Input Data and Distance Measures

Dalya Baron
Carnegie Observatories

Vatican Observatory Summer School on Big Data and Machine Learning 2023 (VOSS-2023)

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.

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

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.

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:** when we derive features from the data, we will typically clean the data in the process so that the data quality will be better.

Raw data vs. derived features

- **Data quality:** when we derive features from the data, we will typically clean the data in the process so that the data quality will be better.

- **Simplicity:** the number of derived features will typically be lower than the number of dimensions in the original data. This means that we are trying to solve a simpler task and many algorithms do not scale well with the number of dimensions. It is always better to use knowledge we already have, rather than hoping it will come out naturally from the algorithm. While in some cases the algorithm will discover what we already know, it will often come at the expense of discovering something new.

Raw data vs. derived features

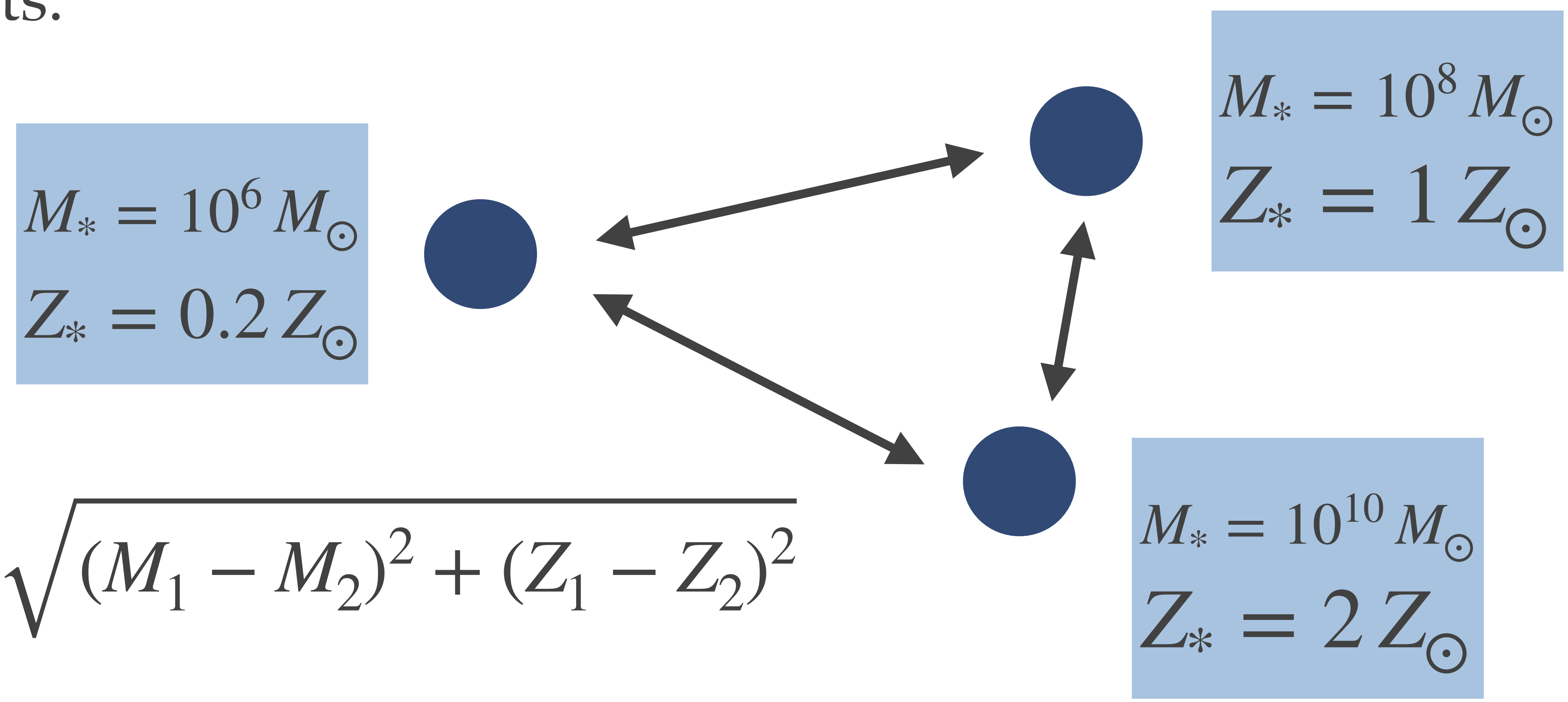
- **Data quality:** when we derive features from the data, we will typically clean the data in the process so that the data quality will be better.
- **Simplicity:** the number of derived features will typically be lower than the number of dimensions in the original data. This means that we are trying to solve a simpler task and many algorithms do not scale well with the number of dimensions. It is always better to use knowledge we already have, rather than hoping it will come out naturally from the algorithm. While in some cases the algorithm will discover what we already know, it will often come at the expense of discovering something new.
- **Interpretability:** it is easier to understand what is the meaning of the algorithm's output using derived features.

Raw data vs. derived features

- **Data quality:** when we derive features from the data, we will typically clean the data in the process so that the data quality will be better.
- **Simplicity:** the number of derived features will typically be lower than the number of dimensions in the original data. This means that we are trying to solve a simpler task and many algorithms do not scale well with the number of dimensions. It is always better to use knowledge we already have, rather than hoping it will come out naturally from the algorithm. While in some cases the algorithm will discover what we already know, it will often come at the expense of discovering something new.
- **Interpretability:** it is easier to understand what is the meaning of the algorithm's output using derived features.
- **Potential for new discoveries:** by using derived features we are limiting ourselves to a fewer number of parameters which we are already familiar with, which reduces the discovery potential. We only derive features for what we already know, what about the things we don't know?

Derived features: aspects to consider

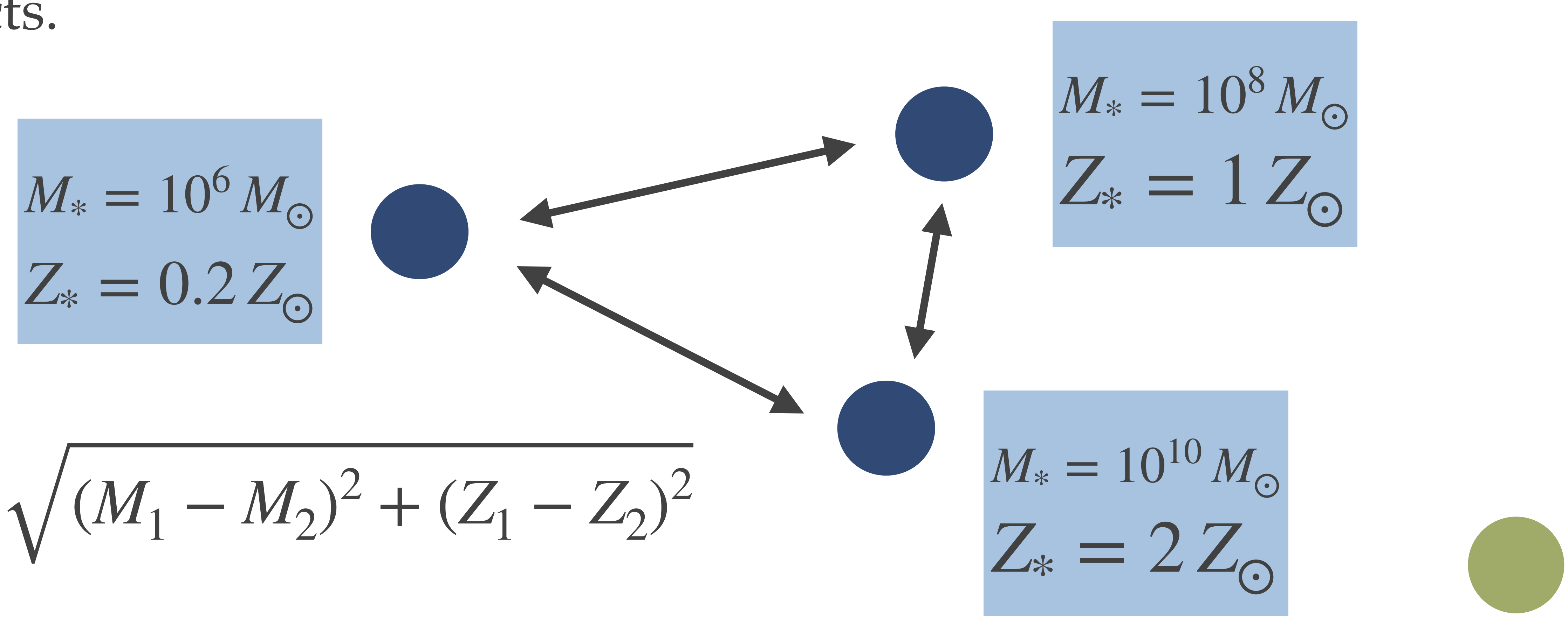
- **Feature scaling and normalization:** features derived from astronomical observations have physical units, and might have different dynamical scales. Features with larger variance will dominate the summed Euclidean distance between individual objects.



$$D_{12} = \sqrt{(M_1 - M_2)^2 + (Z_1 - Z_2)^2}$$

Derived features: aspects to consider

- **Feature scaling and normalization:** features derived from astronomical observations have physical units, and might have different dynamical scales. Features with larger variance will dominate the summed Euclidean distance between individual objects.



$$D_{12} = \sqrt{(M_1 - M_2)^2 + (Z_1 - Z_2)^2}$$

- **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: $f_{norm} = (f - \mu_f)/\sigma_f$.

Derived features: aspects to consider

- **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. A common practice is also to clip the feature values to be between the 0.5th and 99.5th percentiles of the distribution.

Derived features: aspects to consider

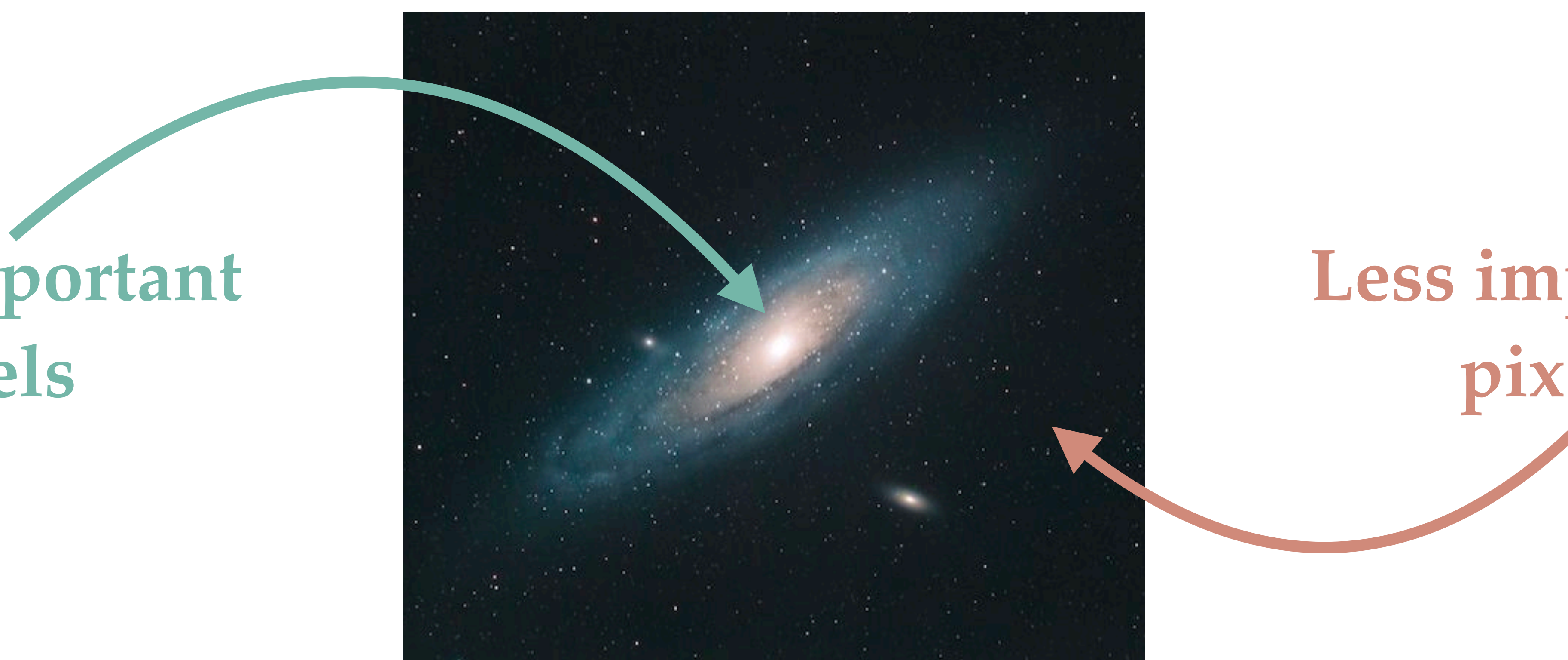
- **Correlated features:** a set of highly-correlated features in the dataset will result in a higher weight in the summed Euclidean distance, which may wash-out other structures in the dataset.
- **What to do?**
  - Use PCA to obtain features in an orthogonal space. Might not be very interpretable.
  - Instead of using the correlated features, use the deviation of one feature from the correlation with another. Might not scale well for multiple features with multiple correlations.

Raw data: aspects to consider

- Similarly to the derived features case, but depending on the data:
  - Feature scaling and normalization.
  - Outliers.
  - Correlated features.

Raw data: aspects to consider

- Similarly to the derived features case, but depending on the data:
  - Feature scaling and normalization.
  - Outliers.
  - Correlated features.
- Not all features are equally-important.

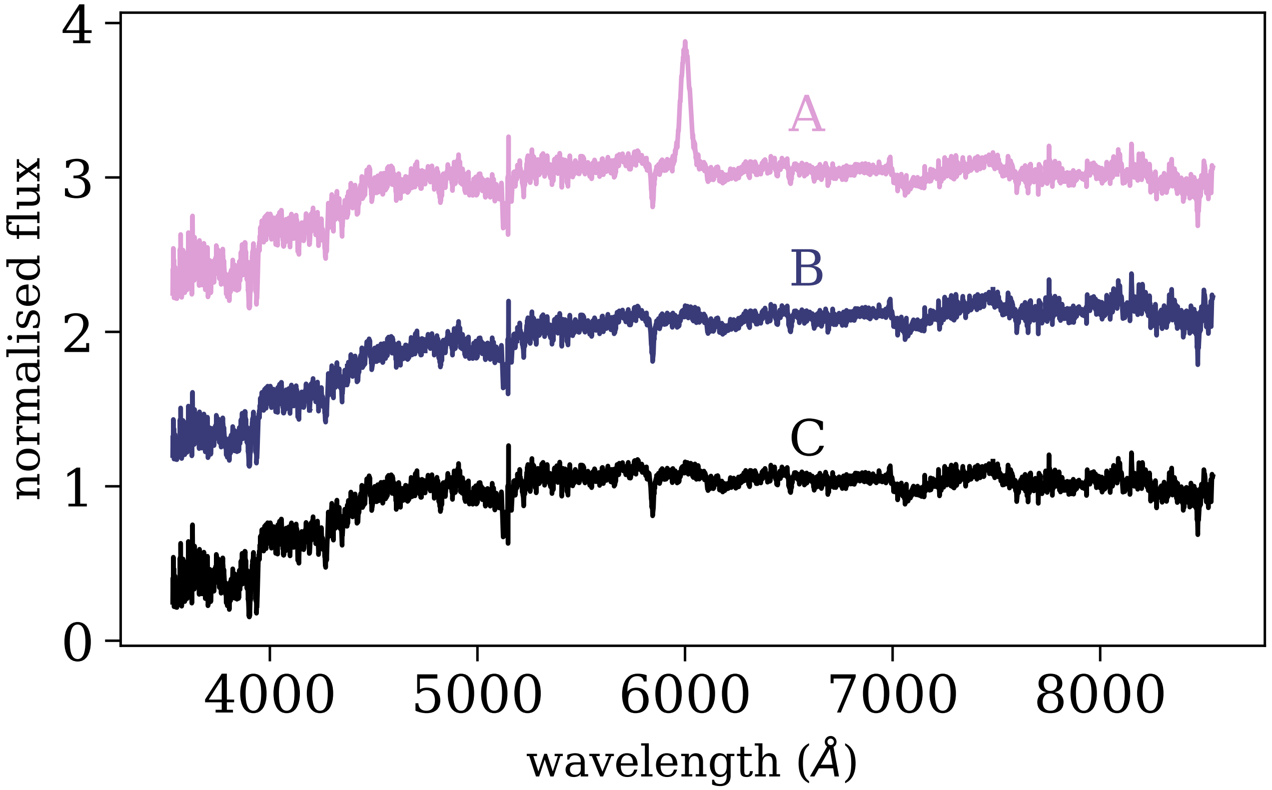


Raw data: aspects to consider

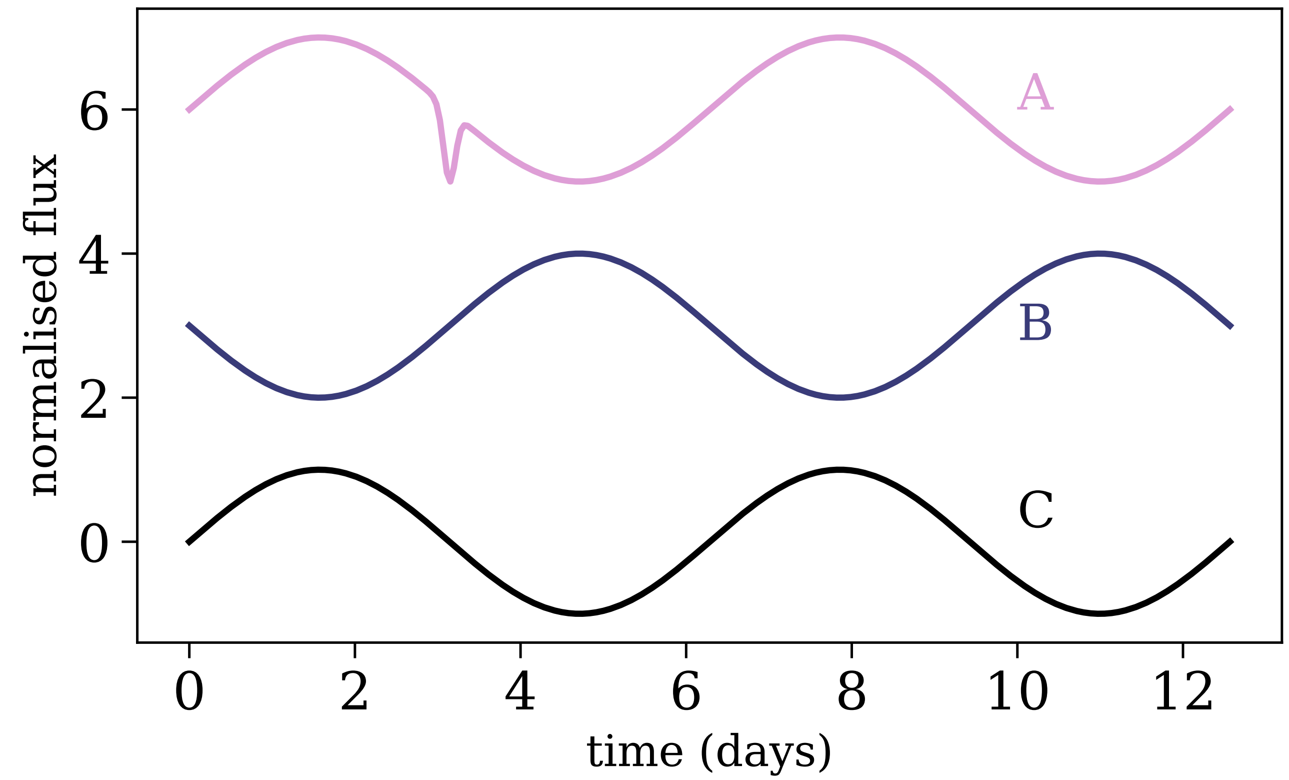
- Similarly to the derived features case, but depending on the data:
  - Feature scaling and normalization.
  - Outliers.
  - Correlated features.
- Not all features are equally-important.
- May be beneficial to transform the data into a different space. For example:
  - Frequency domain for time series data.
  - Wavelet transform for imaging data.
  - Representation using "eigenvectors" of spectral information.

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.

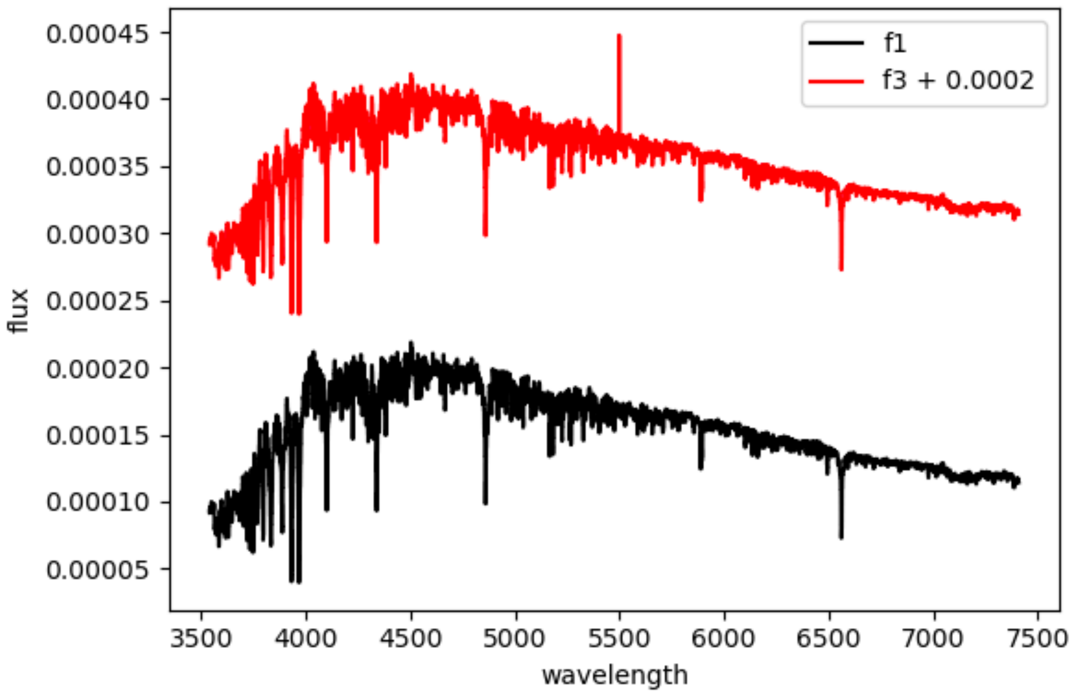
Distance measures

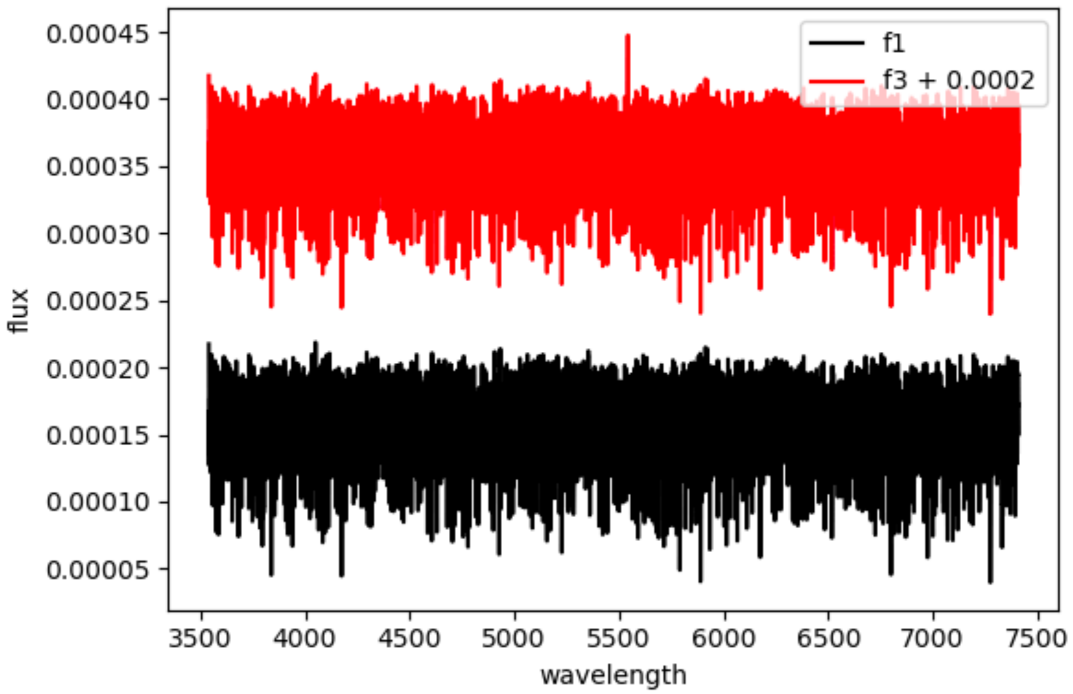
Regardless of whether we want to perform clustering, dimensionality reduction, or outlier detection, the large majority of algorithms start by estimating the pairwise distance between objects in the N-dimensional space.

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: $D_{ij}^2 = \sum_{f:features} (x_{if} - x_{jf})^2$
  - 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: $D_{ij}^2 = \sum_{f:features} (x_{if} - x_{jf})^2$
  - 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.