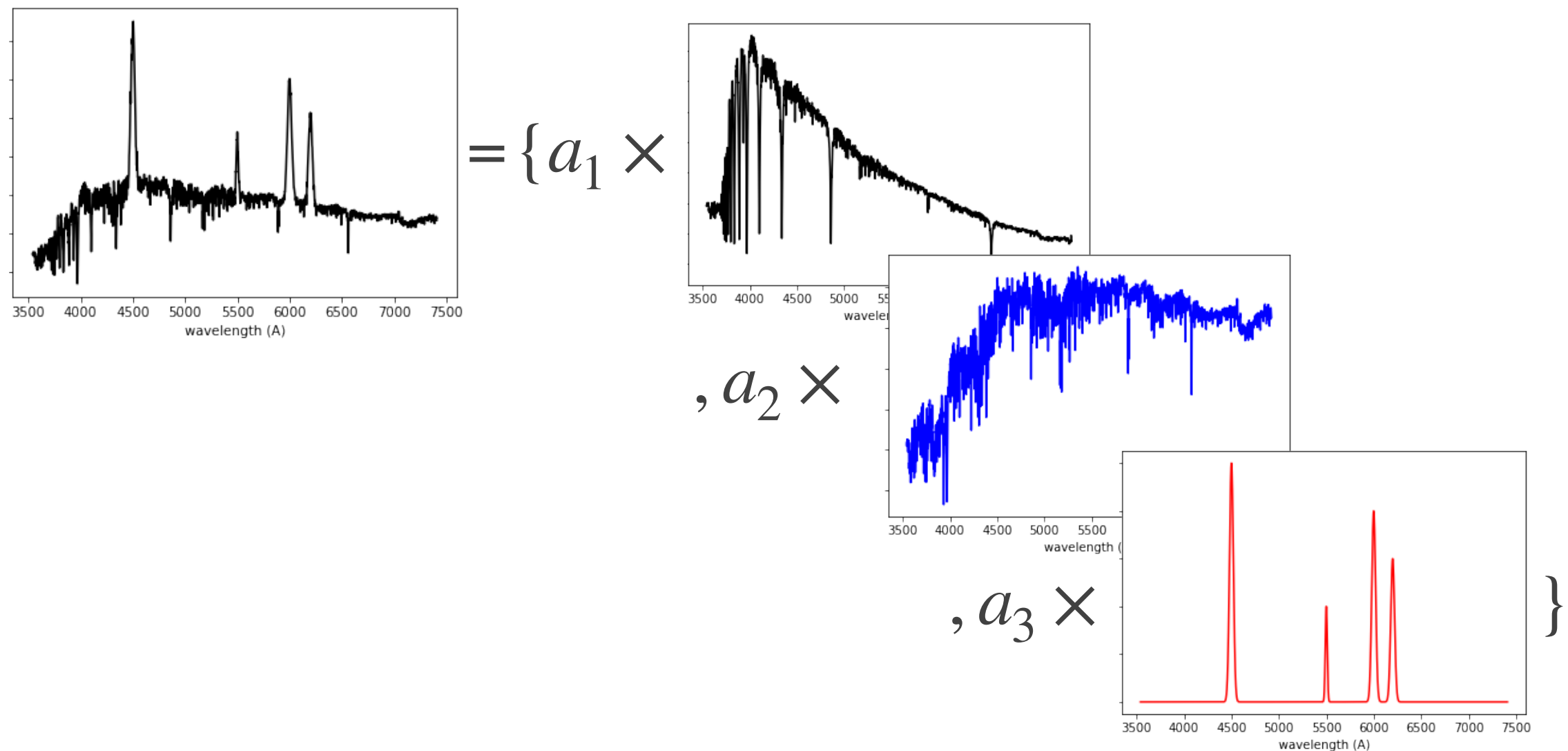


PC3



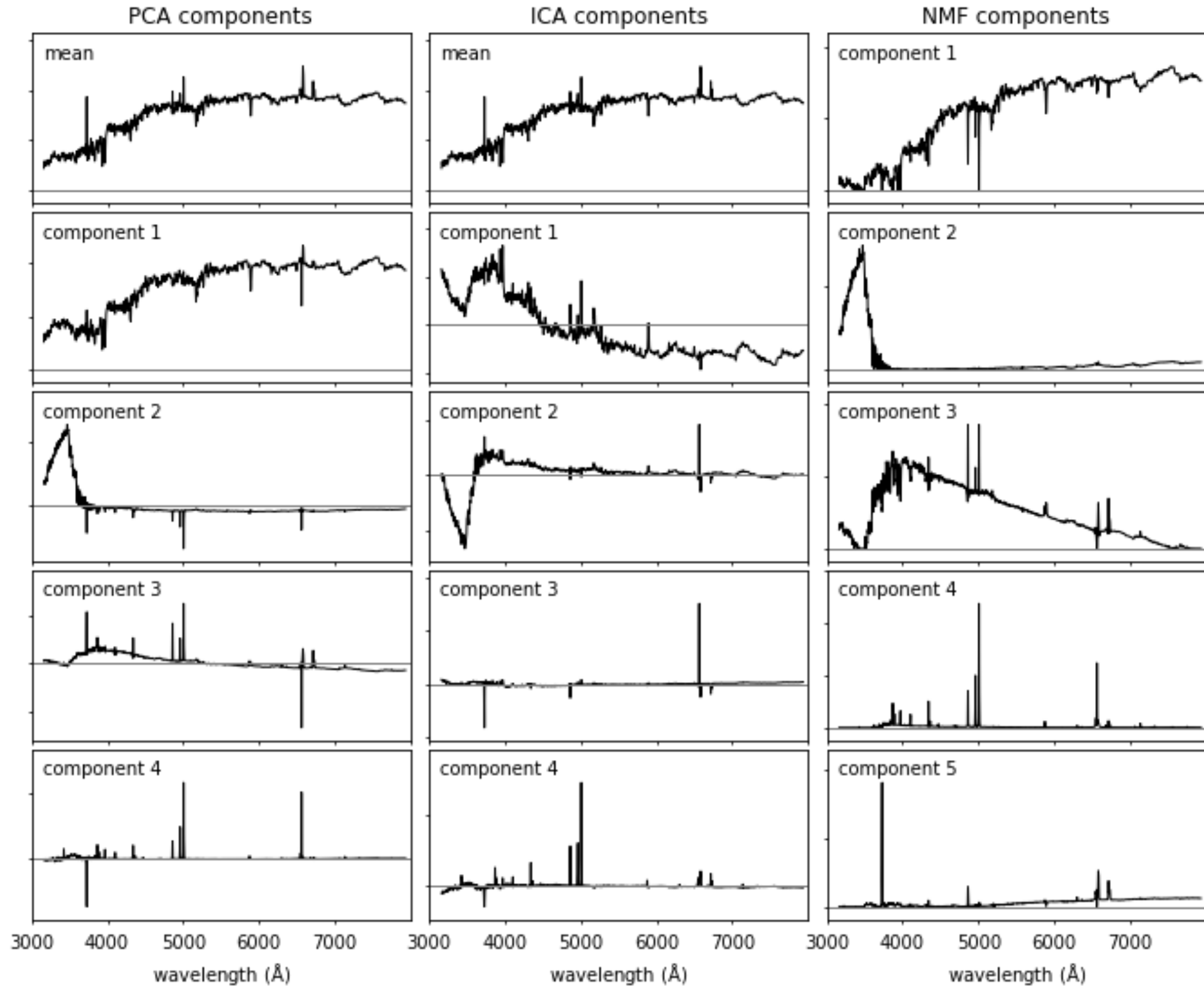
Principal Component Analysis (PCA)

Each object is a linear combination of the principle components.



Now every spectrum is described using 3 numbers, and can be visualized in 3D.

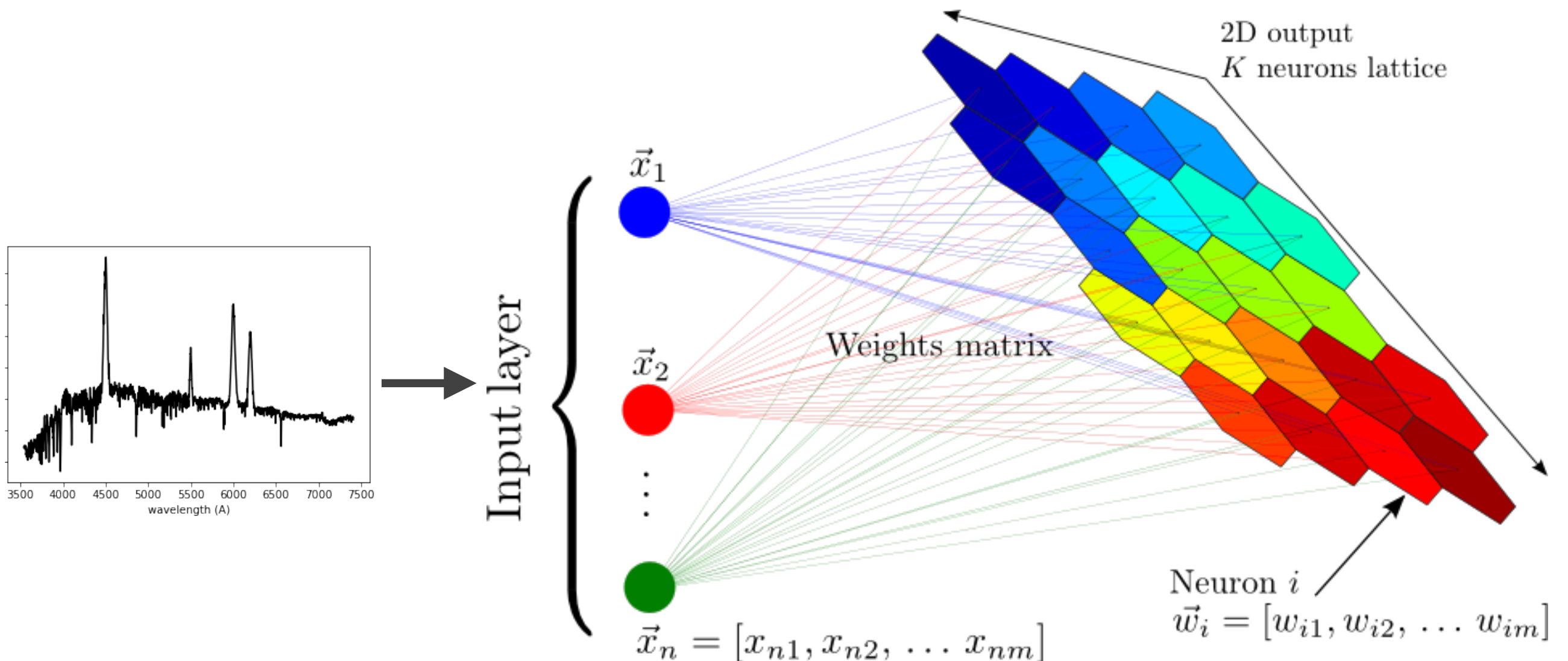
ICA and NNMF



astroML [link](#) (very helpful resource!)

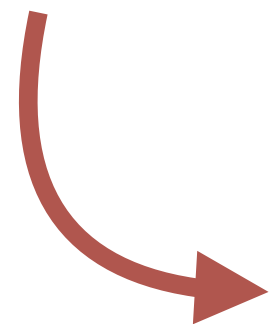
Self-Organizing Maps (SOM)

- ❖ Competitive learning: during training, each input adjusts the weights of the closest (best-matching) node and its neighbors, causing similar inputs to cluster together on the map.
- ❖ The output is a 2D map of nodes which represent the objects in the sample, with properties varying smoothly across the map.



Non-linear dimensionality reduction algorithms

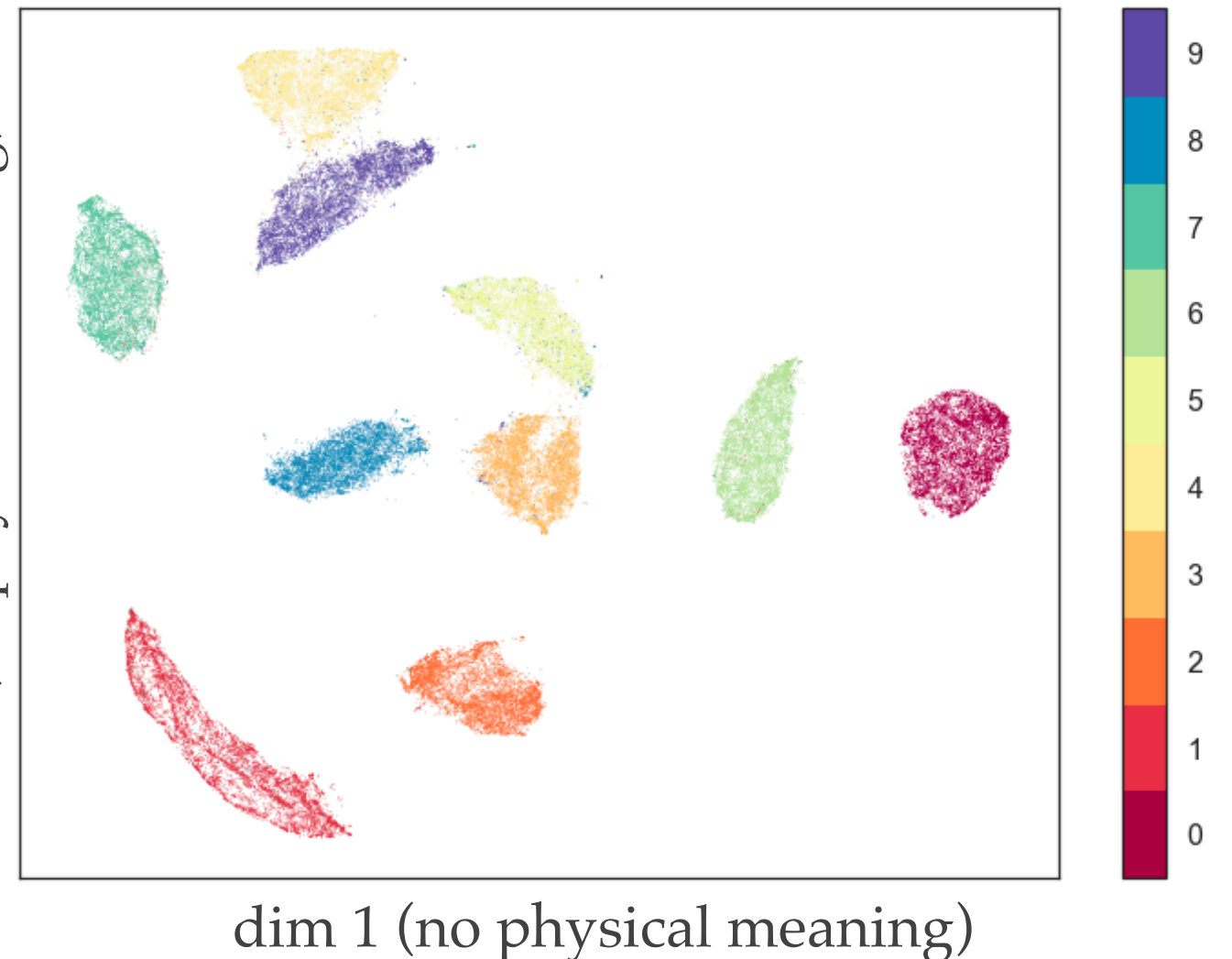
MNIST dataset: 28x28 features per image



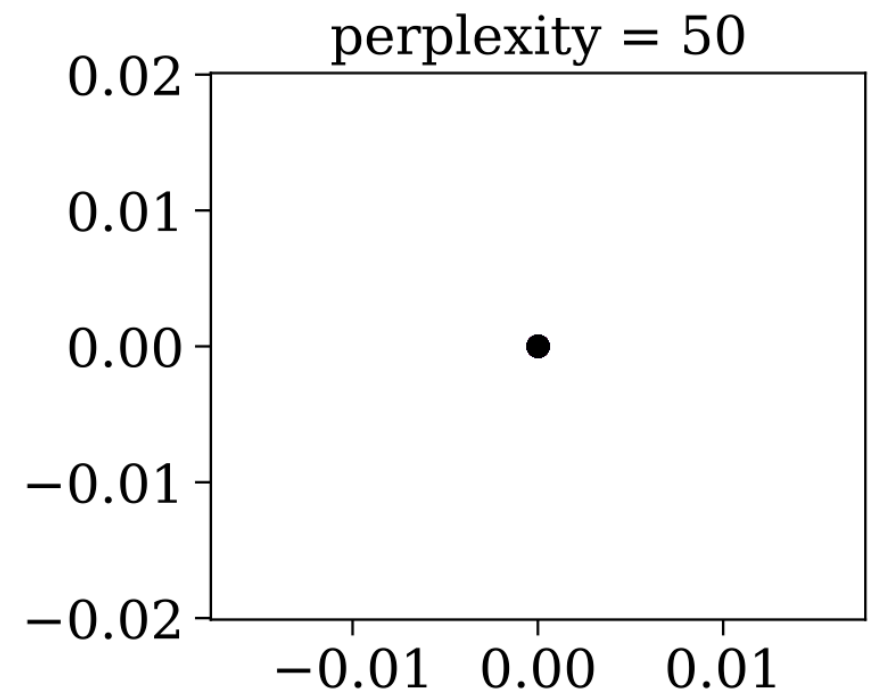
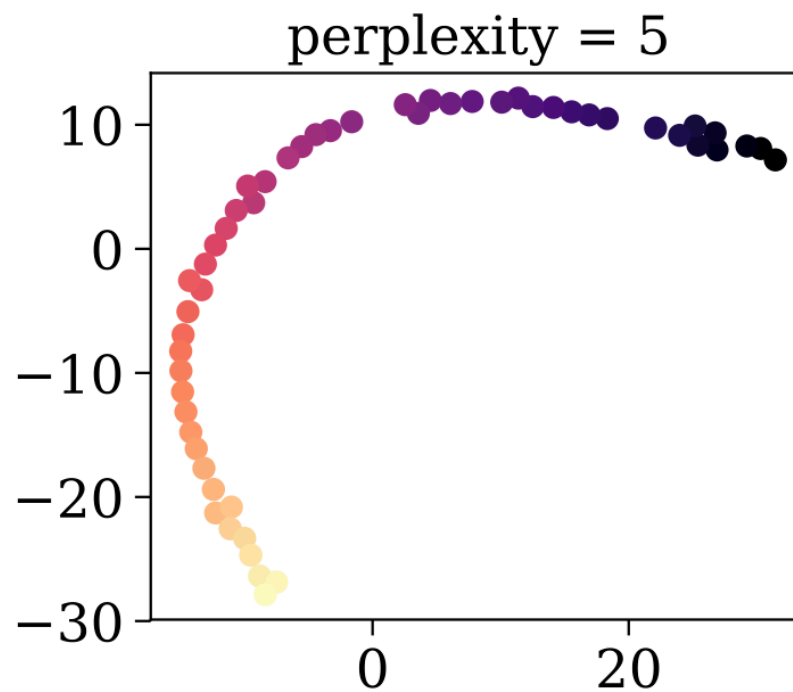
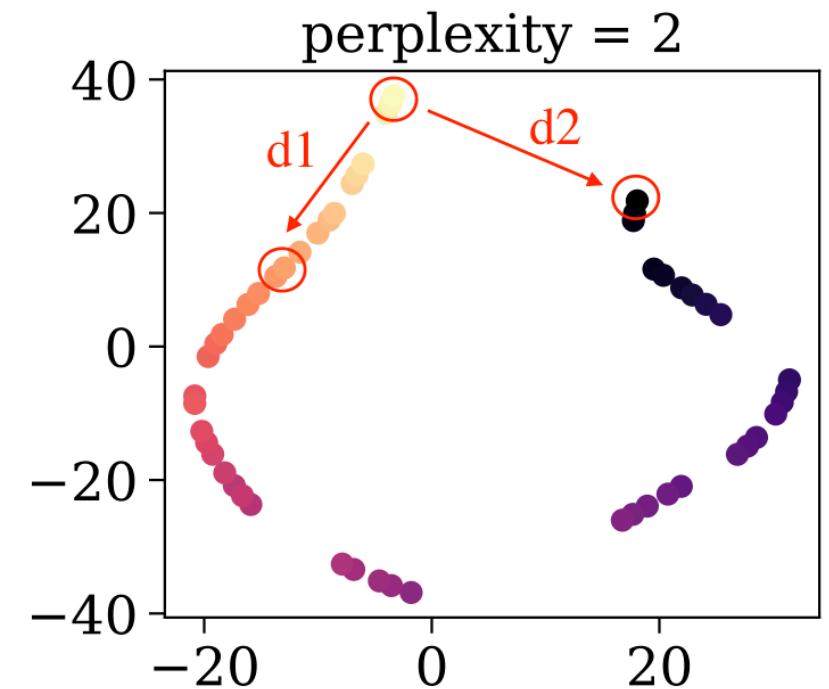
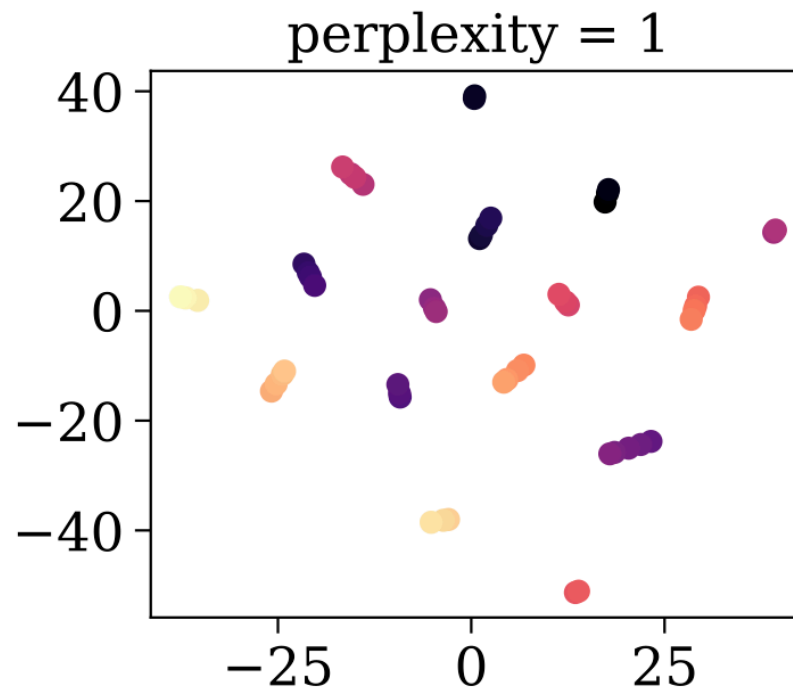
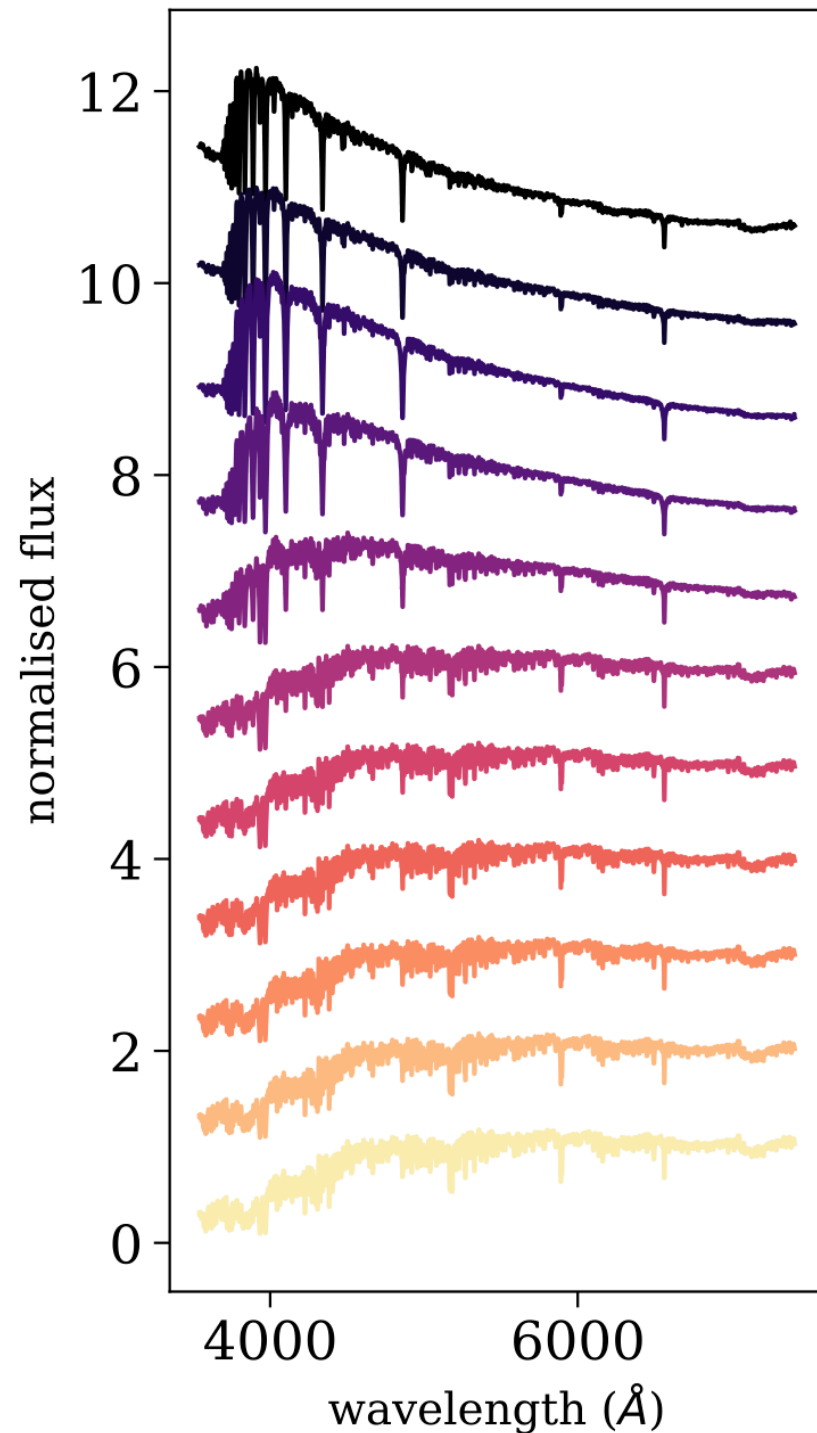
tSNE /
UMAP



dim 2 (no physical meaning)



tSNE hyper-parameter variation

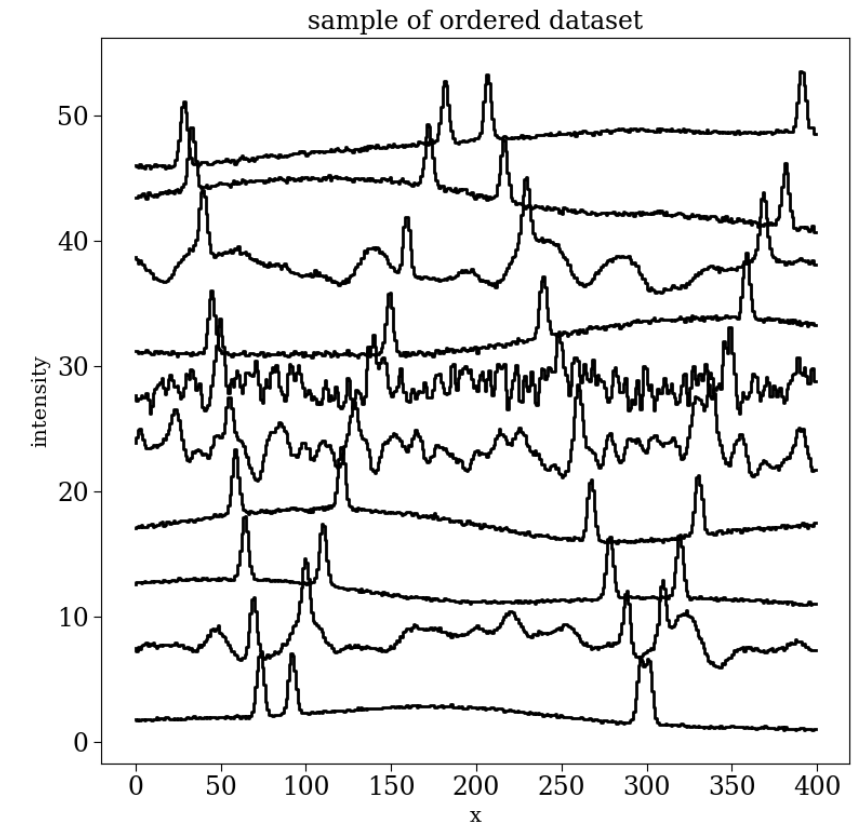
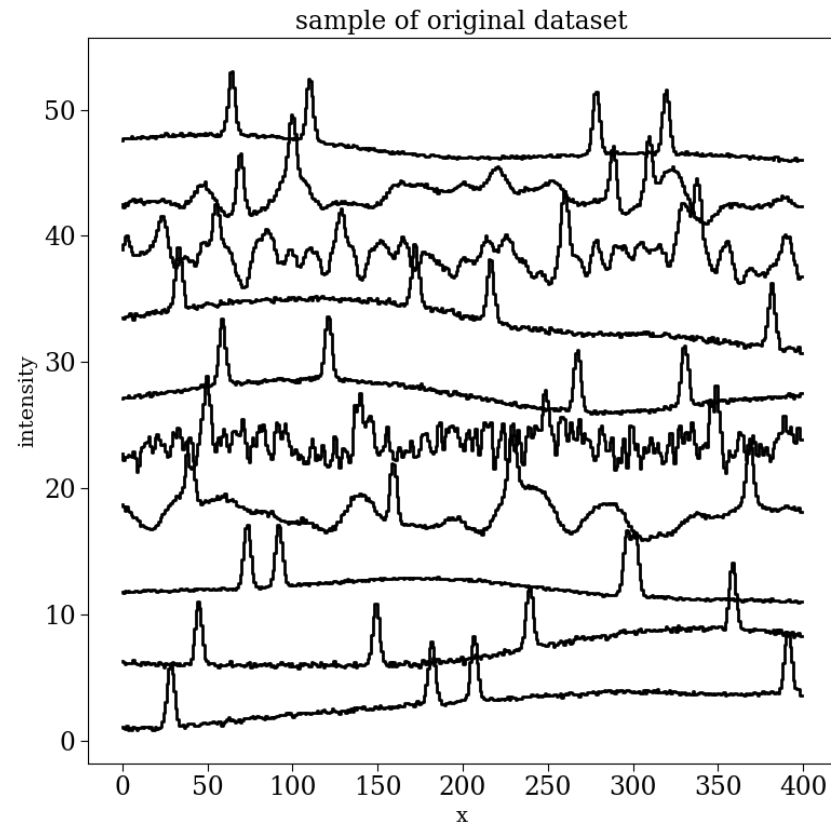


The hyper-parameters in t-SNE and UMAP control the balance between local and global structure preservation—such as perplexity in t-SNE or number of neighbors and minimum distance in UMAP—affecting how tightly points cluster and how the overall layout unfolds.

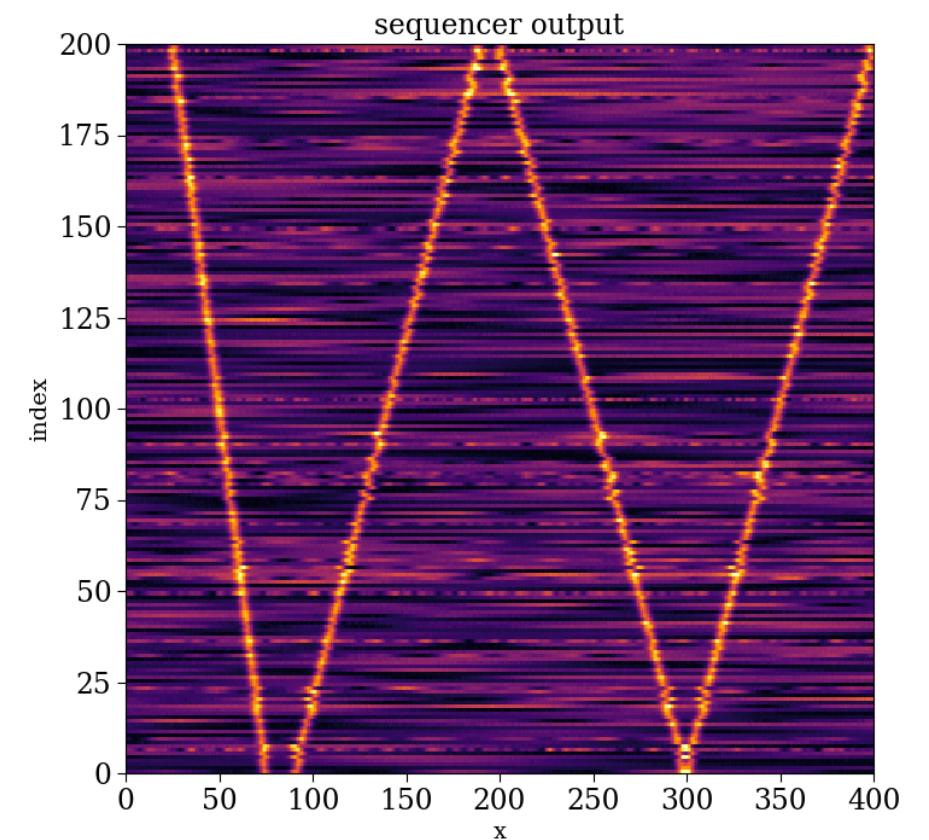
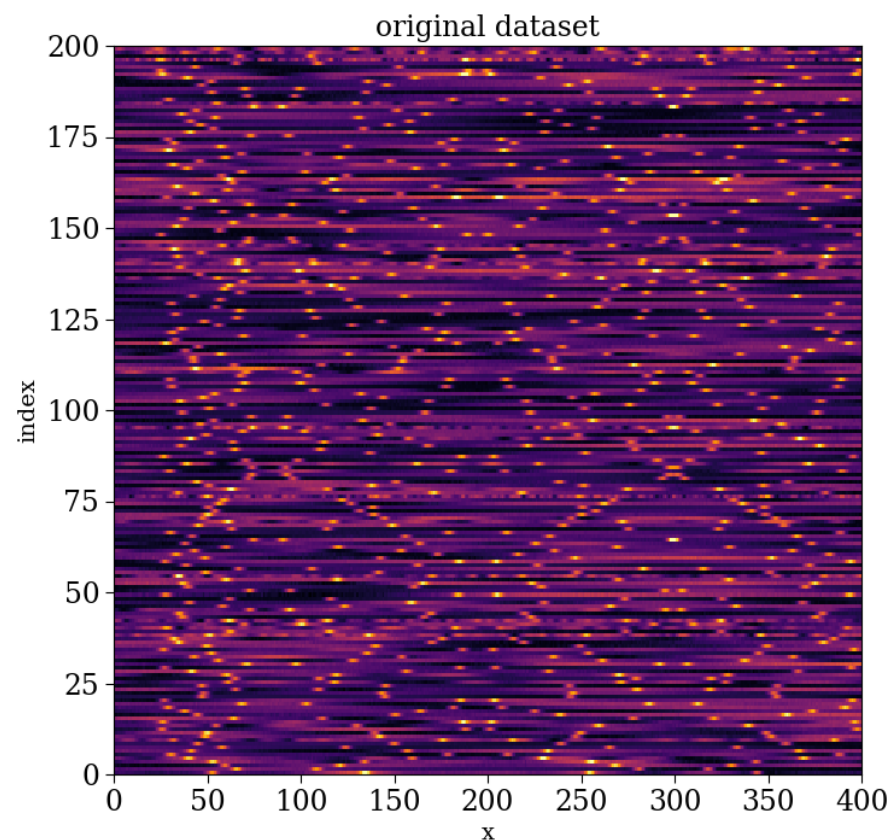
Be careful when interpreting the 2D maps!

- ❖ <https://distill.pub/2016/misread-tsne/>
- ❖ <https://pair-code.github.io/understanding-umap/>

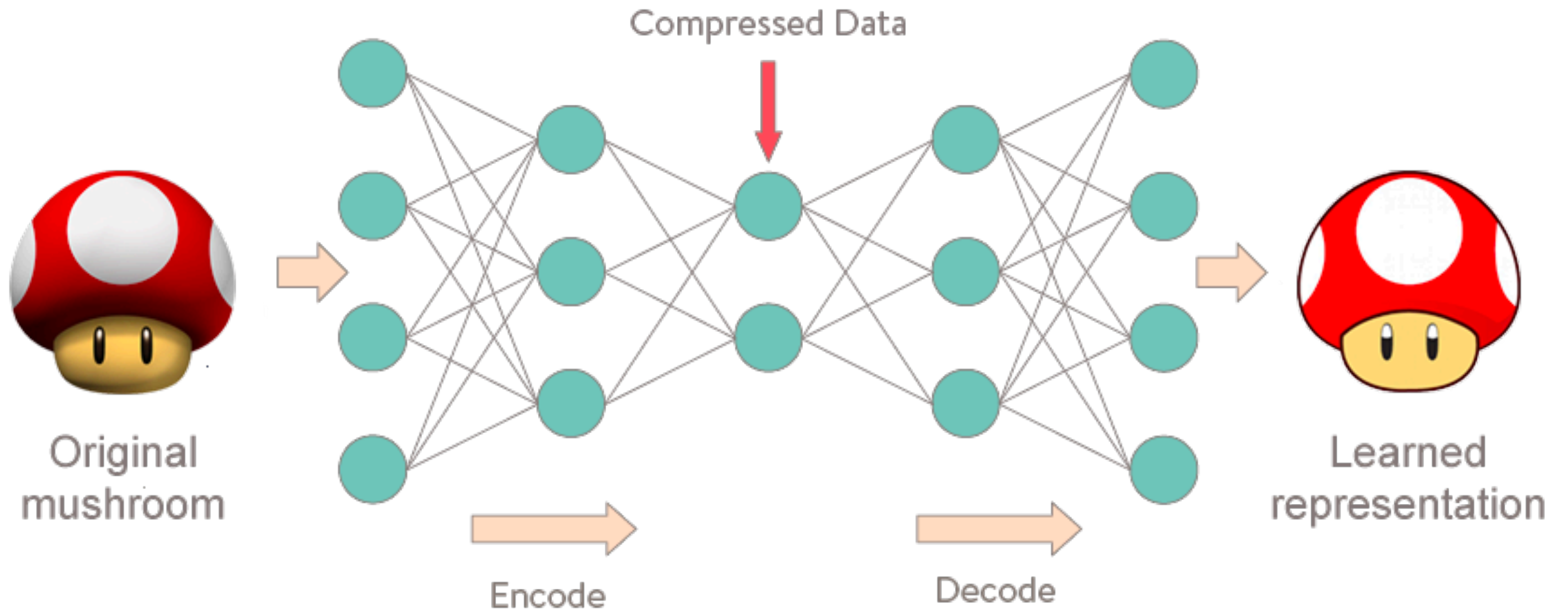
The Sequencer



Sequencer
→



Latent/embedding spaces



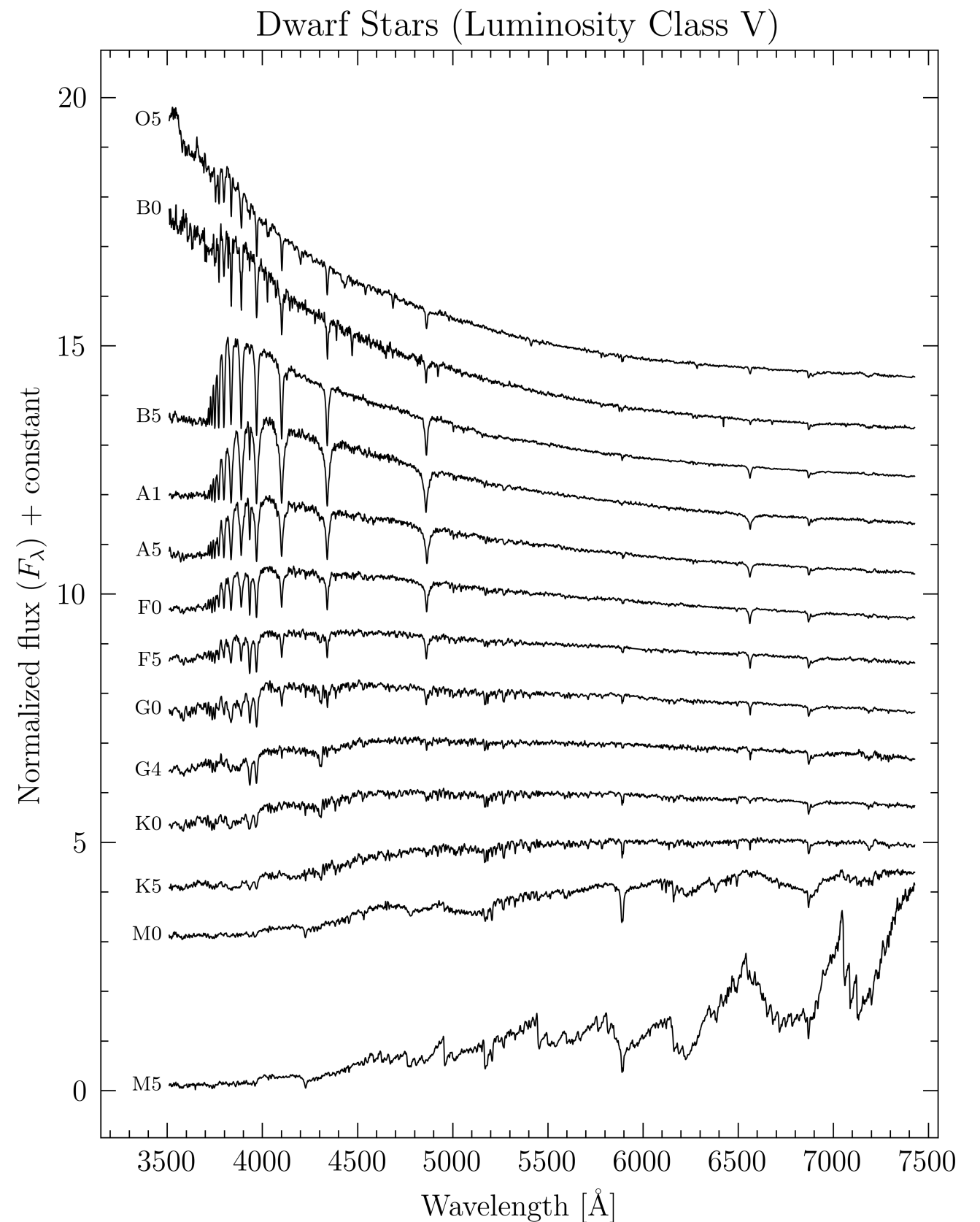
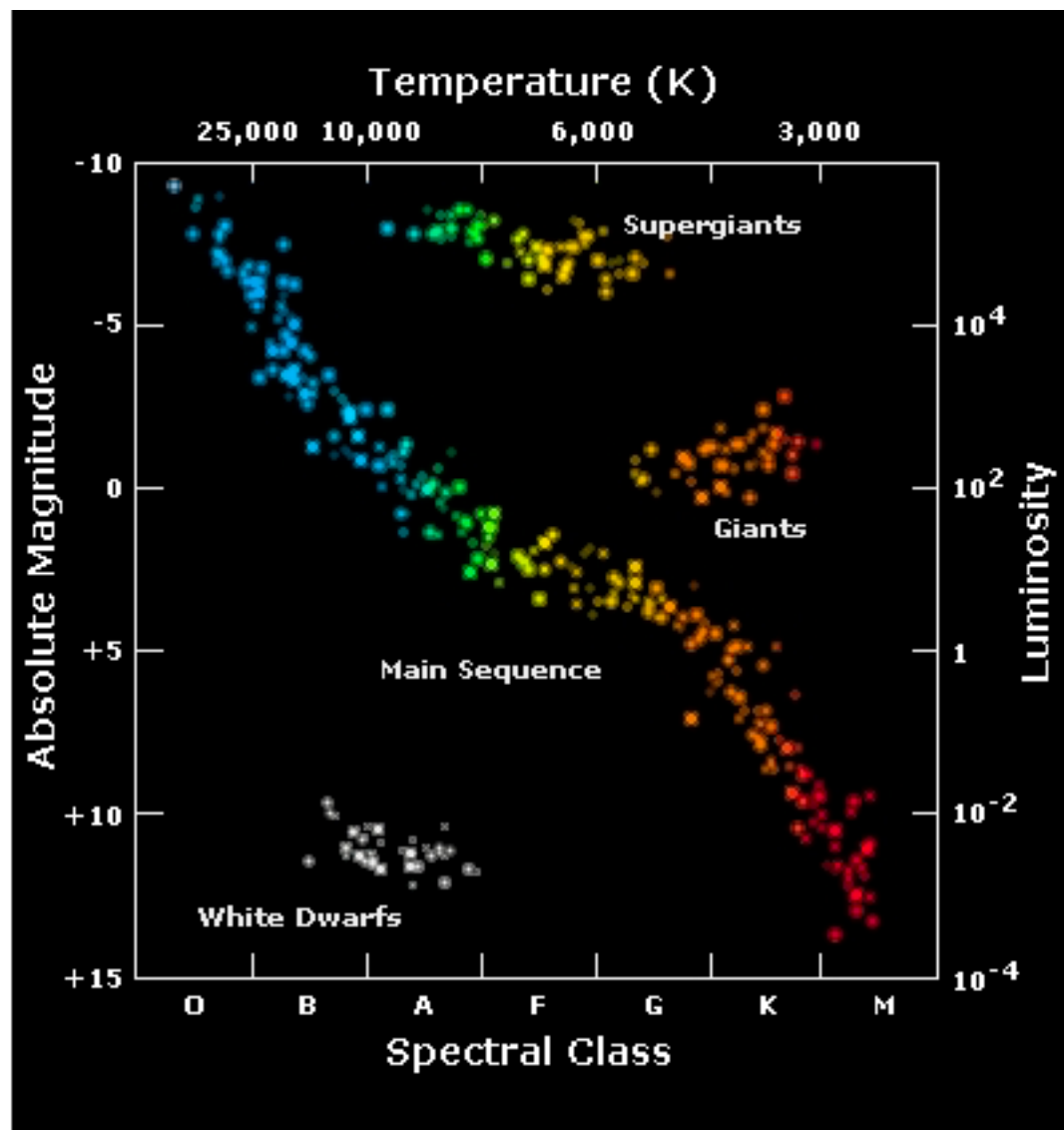
Clustering algorithms

Clustering is a key process in data exploration

- ❖ Clustering is one of the first steps in data exploration. Using clustering, we may try to answer one of the most basic questions we can ask — “what is there in my dataset?”.

Clusters in astronomy

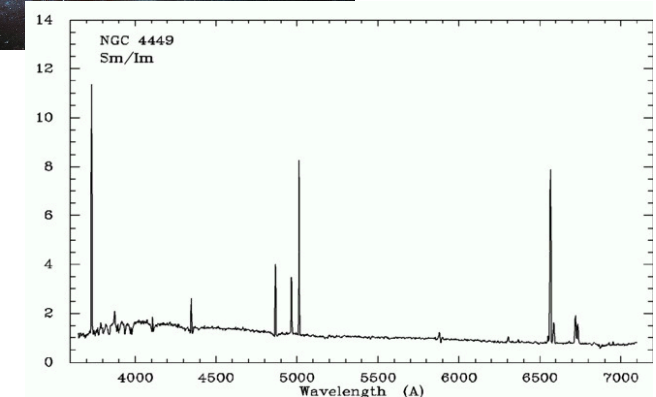
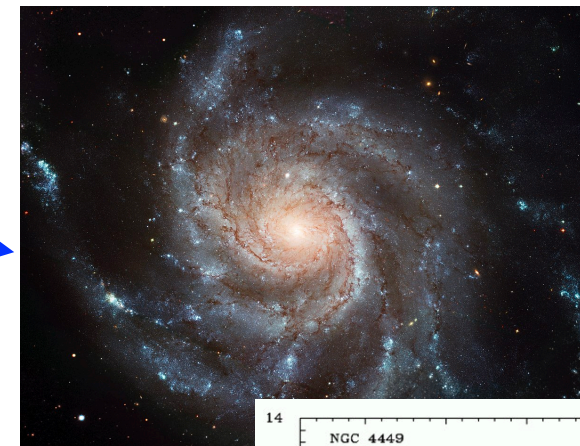
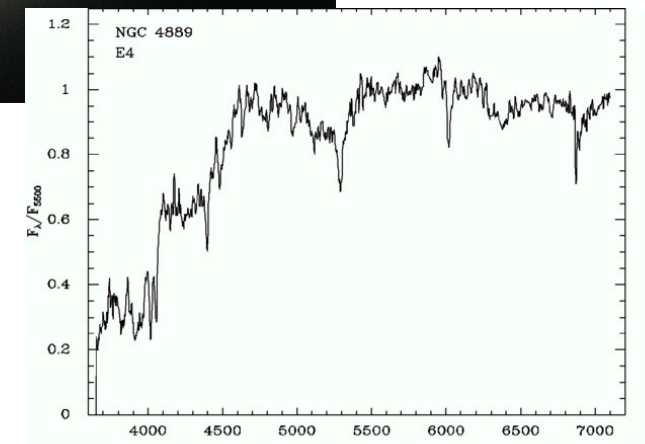
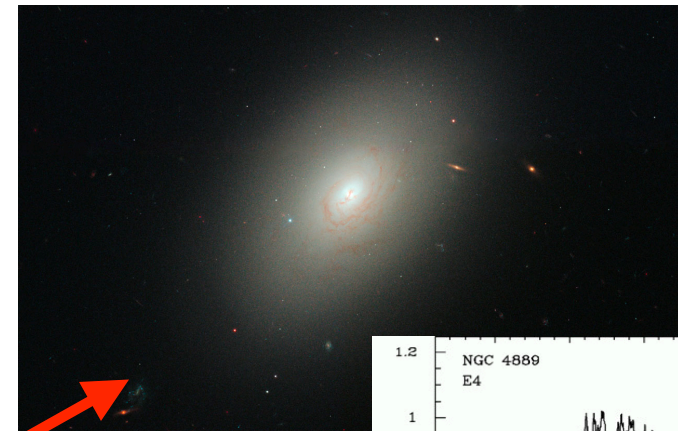
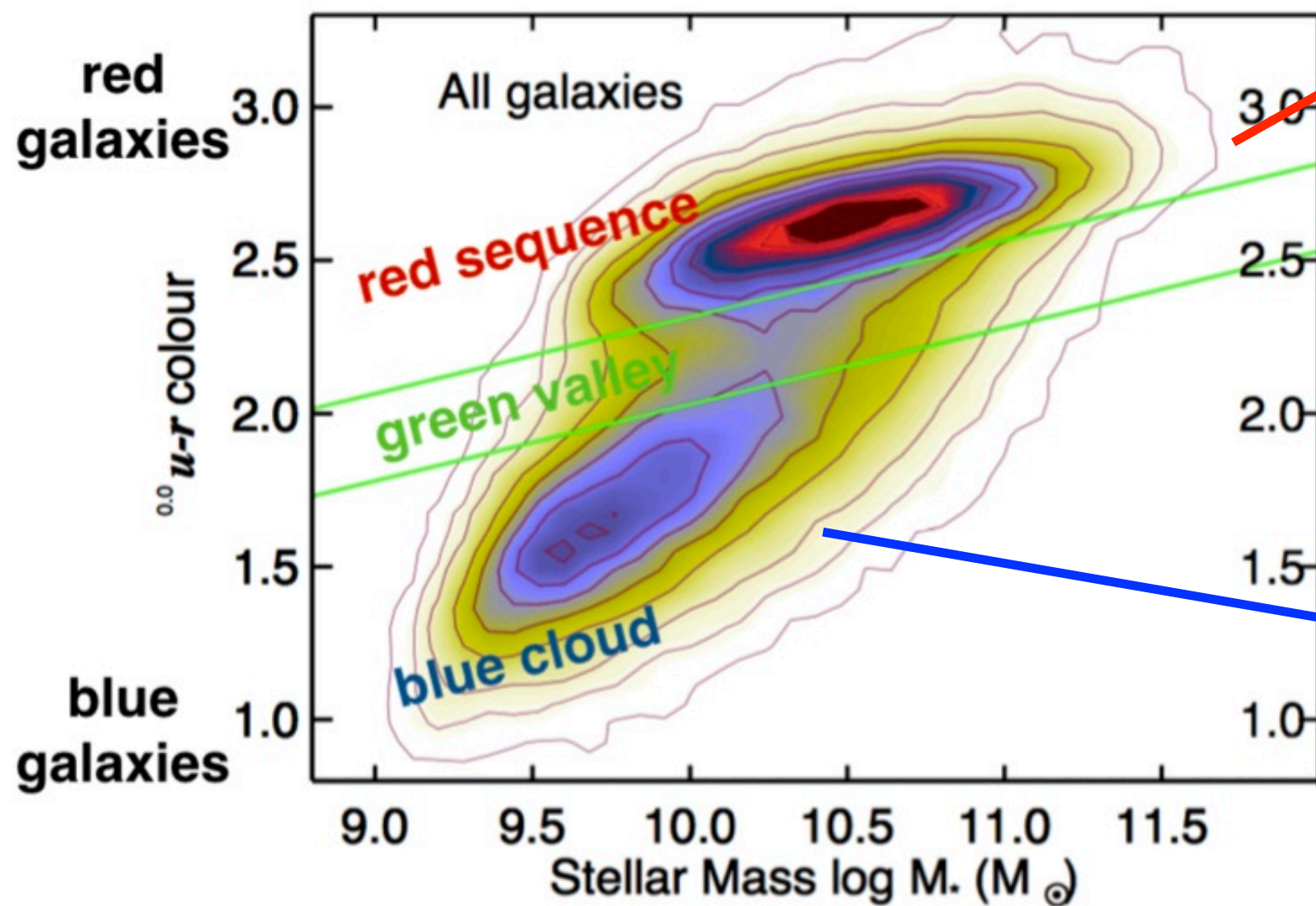
1. Stellar spectral classes



Clusters in astronomy

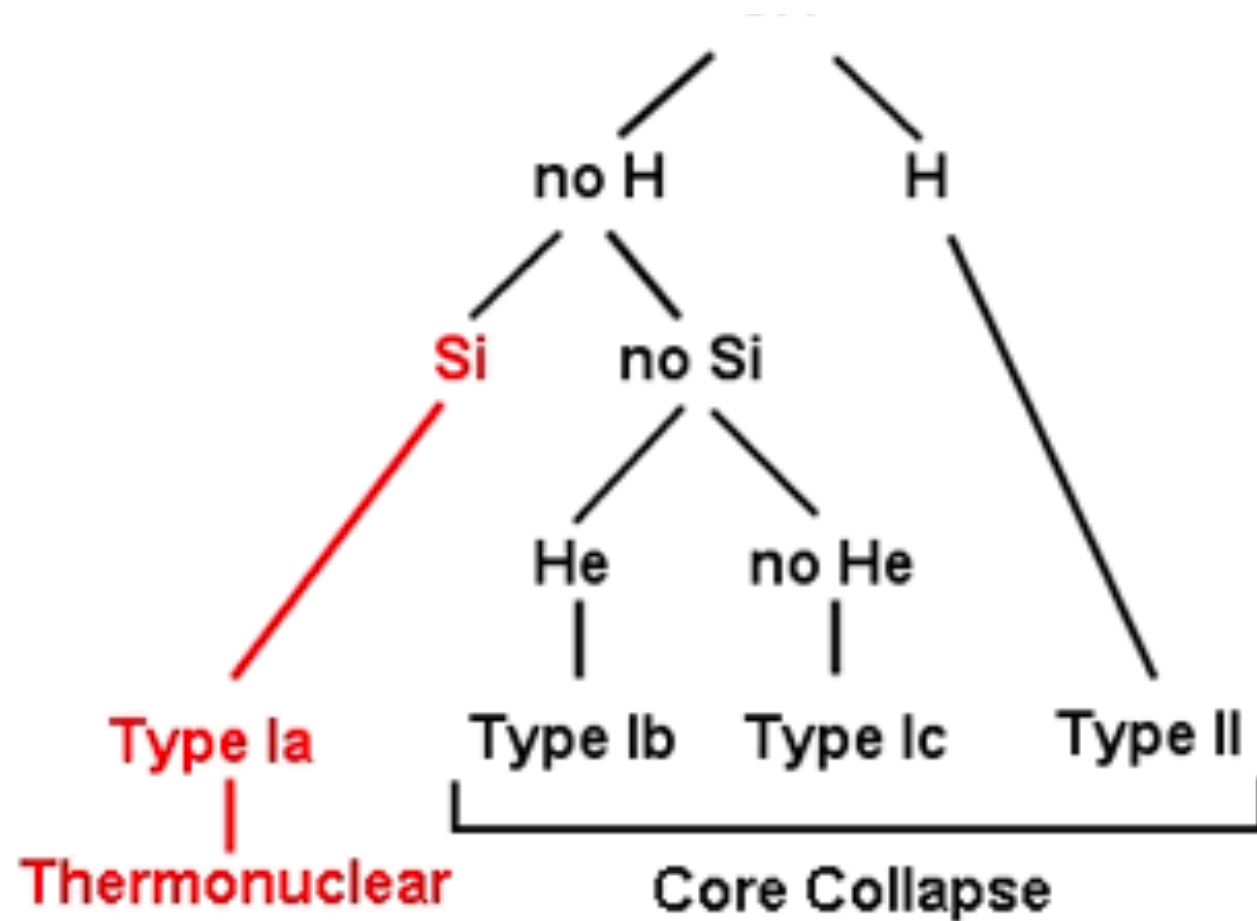
2. Galaxy bimodality

Taken from: Schawinski et al. (2014)

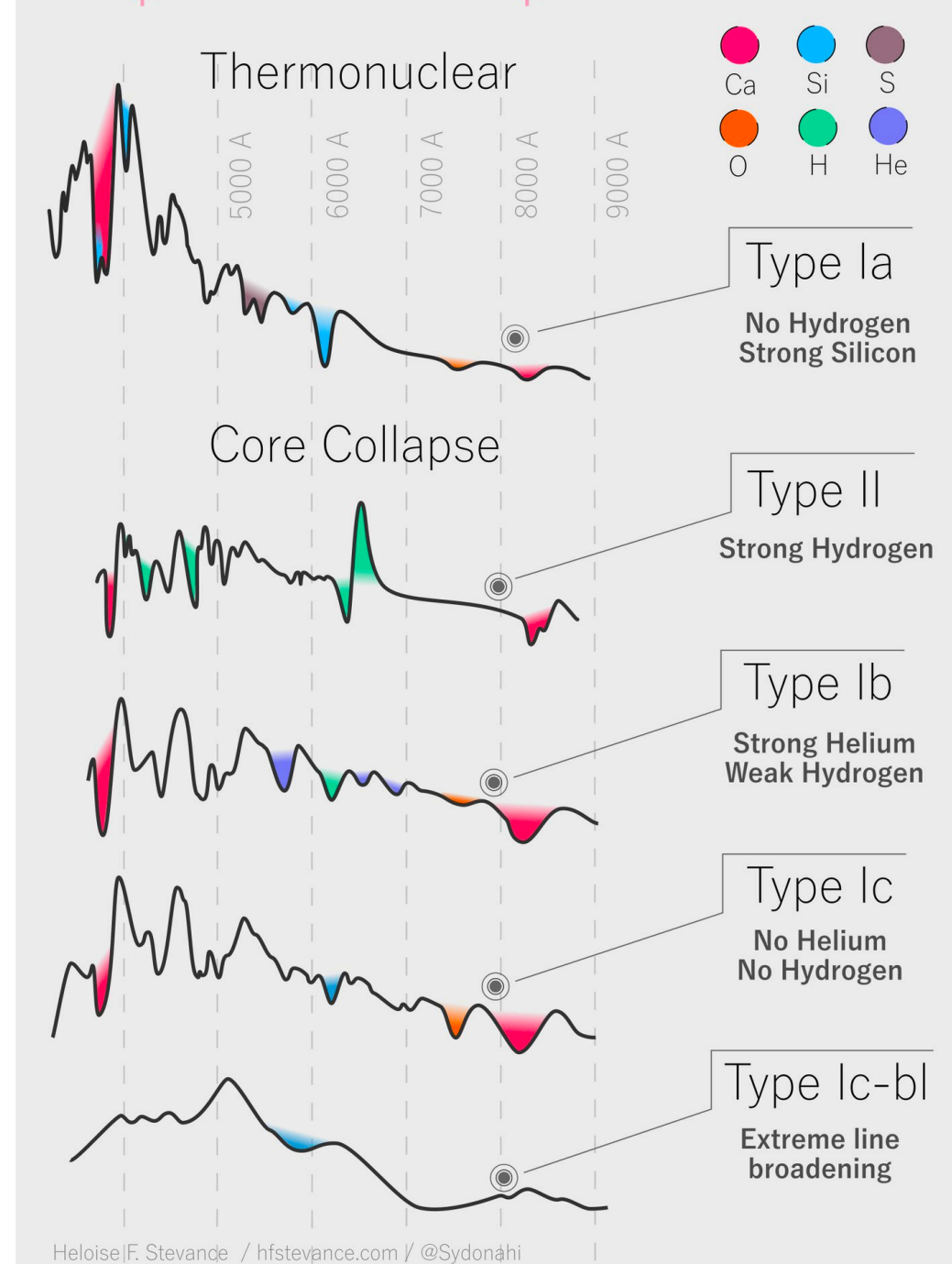


Clusters in astronomy

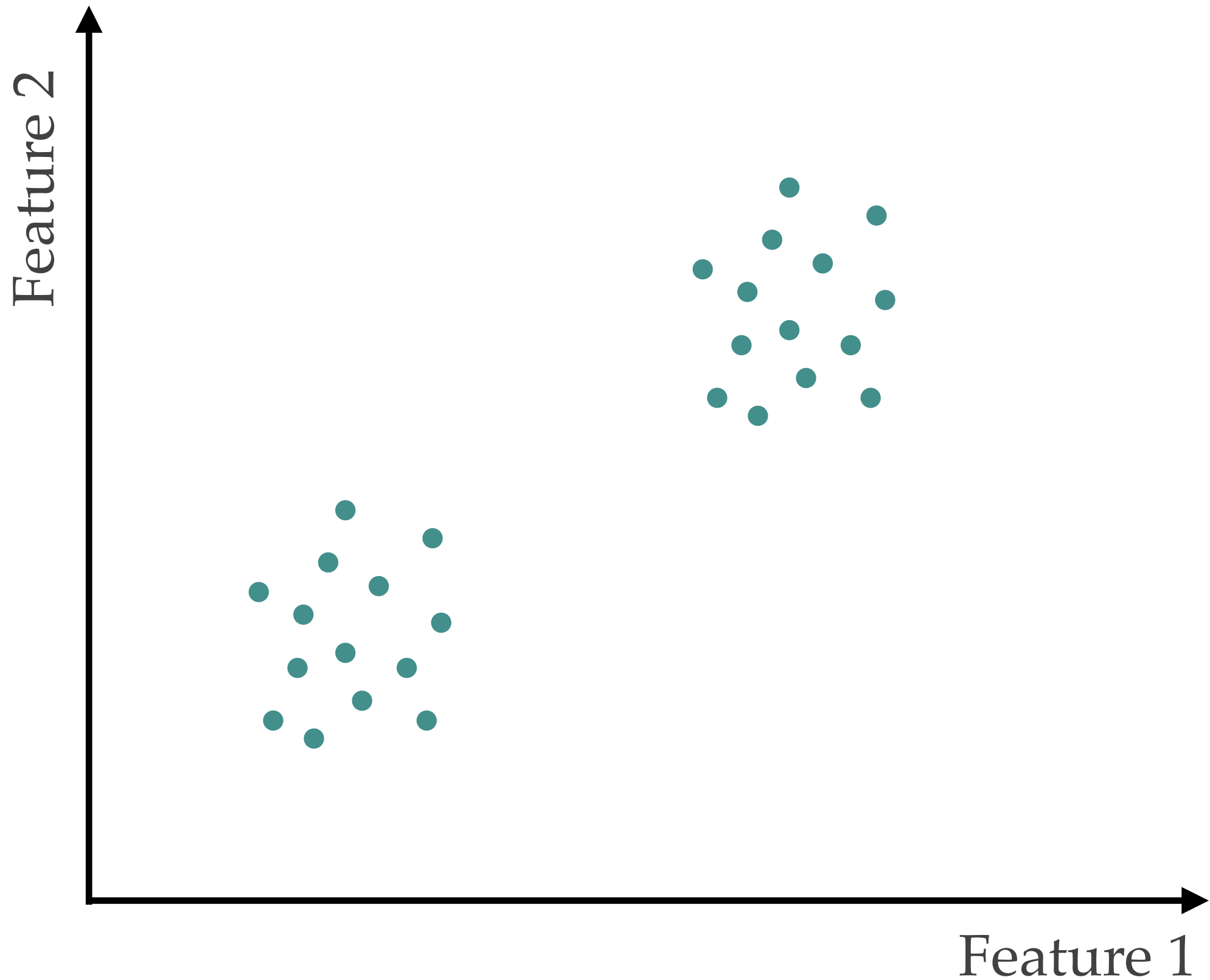
3. Supernova classes: type Ia and type II supernovae



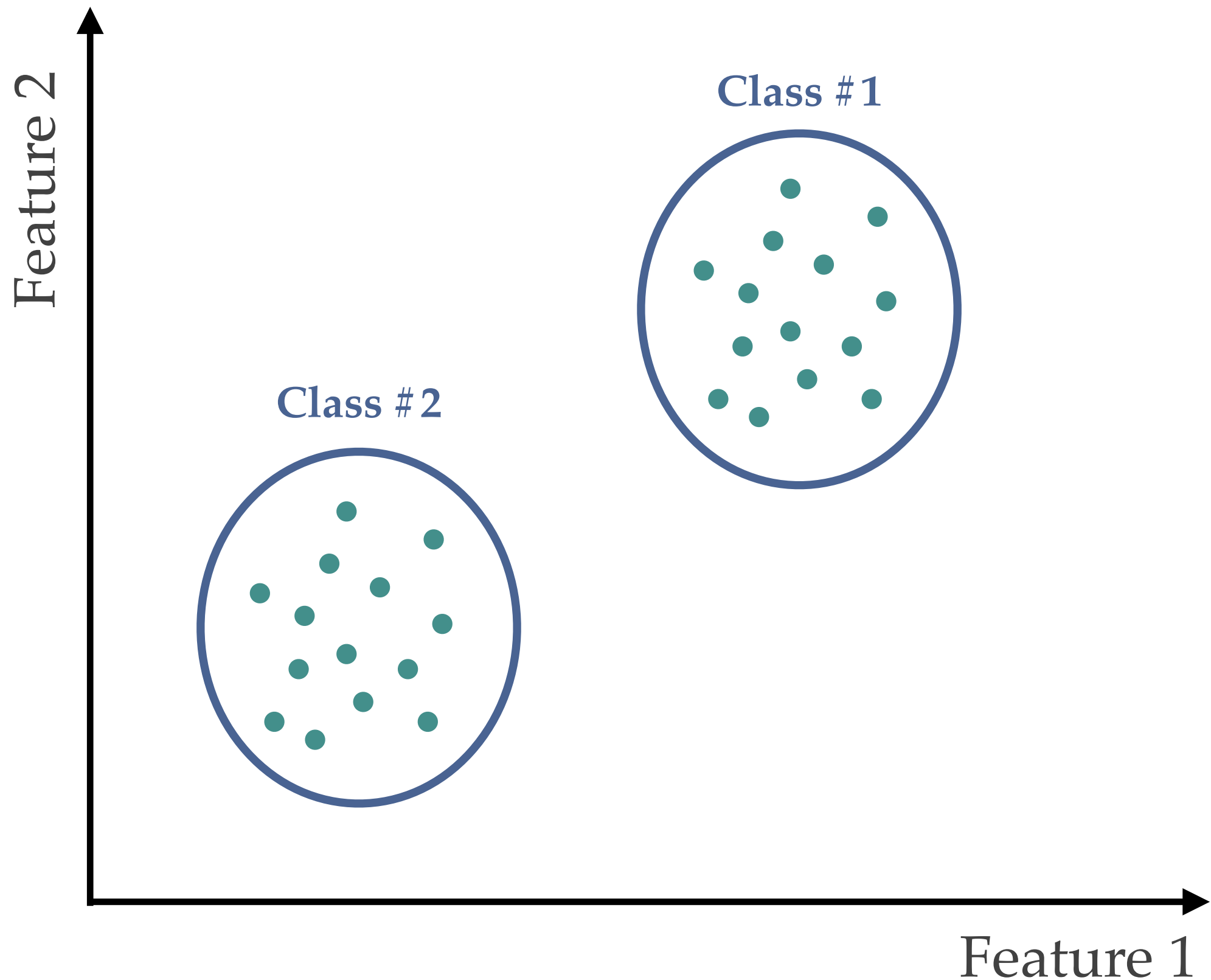
Supernova Spectra



Clustering



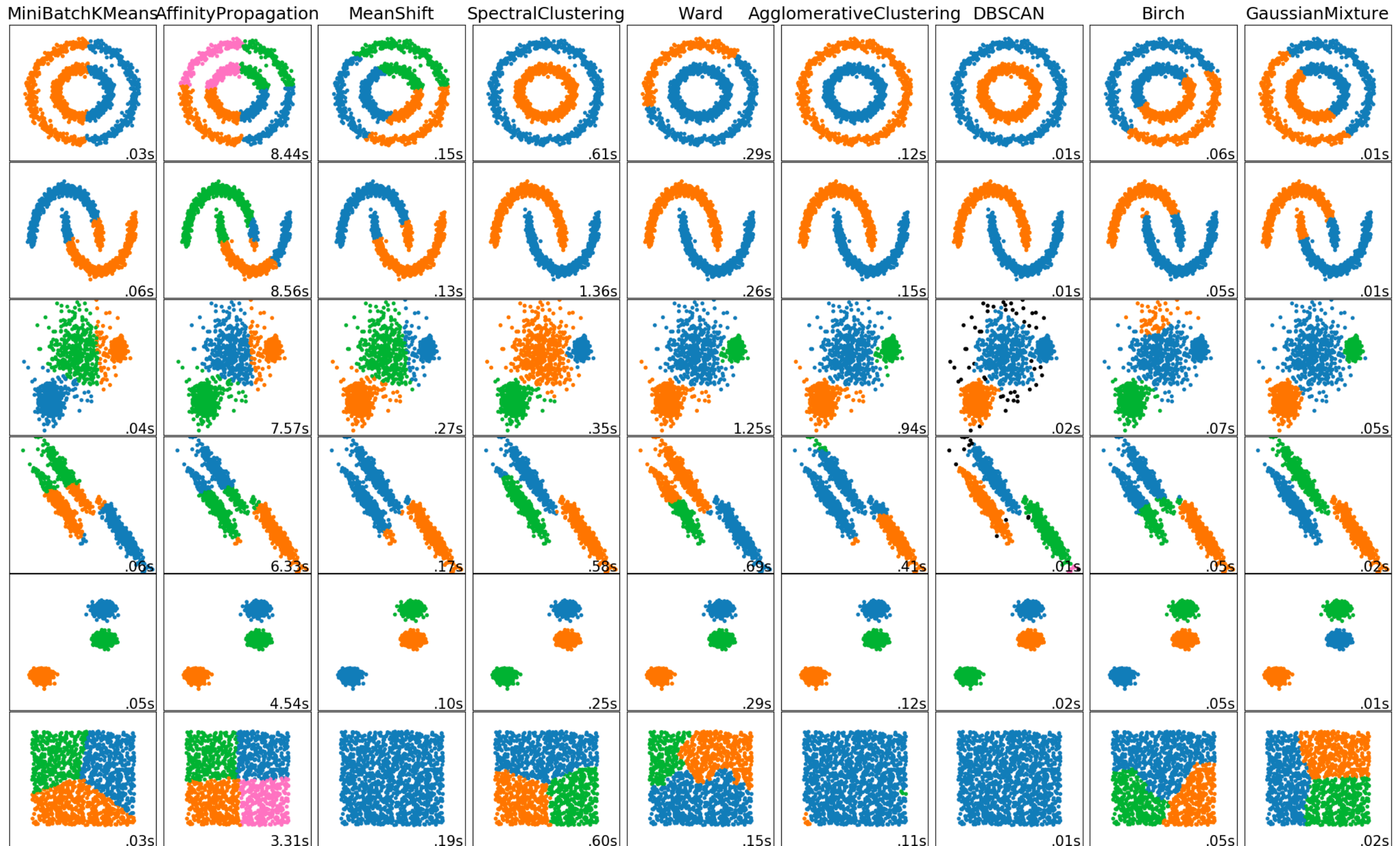
Clustering



Clustering

From Scikit-learn's example gallery:

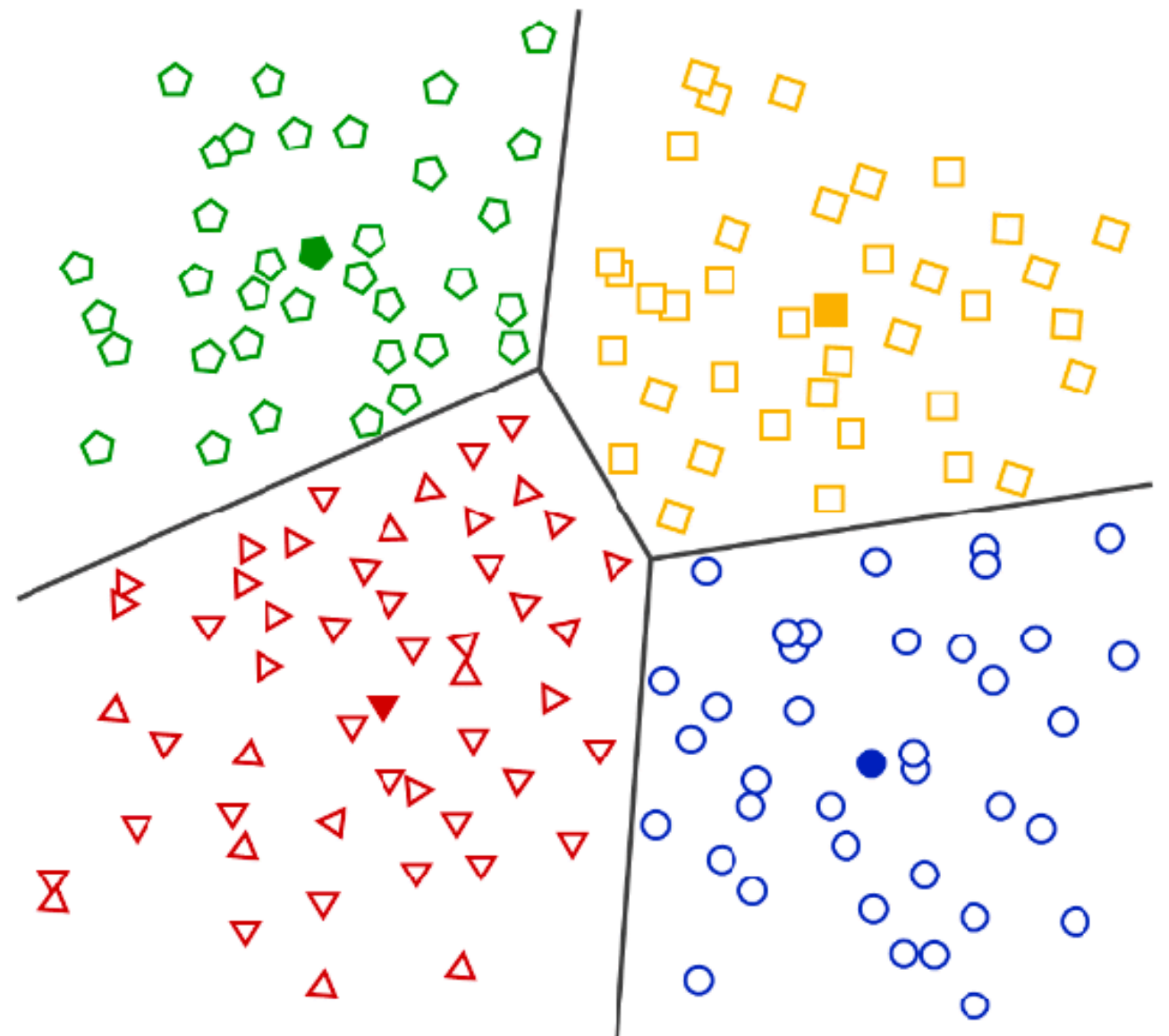
see [this](#) comparison between algorithms



Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

1. **Centroid-based / Partition-based clustering (e.g., K-means):** algorithms that divide the data into non-hierarchical clusters by defining “cluster centers”.

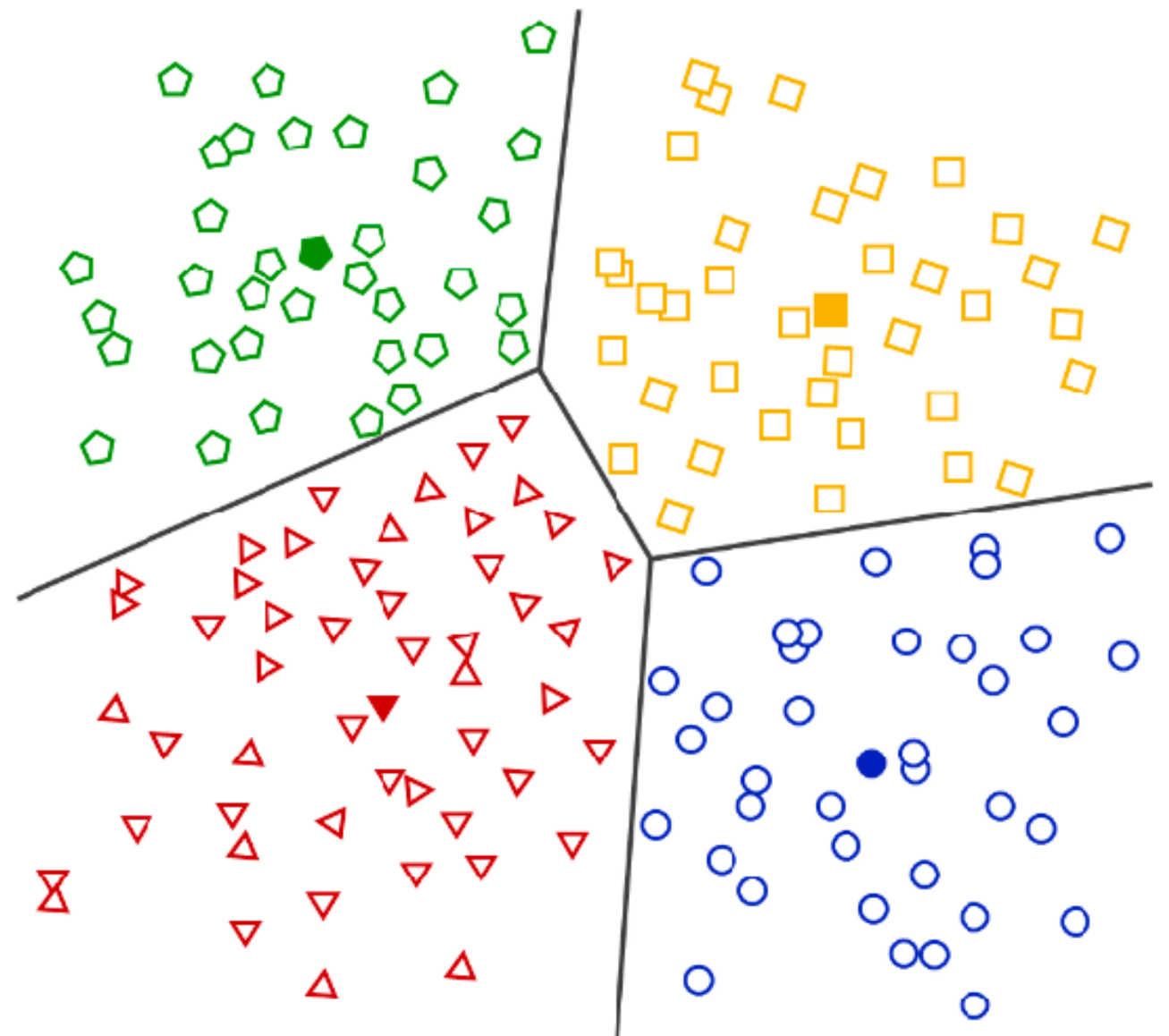


Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

1. Centroid-based / Partition-based clustering (e.g., K-means): algorithms that divide the data into non-hierarchical clusters by defining “cluster centers”.

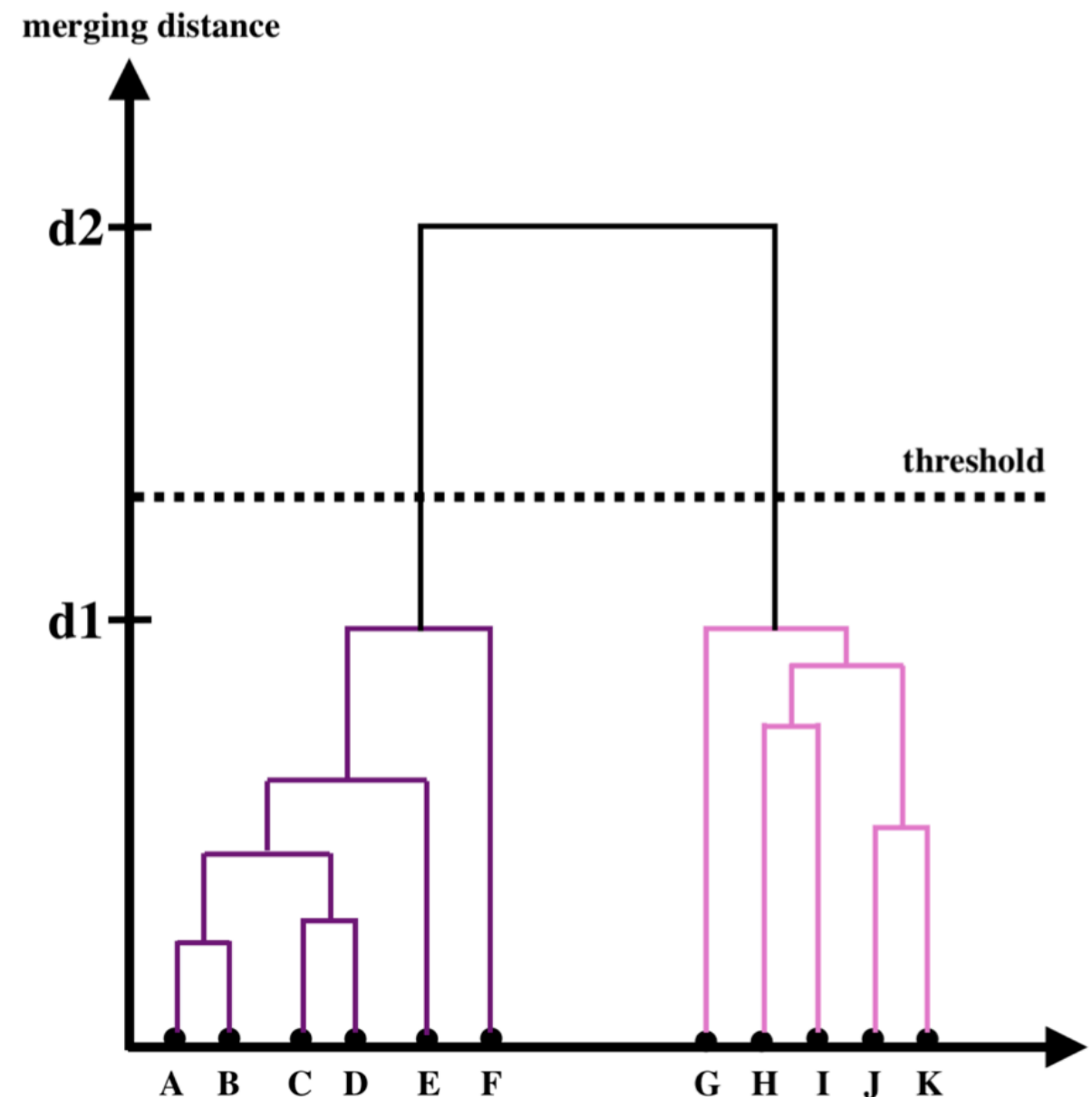
- Scales well with number of **samples** and number of **features**.
- Sensitive to initial conditions (may get stuck in a local minimum) and outliers.
- Even cluster size, flat geometry.



Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

- 2. Hierarchical clustering:** algorithms that create tree of clusters by merging close clusters into larger clusters.

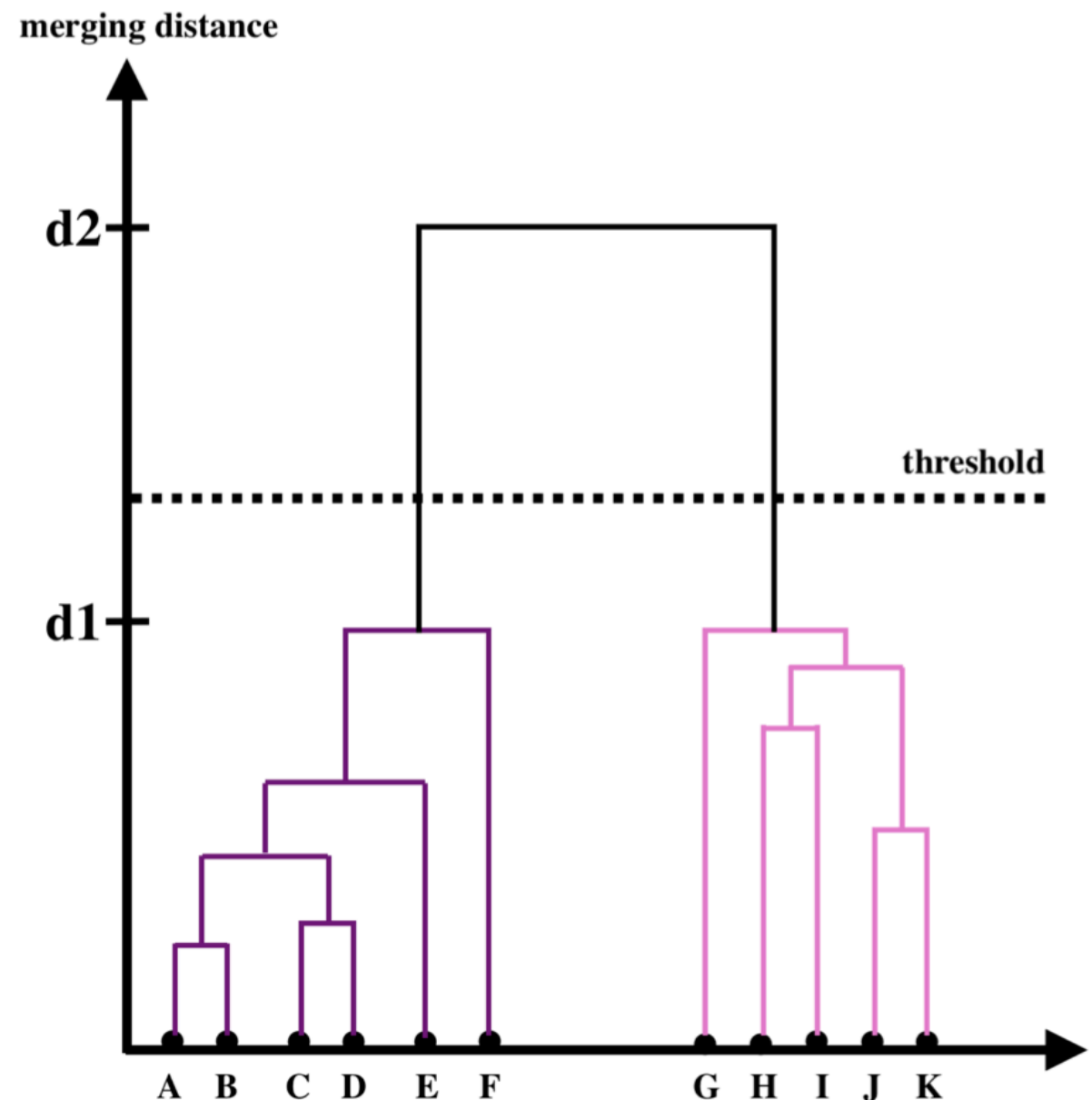


Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

2. Hierarchical clustering: algorithms that create tree of clusters by merging close clusters into larger clusters.

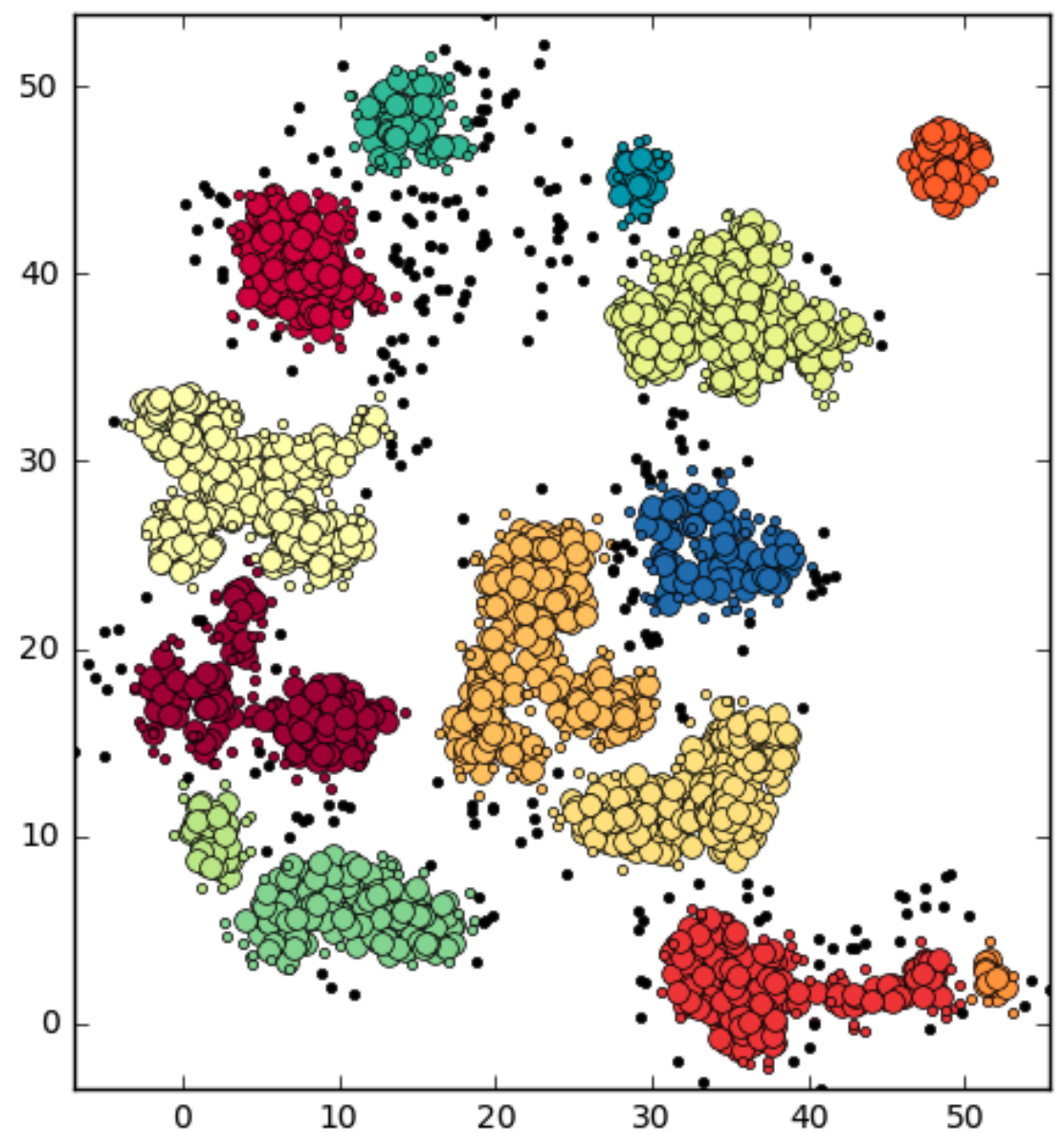
- Typical examples: BIRCH, CURE, ROCK, and Chameleon.
- Scales well with number of **samples** and number of **features**.
- Can work with many clusters, non-even cluster sizes.



Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

3. **Density-based clustering:** high density regions in the data space are considered to belong to the same cluster.



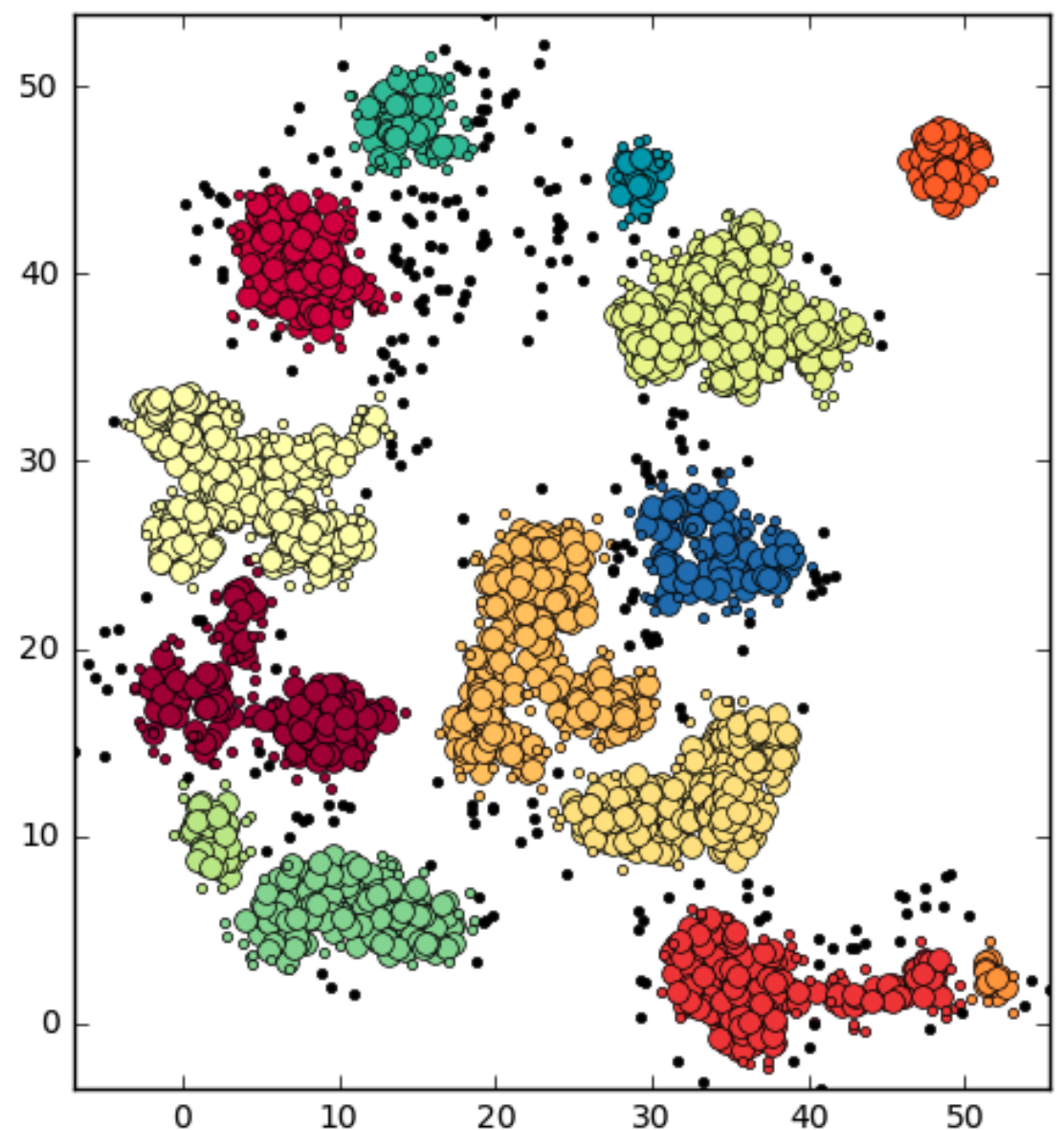
taken from [here](#).

Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

3. **Density-based clustering:** high density regions in the data space are considered to belong to the same cluster.

- Typical examples: DBSCAN, OPTICS, and Mean-shift.
- Scales well with number of **samples**. Not so well with number of **features**.
- Uneven cluster sizes, non-flat geometries. **Marks outliers**.
- Struggles with data with varying density.



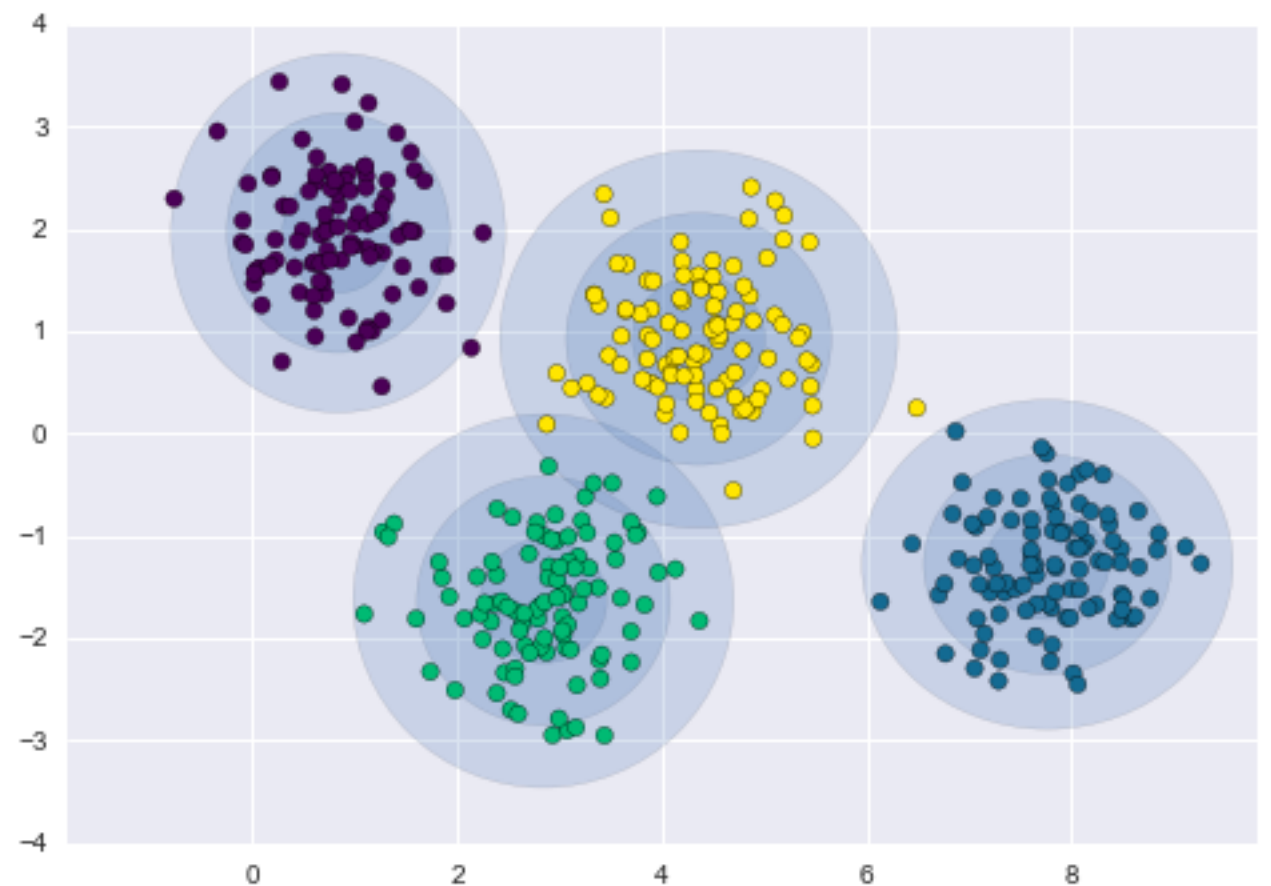
taken from [here](#).

Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

- 4. Distribution-based clustering:** algorithm assumes that data is composed of distributions. Same cluster's data points need to belong to the same probability distribution.

Credit: Jake VanderPlas.
Tutorial available [here](#).



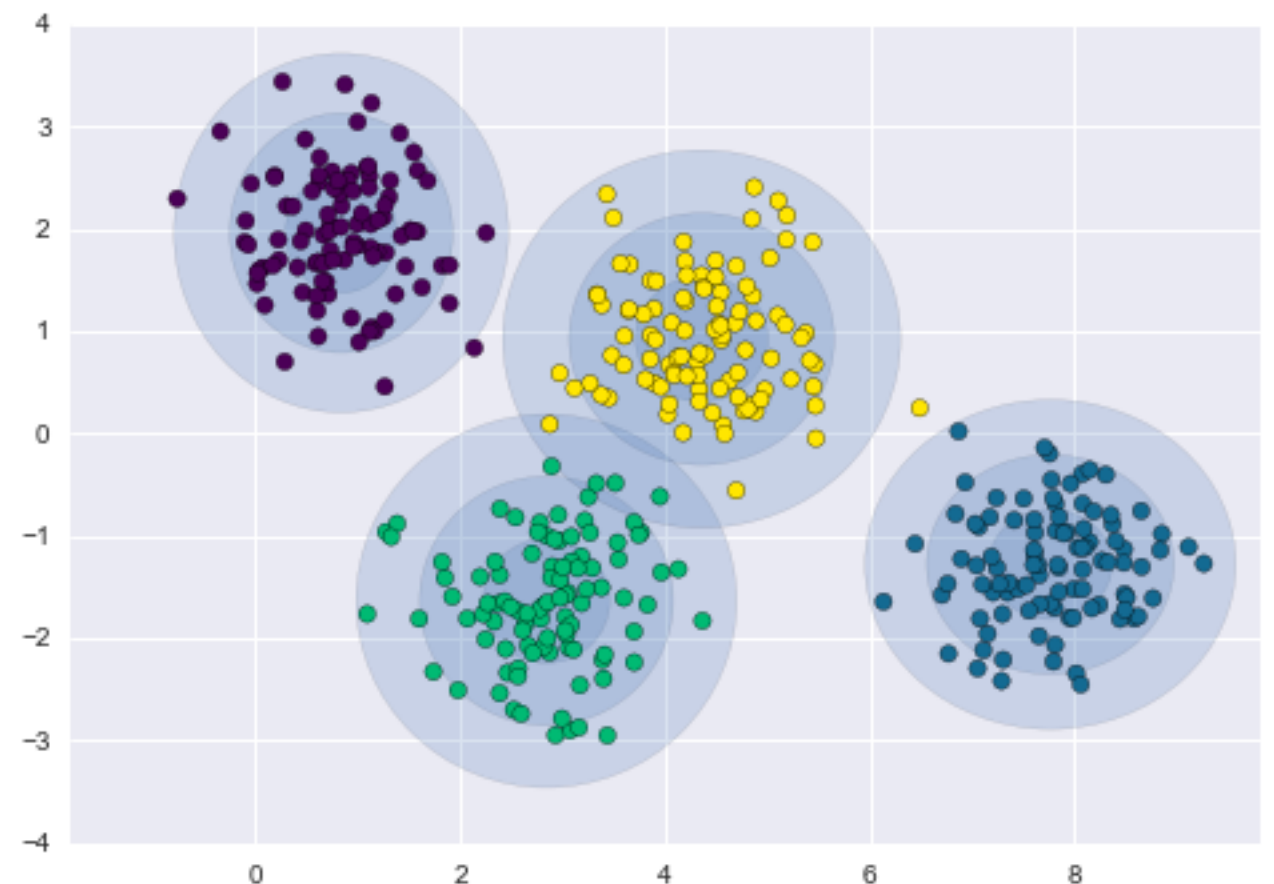
Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

4. **Distribution-based clustering:** algorithm assumes that data is composed of distributions. Same cluster's data points need to belong to the same probability distribution.

- Typical examples: DBCLASD and GMM. **Need to assume the distribution.**
- Does not scale well with the number of samples or features.
- Flat geometry.
- Assigns probabilities for every point.

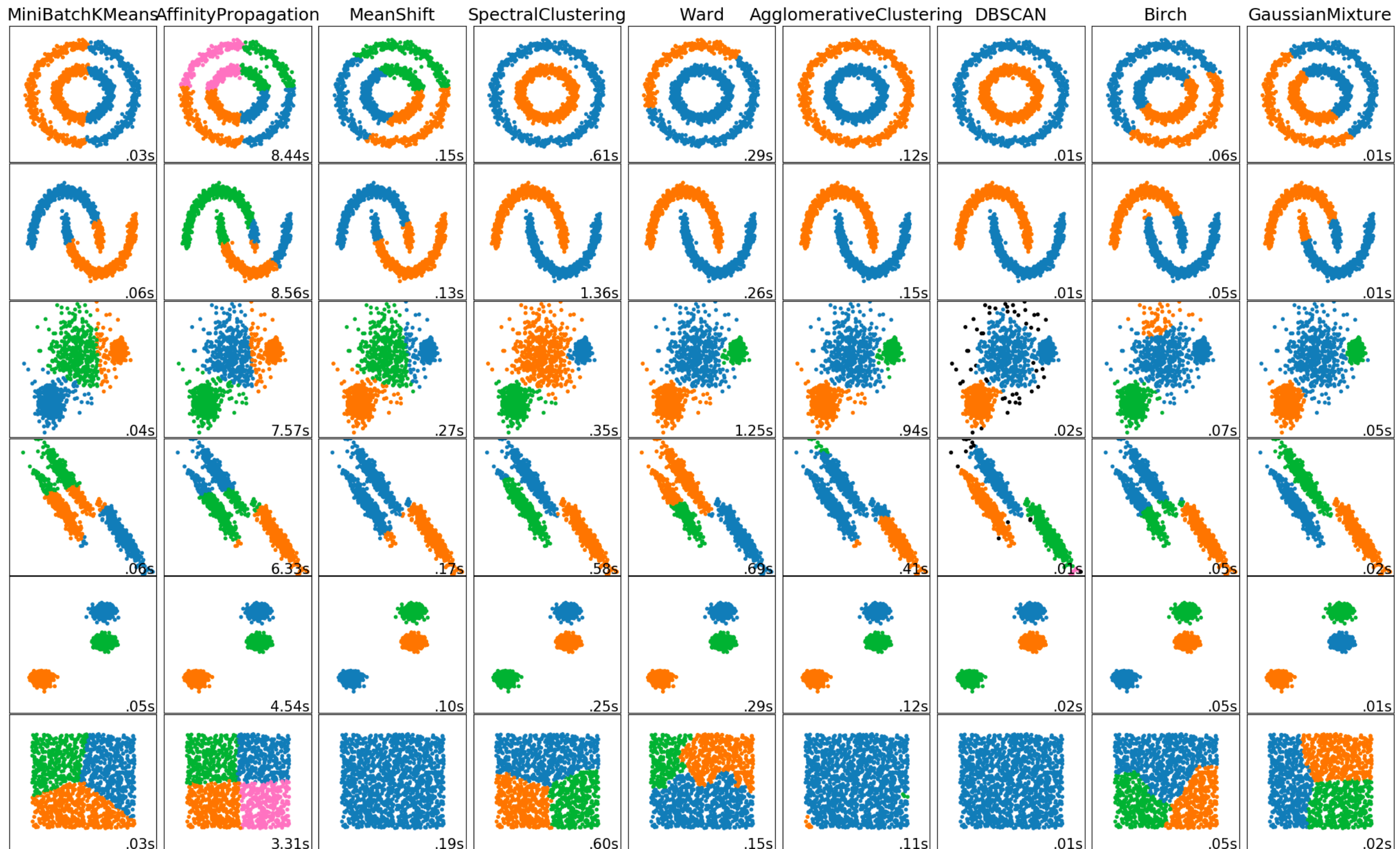
Credit: Jake VanderPlas.
Tutorial available [here](#).



How should I choose which algorithm to use?

From Scikit-learn's example gallery:

see [this](#) comparison between algorithms



Outlier detection algorithms

What is an outlier?

- ❖ **Bad object:** reduction problems, cosmic rays, legos floating around the earth...
- ❖ **Misclassified object:** objects that were incorrectly selected into our dataset. For example: a star in a sample of quasars, variable star accidentally classified as a transient, etc.
- ❖ **Tails of distributions:** objects of the same class that show extreme values in one of their properties.
- ❖ **Unknown unknowns:** objects we did not know we should be looking for, and might represent something new and exciting.

What is an outlier?

- ❖ **Bad object:** reduction problems, cosmic rays, legos floating around the earth...
- ❖ **Misclassified object:** objects that were incorrectly selected into our dataset. For example: a star in a sample of quasars, variable star accidentally classified as a transient, etc.
- ❖ **Tails of distributions:** objects of the same class that show extreme values in one of their properties.
- ❖ **Unknown unknowns:** objects we did not know we should be looking for, and might represent something new and exciting.

In astronomy, processes that take place on a shorter timescale will appear rare in our datasets.

How can we detect outliers?

1. Measure pair-wise distances between all the objects in the sample and identify objects with large distances.

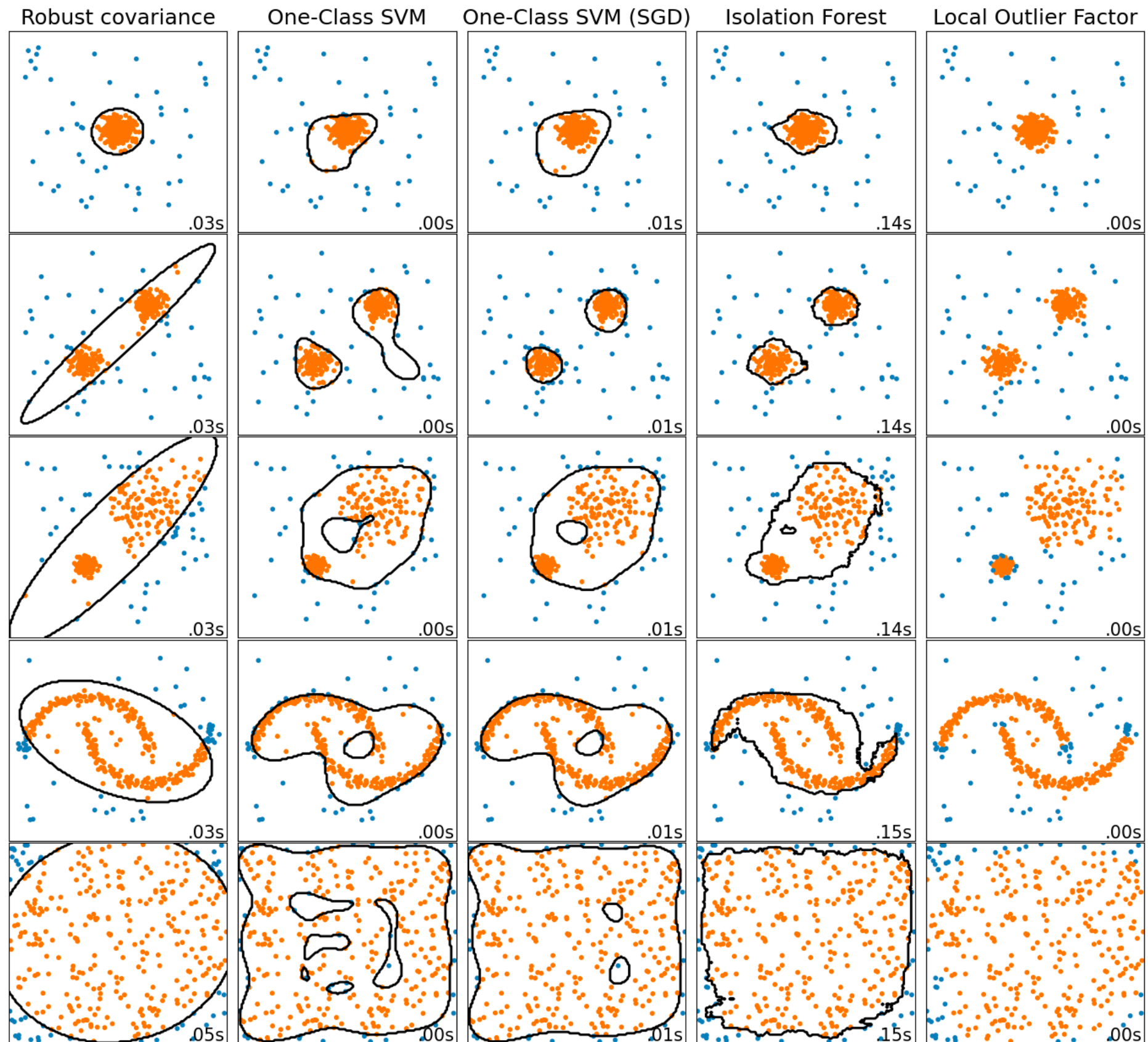
How can we detect outliers?

1. Measure pair-wise distances between all the objects in the sample and identify objects with large distances.
2. Using Supervised Learning algorithms:
 - In the framework of a classification task, objects that have a relatively-low probability to belong to a class will be considered outliers (e.g., Random Forest).

How can we detect outliers?

1. Measure pair-wise distances between all the objects in the sample and identify objects with large distances.
2. Using Supervised Learning algorithms:
 - In the framework of a classification task, objects that have a relatively-low probability to belong to a class will be considered outliers (e.g., Random Forest).
3. Using Unsupervised Learning algorithms:
 - Some clustering algorithms (e.g., Hierarchical clustering, DBSCAN, OPTICS, GMMs) flag outliers.
 - Apply a dimensionality reduction algorithm and identify outliers in the low-dimensional representation.

Outlier Detection Algorithms



Taken from [scikit-learn](https://scikit-learn.org/).

Interpreting the output on unsupervised learning algorithms

Dimensionality Reduction:

- Color by metadata (=derived features) to reveal structure.
- Repeat runs to check stability.
- Compare to original features.

Clustering:

- Validate with known labels or metrics.
- Inspect cluster centers / examples.
- Correlate clusters with physical properties.

Outlier Detection:

- Examine outliers individually.
- Check consistency across methods.
- Use domain knowledge to filter artifacts.

Good Practices

- ❖ Start simple:

- ❖ Simulate simple low-dimensional dataset, without noise, where the output can be anticipated.
- ❖ Compare the output of the algorithm for different data representations and different choices of hyper-parameters.

- ❖ Gradually complicate the model:

- ❖ Add more dimensions (some of them should be uninformative).
- ❖ Add noise.
- ❖ Compare the output for different representations and hyper-parameters.

- ❖ Physically-motivated model:

- ❖ Simulate a physically-motivated dataset.
- ❖ Experiment with different noise properties, different representations, and hyper-parameters.

- ❖ Try to break the algorithm!

Hands-on application

- ❖ Hands-on practice on data preparation and unsupervised learning algorithms:
 - ❖ Jupyter notebooks with examples and exercises will be provided.
 - ❖ Slides that describe the algorithms in detail will be provided.
 - ❖ No external data is necessary, but you can test on your own data!
- ❖ Advanced: Hands-on exercise on interacting with a frontier AI model on applying an unsupervised ML algorithm to some data.
 - ❖ Data: PHANGS multi-wavelength data and derived features will be provided, but you are welcome to use your own data!