Data Analysis and Knowledge Discovery

Data Understanding II: Higher-dimensional Data

Jari Björne

University of Turku
Department of Computing
jari.bjorne@utu.fi

Lecture material: Antti Airola and Jari Björne

Outline

Section 1

Data Preparation

Data preparation

Data understanding provides general information about the data such as

- the existence (and partly the character) of missing values
- outliers
- the character of attributes
- dependencies between attributes

Data preparation

Data preparation uses this information to

- select attributes
- reduce the dimension of the data set
- select records
- treat missing values
- treat outliers
- integrate, unify and transform data
- improve data quality

Feature extraction

Feature extraction refers to the construction of (new) features from the given attributes.

Example. We are interested in finding the best workers in a company.

- Attributes like
  - the tasks a worker has finished within each month
  - the number of hours he has worked each month
  - the number of hours that are normally needed to finish each task
- In principle, these attributes contain information about the efficiency of the worker.
- But instead of using these three "raw" attributes, it might be more useful to define a new attribute *efficiency* which is the hours actually spent to finish the task divided by the hours normally needed to finish the tasks.

Feature extraction via nonlinear transformations

- Especially when using simple (e.g. linear) models, transforming attributes with non-linear functions can be helpful
- $x^p, 1/x, e^x, log(x), sin(x), cos(x), \frac{1}{1+e^x}, max(x, 0)...$
- $x_i * x_j, \frac{x_i}{x_j}, x_i^p x_j^q, max(x_i, x_j), x_i * x_j * x_k...$
- Let's assume we want to predict y from $x_1, x_2, ..., x_n$. How to find good transformations?
  - prior knowledge: we have some idea about how y could depend on x
  - visualization: for example scatter plot of y vs. x reveals some relationship
  - trial and error: generate different feature transformations and see how well they do
- Some methods automatically model or learn complex non-linear functions of input attributes (e.g. neural networks, kernel methods)

Feature extraction and dimensionality reduction

- Dimensionality reduction techniques like PCA can also be considered as feature extraction methods.
- But such automatic feature extraction methods usually lead to features that can no longer be interpreted in a meaningful way. $\rightarrow$ E.g. how to understand a feature which is a linear combination of 10 attributes?
- In most cases either knowledge-based, problem dependent feature extraction methods or feature selection techniques are preferred.

Feature extraction

Especially for complex data types feature extraction is required.
Example.

Text data analysis. Frequency of keywords ...

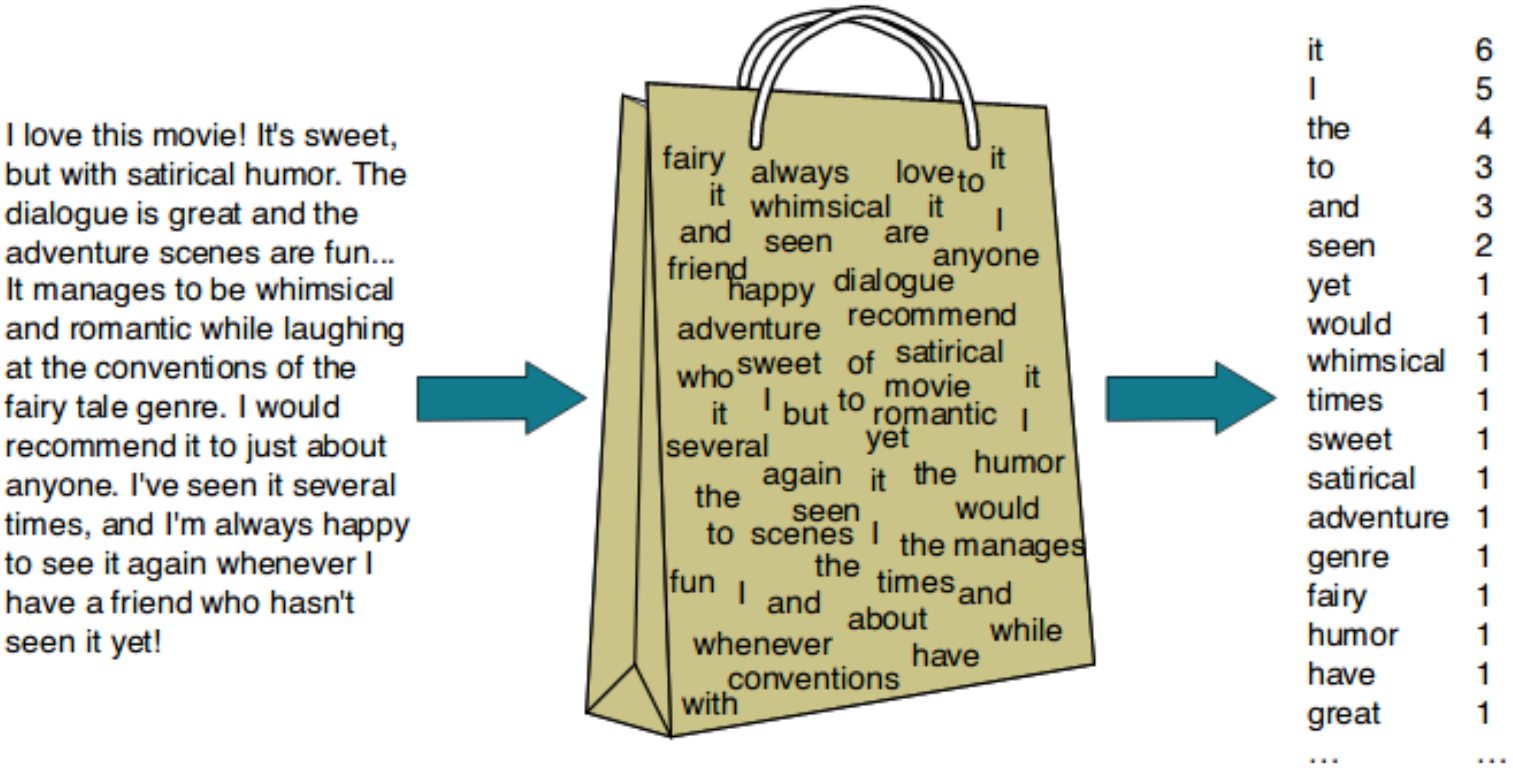
Time series data analysis. Fourier or wavelet coefficients ...

Image data analysis. Fourier or wavelet coefficients ...

Graph data analysis. number of vertices, number of edges ...

Feature extraction: Frequency of keywords

Bag of words: A set of common or task relevant words (e.g. $n = 500$) are the features (dimensions), and the number of occurrencies in one example (e.g. email classified as spam or not) is the value of the feature.



Source: https://dudeperf3ct.github.io/lstm/gru/nlp/2019/01/28/Force-of-LSTM-and-GRU/

Feature selection

Feature selection refers to techniques to choose a subset of the features (attributes) that is as small as possible and sufficient for the data analysis. Possible reasons for removing features

- Prior knowledge: we are certain that the feature is not relevant to the considered question
- Quality control: majority of values missing or of very bad quality
- Non-informative: e.g. a feature has the same value for all instances
- Redundancy: identical or close to perfectly correlated features (e.g. length in mm and cm)
- In supervised learning we may want to automatically search for a subset of features with best predictive power. This should be never done as a pre-processing step, but rather as part of the modeling phase (danger of overfitting, see upcoming lectures)

Record selection

Timeliness. Some of the older data might be outdated and might not be useful or even misleading for the data analysis task. Then only the recent data should be selected.

Representativeness. The sample in the database might not be representative for the whole population. When we have information about the distribution of the population, we can draw a representative subsample from our database.

Rare events. When we are interested in predicting rare events (e.g. stock market crashes, failures of a production line), it can be helpful

- to incorporate this in the cost function or
- to artificially increase the proportion of these rare events in the data set by adding copies of them or
- to choose only a subset of the other data.

Data cleansing

Data cleansing or data scrubbing refers to detecting and correcting or removing

- inaccurate,
- incorrect or
- incomplete

records from a data set.

Improve data quality

- Turn all characters into capital letters to level case sensitivity.
- Remove spaces and nonprinting characters.
- Fix the format of numbers, date and time (including decimal point).
- Split fields that carry mixed information into two separate attributes, e.g. *"Chocolate, 100g"* into *"Chocolate"* and *"100.0"*. This is known as field overloading.
- Use a spell-checker or stemming to normalize spelling in free text entries.
- Replace abbreviations by their long form (with the help of a dictionary).

Improve data quality

- Normalize the writing of adresses and names, possibly ignoring the order of title, surname, forename, etc. to ease their re-identification
- Convert numerical values into standard units, especially if data from different sources (and different countries) are used.
- Use dictionaries containing all possible values of an attribute, if available, to assure that all values comply with the domain knowledge.

Missing values

Ignorance/Deletion. If only a few records have missing values and it can be assumed that the values are missing completely at random (MCAR) (observed at random (OAR)), these records can be deleted for the following data analysis steps.

Imputation. The missing values may be replaced by some estimate.

- The mean, the median or the mode of the attribute. (MCAR/OAR required!)
- By an estimation based on the other attributes. (MAR required!)

Explicit value. Missing values are characterized by a specific value, say `MISSING` or `?`. The chosen model in the modelling steps must be able to handle missing values. (Most models assume MCAR/OAR!)

Discretization

Discretization techniques refer to splitting a numerical range into a number of finite bins.

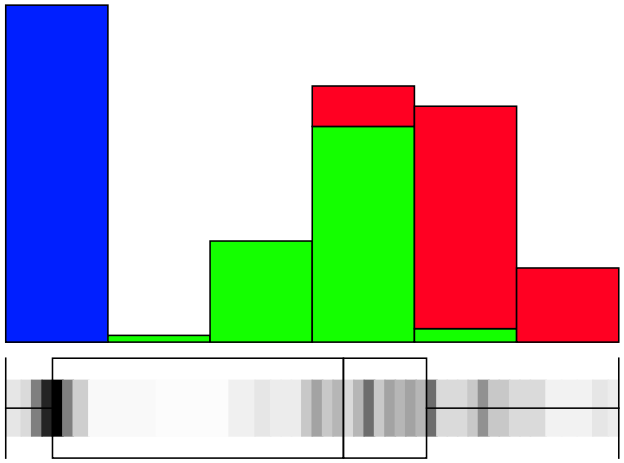
Equi-width discretization. Splits the range into intervals (bins) of the same length.

Equi-frequency discretization. Splits the range into intervals such that each interval (bin) contains (roughly) the same number of records.

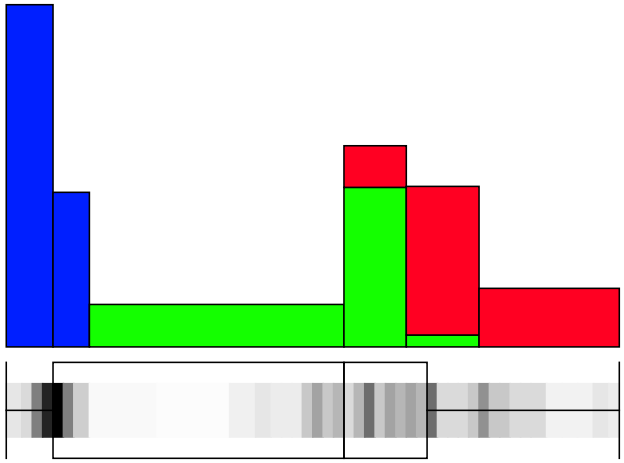
V-optimal discretization. Minimises $\sum_i n_i V_i$ where n_i is the number of data objects in the ith interval and V_i is the sample variance of the data in this interval.

Minimal entropy discretization. Minimises the entropy. (Only applicable in the case of classification problems.)

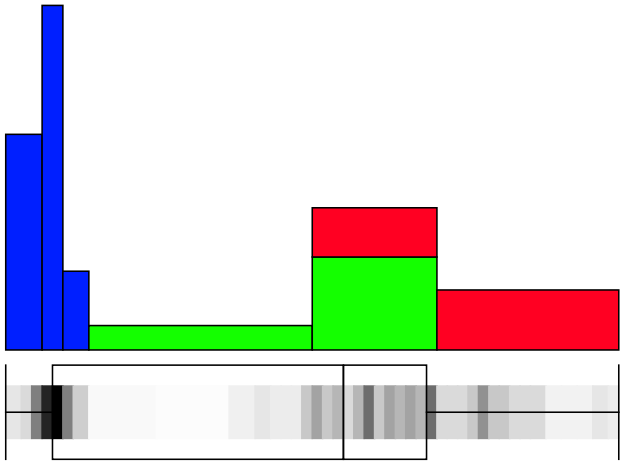
Discretization



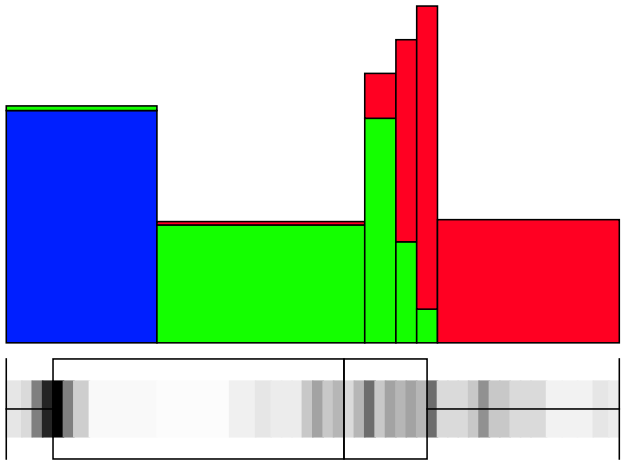
Equi-width



Equi-frequency



V-optimal



Minimal entropy

Data integration

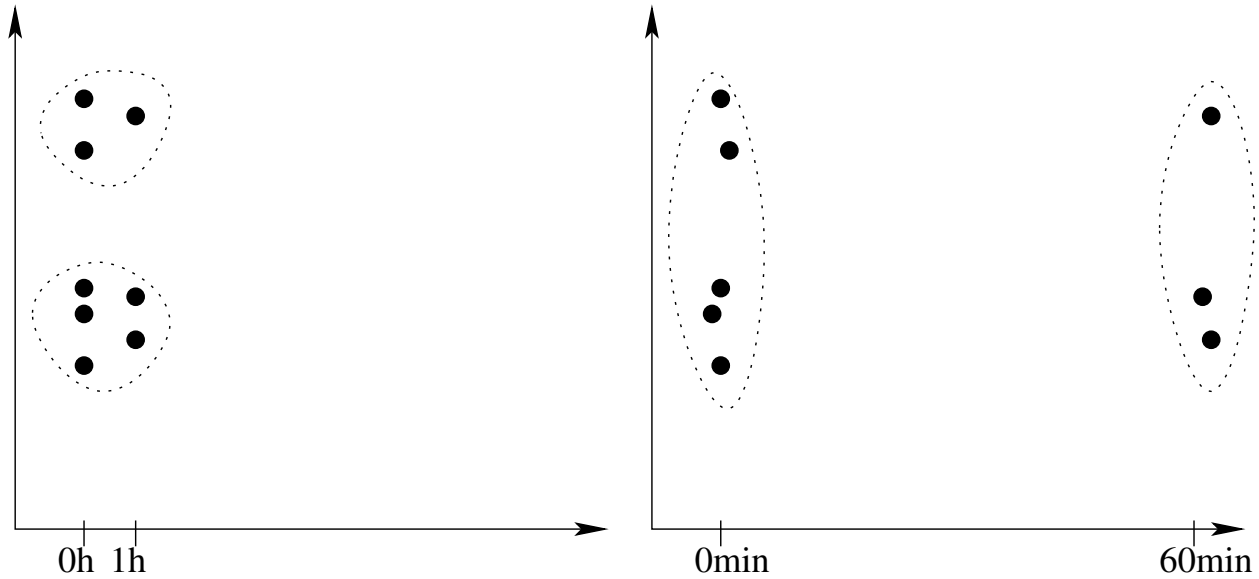
- Often need to merge together data from different data sources
  - Customer data bases of two compaines merged
  - Different hospitals provide data for clinical study
  - Combining satellite images, mineral deposit maps, geochemical information etc. for geoinformatics analysis
- unifying structure: "firstname, lastname", "lastname, firstname" "firstname", "lastname"...
- missing values: certain attribute recorded in one data source but not another
- duplicates: same instance appears in multiple data sources
- joins: linking entries in one data base to another (e.g. customer x bought product y)
- overlaying maps recorded at different resolutions
- ...

Section 2

Normalisation/Standardisation

Normalisation/Standardisation

For some data analysis techniques (e.g. PCA, t-SNE, nearest neighbours, cluster analysis) the influence of an attribute depends on the scale or measurement unit.



To guarantee impartiality, some kind of standardisation or normalisation should be applied.

Normalisation/Standardisation

min-max normalization. For a numerical attribute X with $\min_X$ and $\max_X$ being the minimum and maximum value in the sample, the min-max normalization is defined as

$$n : \mathrm{dom}X \to [0, 1], \qquad x \mapsto \frac{x - \min_X}{\max_X - \min_X}$$

z-score standardization. For a numerical attribute X with sample mean $\hat{\mu}_X$ and empirical standard deviation $\hat{\sigma}_X$, the z-score standardization is defined as

$$s : \mathrm{dom}X \to \mathbb{R}, \qquad x \mapsto \frac{x - \hat{\mu}_X}{\hat{\sigma}_X}$$

robust z-score standardization. The sample mean and empirical standard deviation are easily affected by outliers. A more robust alternative is (see also boxplots):

$$s : \mathrm{dom}X \to \mathbb{R}, \qquad x \mapsto \frac{x - \tilde{x}}{IQR_X}$$

decimal scaling. For a numerical attribute X and the smallest integer value s larger than $\log_{10}(\max_X)$, the decimal scaling is defined as

$$d : \mathrm{dom}X \to [0, 1], \qquad x \mapsto \frac{x}{10^s}$$

Normalizing/Standardizing the Data Matrix

A data matrix has m instances (rows) and n attributes (columns)

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_{1*} \\ \vdots \\ \boldsymbol{x}_{m*} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{*1}, ..., \boldsymbol{x}_{*n} \end{bmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}$$

.

- Elements: x_{ij} refers to the j:th attribute of i:th instance
- Row vectors: $\boldsymbol{x}_{i*}$ corresponds to i:th instance
- Column vectors: $\boldsymbol{x}_{*j}$ corresponds to j:th attribute
- Iris data: rows are individual flowers (instances) and columns sepal/petal length/width (attributes). Note that $\boldsymbol{X}$ does not contain the target (class) variable species.

Centering the Data Matrix

Let's normalize the data before going further

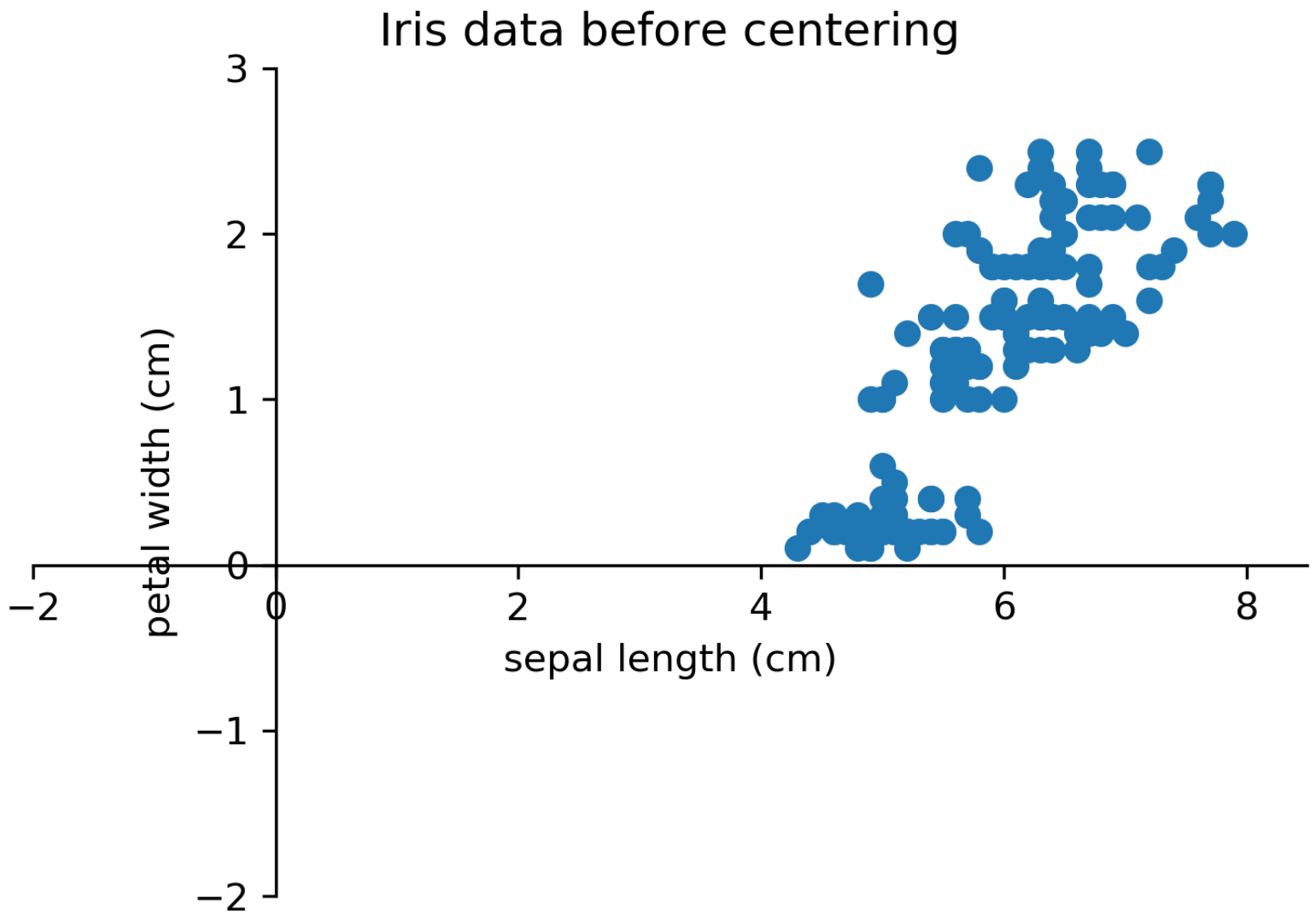
$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{x}_{1*} - \boldsymbol{\mu} \\ \vdots \\ \boldsymbol{x}_{m*} - \boldsymbol{\mu} \end{bmatrix}$$

Here, $\boldsymbol{\mu} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{x}_{*1}$ is the mean vector of all the rows of $\boldsymbol{X}$. Its

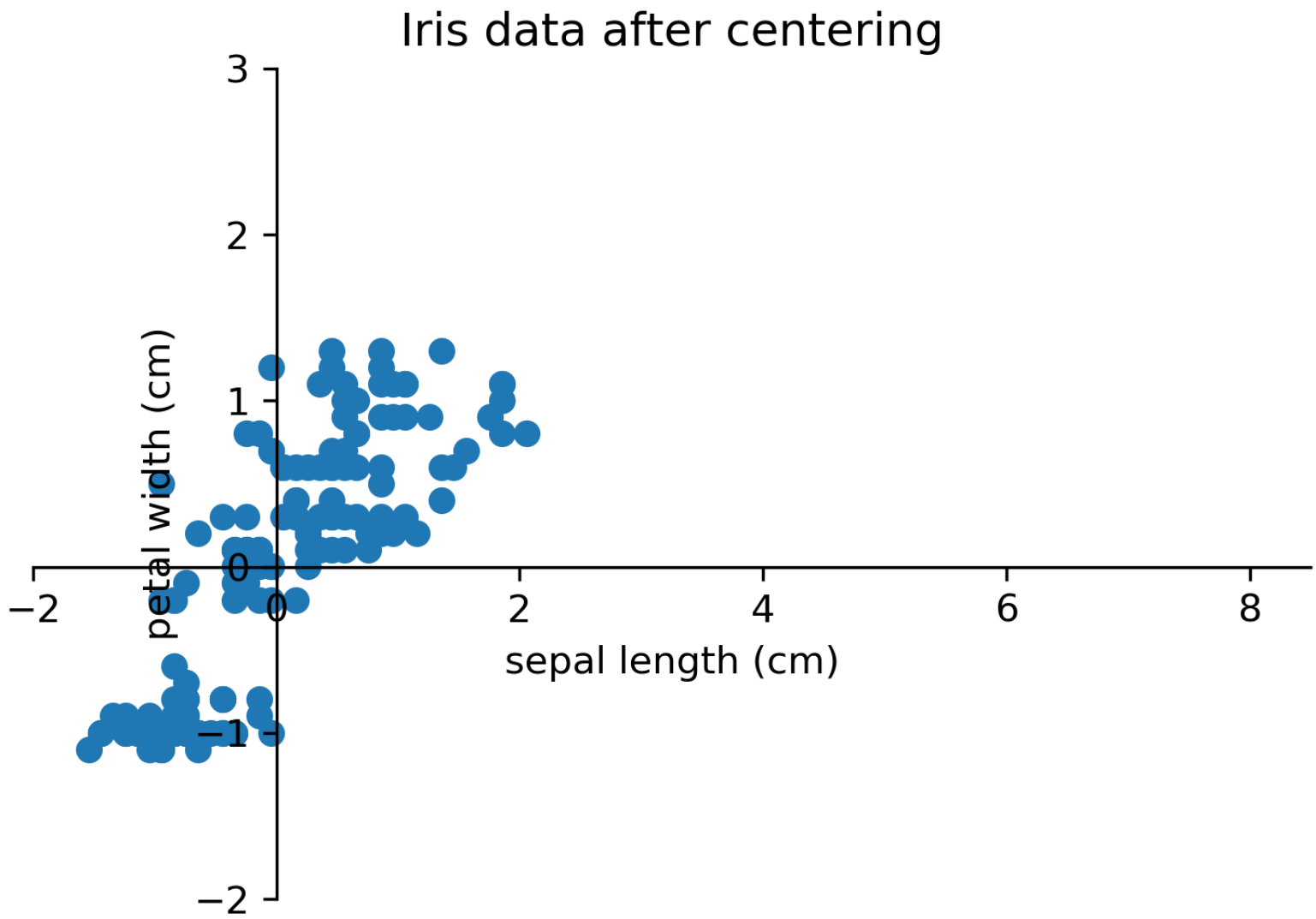
element μ_i contains the sample mean for the i:th attribute.

(Optional) we may also normalize the variance of each attributes to 1 (so-called z-score standardization) so that attributes with larger measurement scales / variance would not dominate subsequent analysis. In this case we set $z_{ij} = \frac{x_{ij} - \mu_j}{s_j}$, where s_j is the standard deviation of j:th attribute.

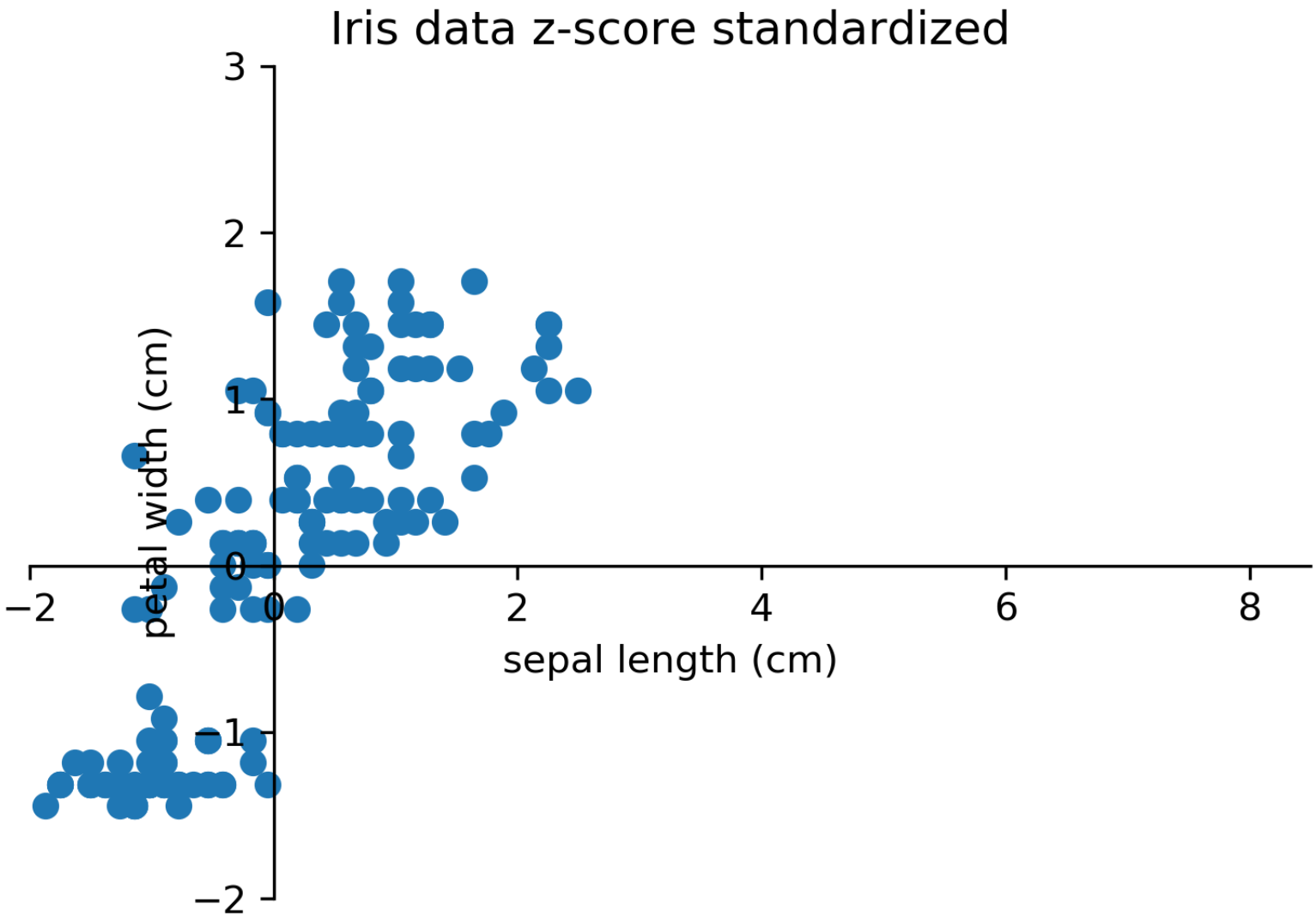
Centering and standardization



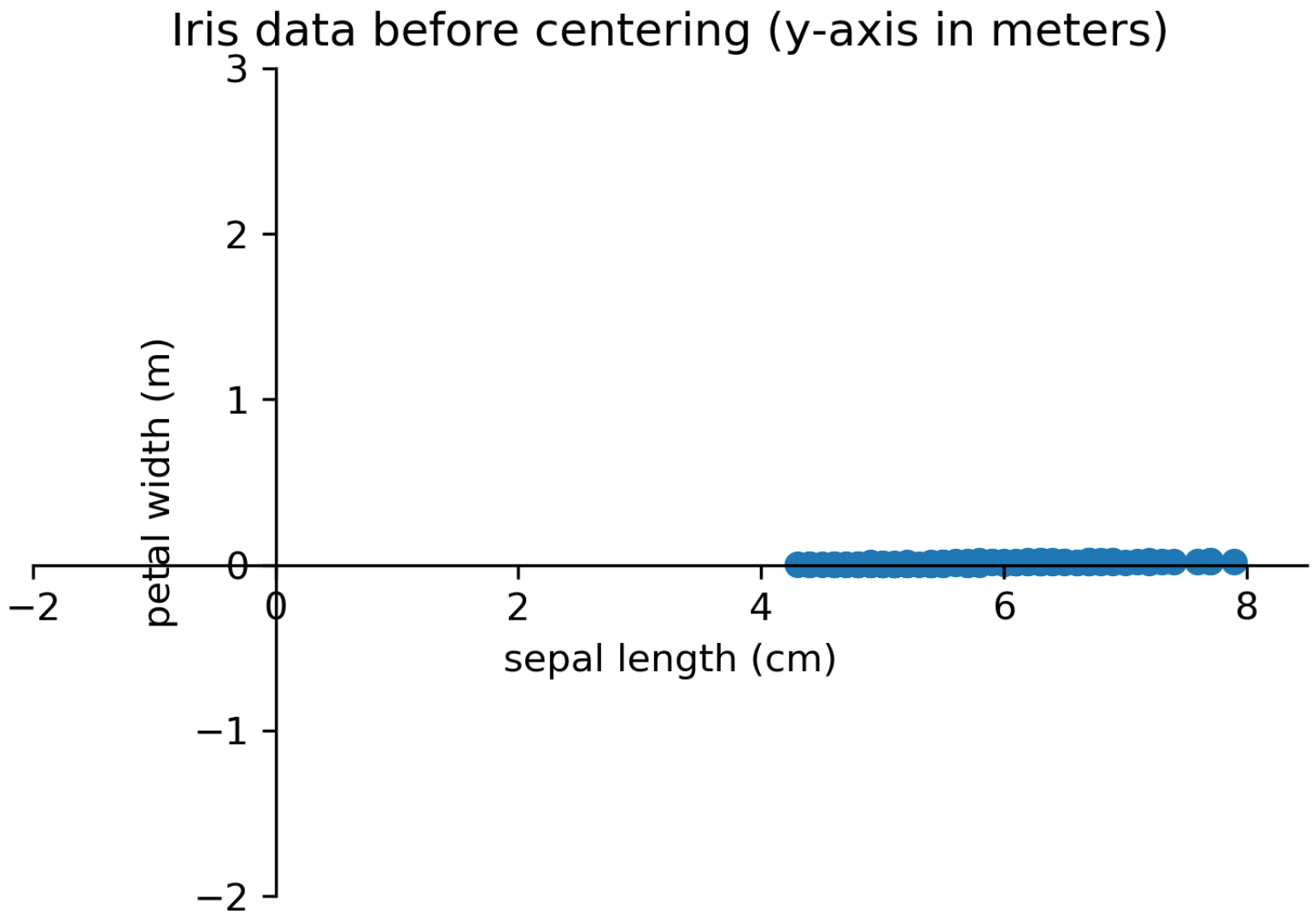
Centering and standardization



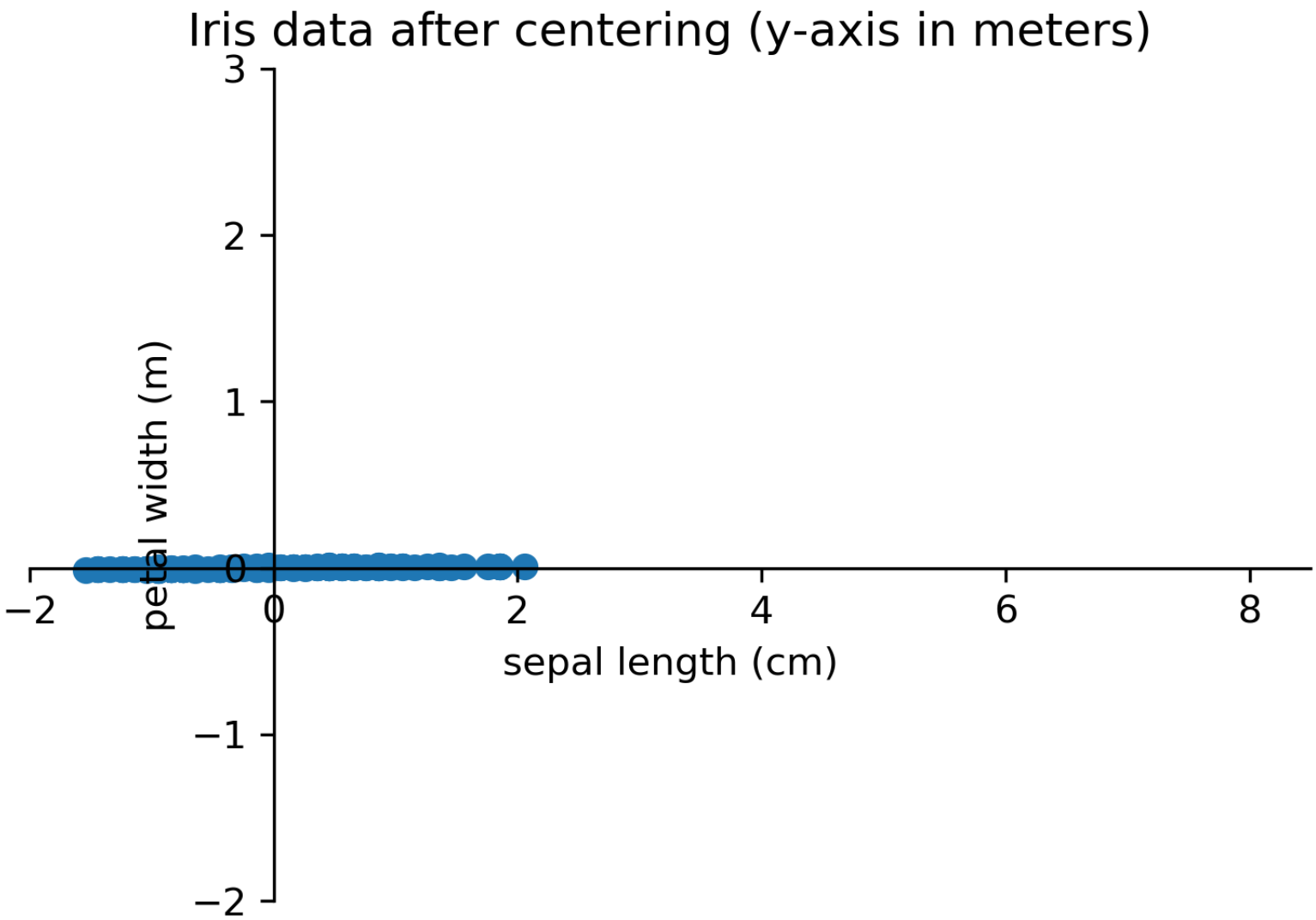
Centering and standardization



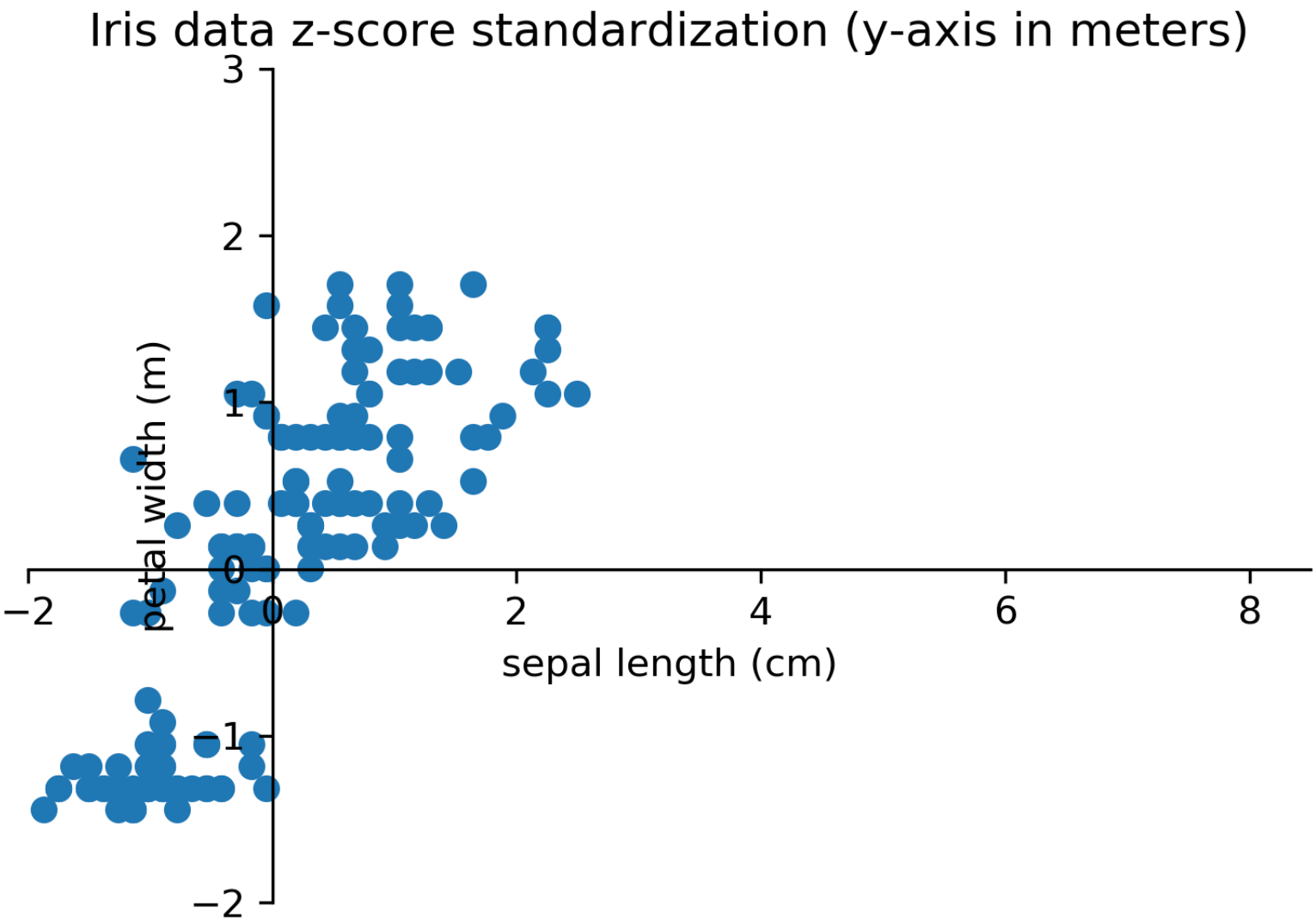
Centering and standardization



Centering and standardization



Centering and standardization



Section 3

Methods for higher-dimensional data

Methods for Higher-dimensional Data

- A display or plot is usually two-dimensional, so only two axes (attributes) can be incorporated.
- 3D techniques can be used to incorporate three axes (attributes).
- The number of possible 2D scatter plots grows in a quadratic fashion with the number of attributes. For m attributes there are $\binom{m}{2} = m(m-1)$ possible scatter plots. For 50 attributes there are 2450 scatter plots.

Example Data Set

```
Sepal.Length Sepal.Width Petal.Length Petal.Width Species

5.1   3.5   1.4   0.2   Iris-setosa
...
...
5.0   3.3   1.4   0.2   Iris-setosa
7.0   3.2   4.7   1.4   Iris-versicolor
...
...
5.1   2.5   3.0   1.1   Iris-versicolor
5.7   2.8   4.1   1.3   Iris-versicolor
...
...
5.9   3.0   5.1   1.8   Iris-virginica
```

In some datasets the number of attributes can be up to hundreds of thousands or more (e.g. length of genome, number of words in a language, polynomial interaction attributes).

Methods for Higher-dimensional Data

A principled approach for incorporating all the attributes in a plot:

- Try to preserve as much of the "structure" of the high-dimensional data set as possible when representing (plotting) the data in two (or three) dimensions.
- Define a measure that evaluates lower-dimensional representations (plots) of the data in terms of how well a representation preserves the original "structure" of the high-dimensional data set.
- Find the representation (plot) that gives the best value for the defined measure.
- If the target variable (e.g. species) is also given, the reduction technique may take this into account

There is no unique measure for "structure" preservation.

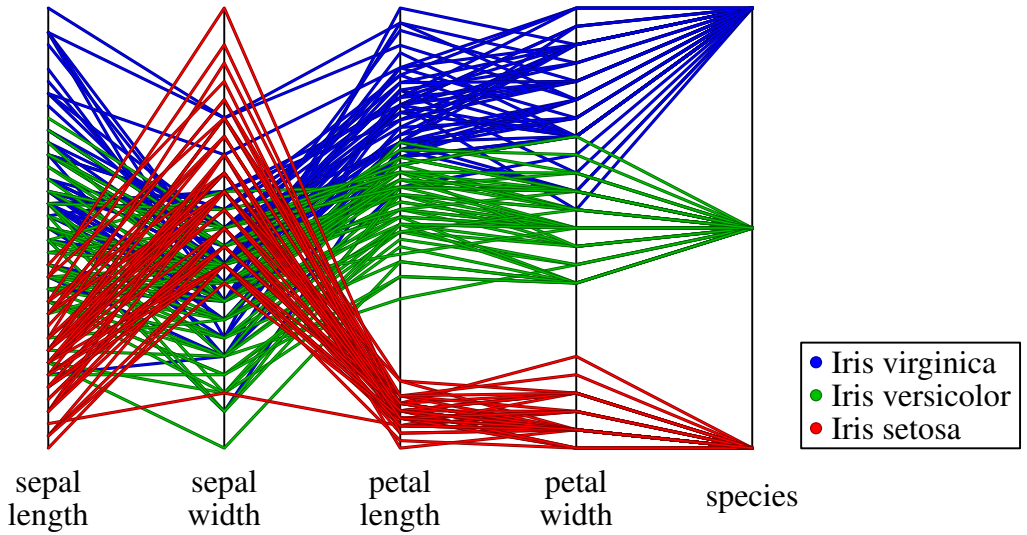
Section 4

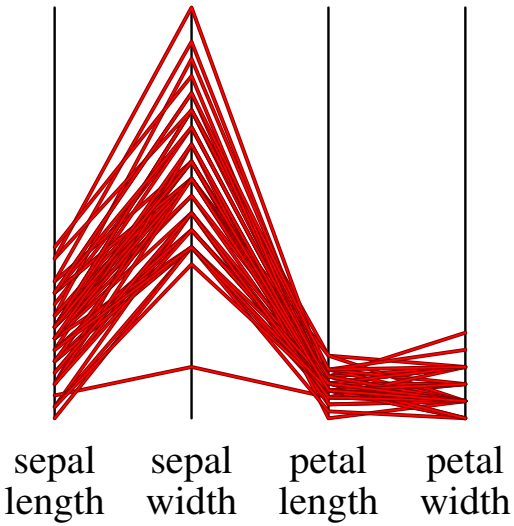
Parallel coordinates

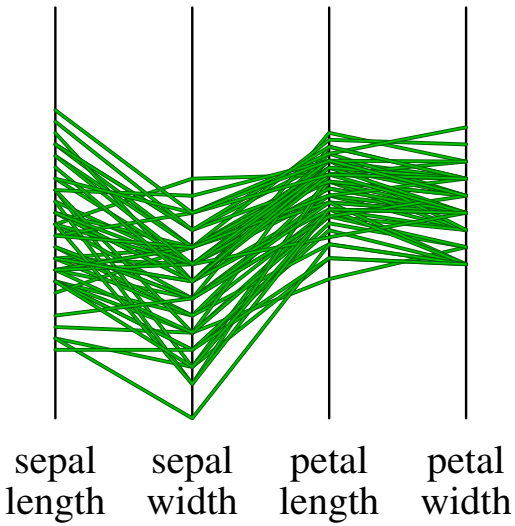
Parallel coordinates

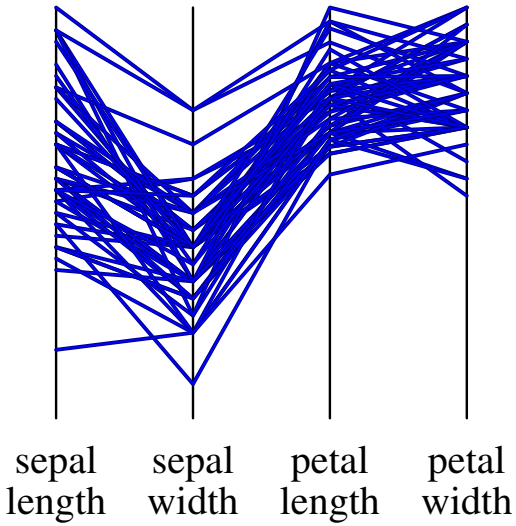
- Parallel coordinates draw the coordinate axes parallel to each other, so that there is no limitation for the the number of axes to be displayed.
- For a data object, a polyline is drawn connecting the values of the data object for the attributes on the corresponding axes.

Parallel coordinates: Iris data

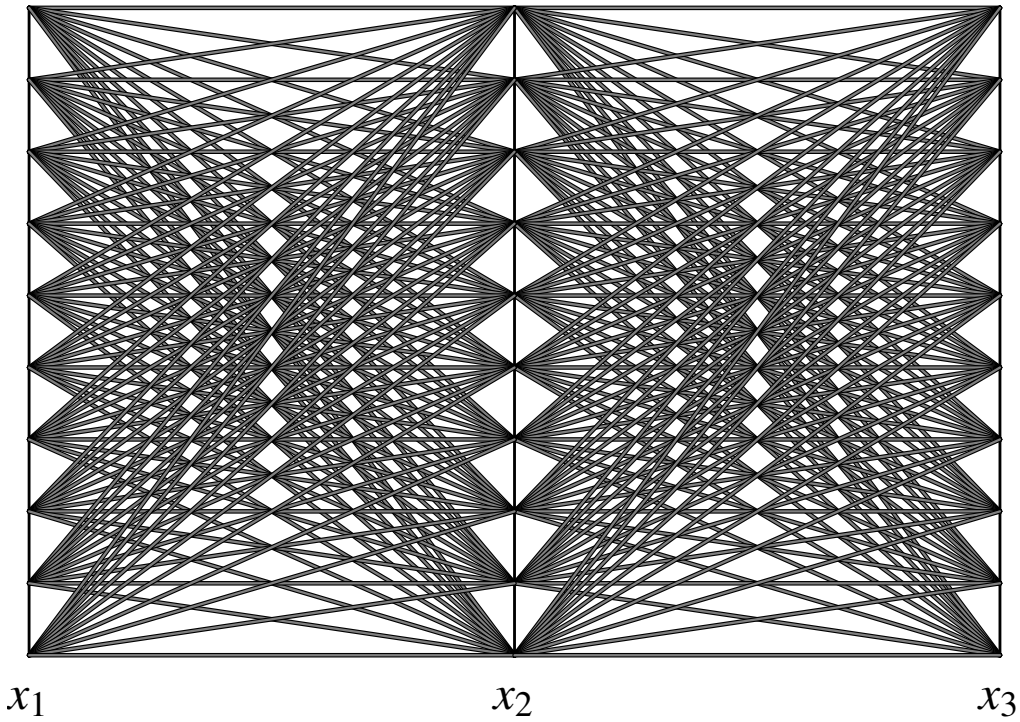








Parallel coordinates: "Cube data"



Section 5

Dimensionality reduction

Dimensionality reduction

- Basic idea: find a map $\mathbb{R}^n \rightarrow \mathbb{R}^q$ to transform our data from n-dimensional space to q-dimensional one
- $q << n$ (for visualization purposes, $q = 2$ or $q = 3$)
- mapping should in some meaningful way preserve the structure of the data
- we should be able to compute the mapping efficiently

Linear map

- Linear map: new attributes are linear combinations of old ones
- Let us invent a new combination attribute for Iris data!
  - $new\_feature = sep\_len + 2 * sep\_width + 0.5 * pet\_len - pet\_wid$
  - $\boldsymbol{m}$ = [1, 2, 0.5, -1] encodes this transformation
  - we can compute this feature for new instance by computing inner product $\langle \boldsymbol{m}, \boldsymbol{z} \rangle = \sum_{i=1}^{n} m_i z_i$
- Let's invent q new combination attributes!

Linear map

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{m}_{1*} \\ \vdots \\ \boldsymbol{m}_{q*} \end{bmatrix} = \begin{pmatrix} m_{11} & \dots & m_{1n} \\ m_{21} & \dots & m_{2n} \\ \vdots & \vdots & \vdots \\ m_{q1} & \dots & m_{qn} \end{pmatrix}$$

- $\boldsymbol{M}$ defines a linear map $\mathbb{R}^n \rightarrow \mathbb{R}^q$
- each row of $\boldsymbol{M}$ encodes coefficients for computing a new attribute, that is a linear combination of existing ones
- mapping by matrix-vector multiplication: $\boldsymbol{M}\boldsymbol{z}^T = [\langle \boldsymbol{m}_{1*}, \boldsymbol{z} \rangle, \langle \boldsymbol{m}_{2*}, \boldsymbol{z} \rangle, ..., \langle \boldsymbol{m}_{q*}, \boldsymbol{z} \rangle]^\top$
- mapping the whole (centered) data matrix at once: $\boldsymbol{M}\boldsymbol{Z}^T$

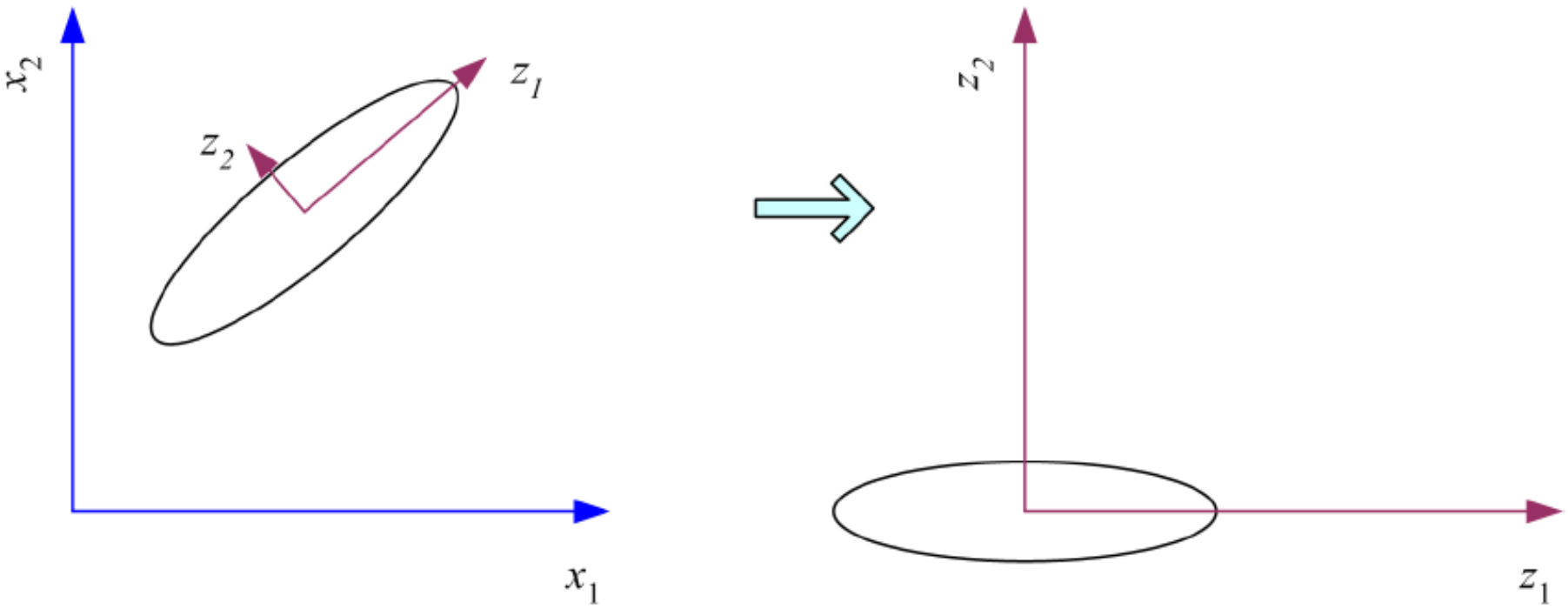
Linear map

- Given a $q \times n$ matrix $\boldsymbol{M}$, you can use it to map your data to new low-dimensional representation
- New attributes are linear combinations of the old ones
- How can we find such $\boldsymbol{M}$ that the mapping would in some meaningful way preserve the structure of the data?
- Answer: Principal Component Analysis (PCA)

Principal component analysis

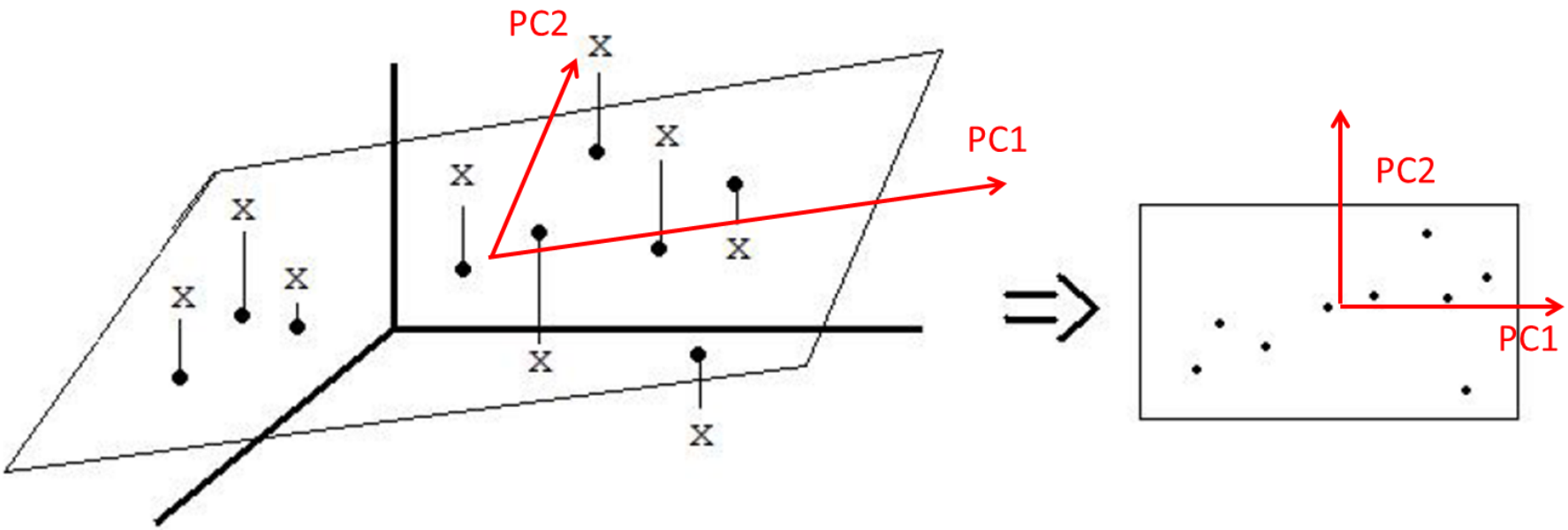
Principal component analysis (PCA) uses the variance in the data as the structure preservation criterion.

PCA tries to preserve as much of the original variance of the data when projected to a lower-dimensional space.



PCA example: 3D to 2D

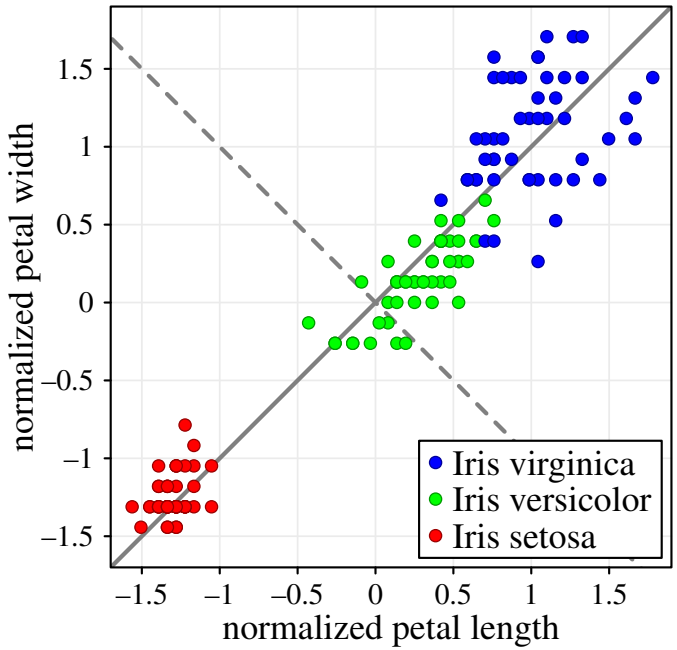
The 2D plane where the 3D points are projected is defined by 2 first principal components.



Principal component analysis

- PCA constructs a projection from the high-dimensional space to a lower-dimensional space (plane or hyperplane).
- Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.
- This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.
- The resulting vectors are an uncorrelated orthogonal basis set.
- PCA is sensitive to the relative scaling of the original variables.

Principal component analysis



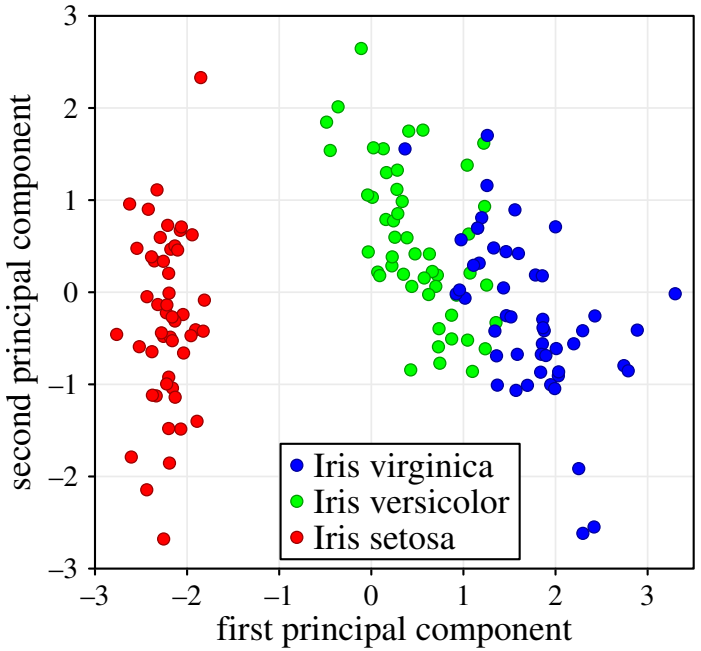
PCA applied to the Iris data set restricted to the (zscore normalised) petal length and width.

The principal components (1. solid line, 2. dashed line) are always orthogonal.

Principal component analysis: Normalisation

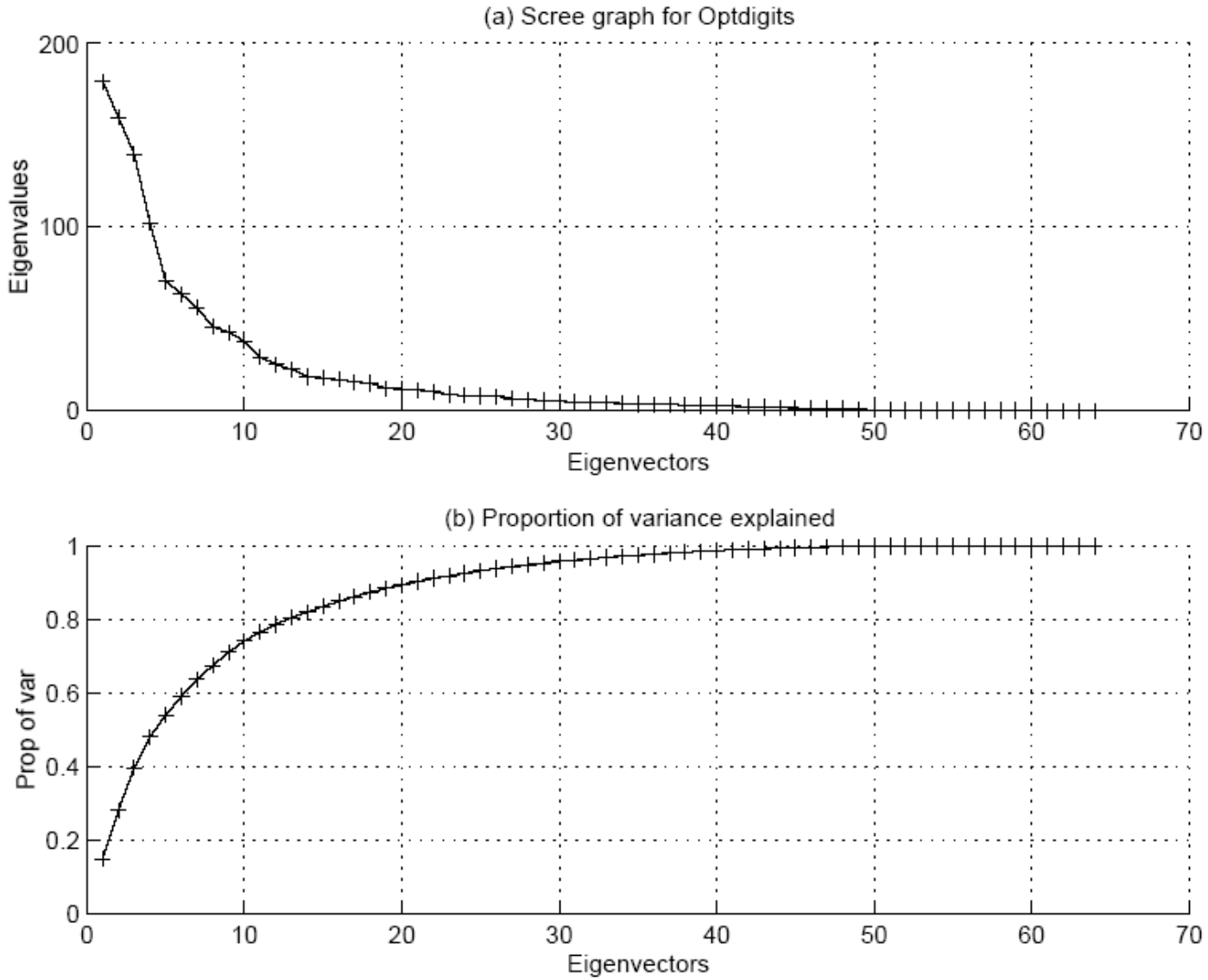
- Usually, the data should be z-score standardised $x \mapsto \frac{x-\hat{\mu}_X}{\hat{\sigma}_X}$ to ensure that all attributes contribute equally to the overall variance, where $\hat{\mu}_X$ and $\hat{\sigma}_X$ are the mean value and the sample standard deviation (the square root of the sample variance) of attribute X.
- When we change the measurement of the petal length from centimetres to metres, but leave the measurement of the petal width in centimetres, the first principal component becomes the vector $(0.0223, 0.9998)^\top$ without z-score standardisation.
- The variance of the petal length becomes negligible compared to the variance of the petal width.

Principal component analysis

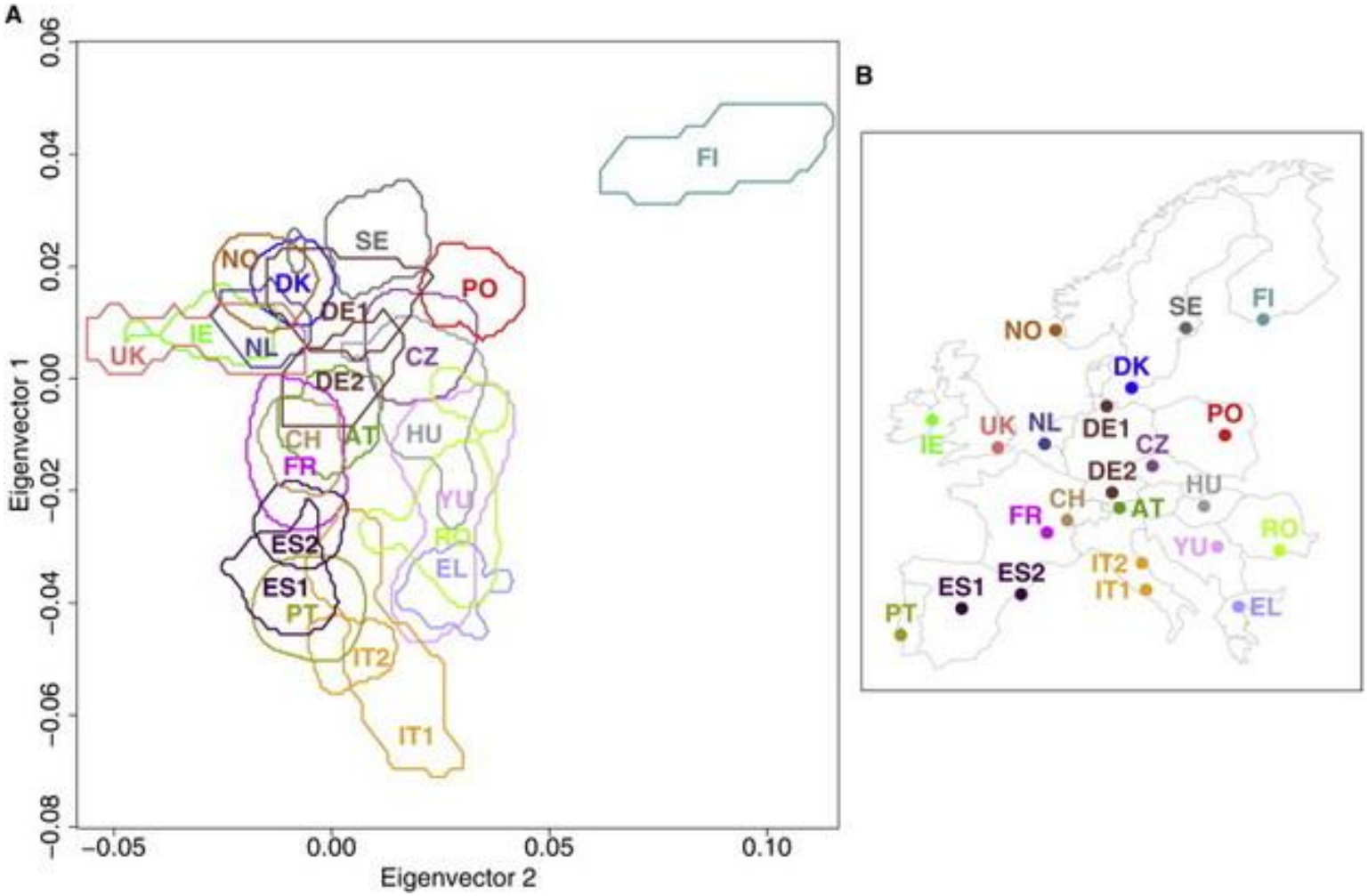


Projection to the first two principal components of PCA for the Iris data set taking all four numerical attributes into account.

Principal component analysis: Dimension reduction



Methods for higher-dimensional data



Lao, O., T. T. Lu, et al. (2008). Correlation between Genetic and Geographic Structure in Europe. Current Biology 18(16).

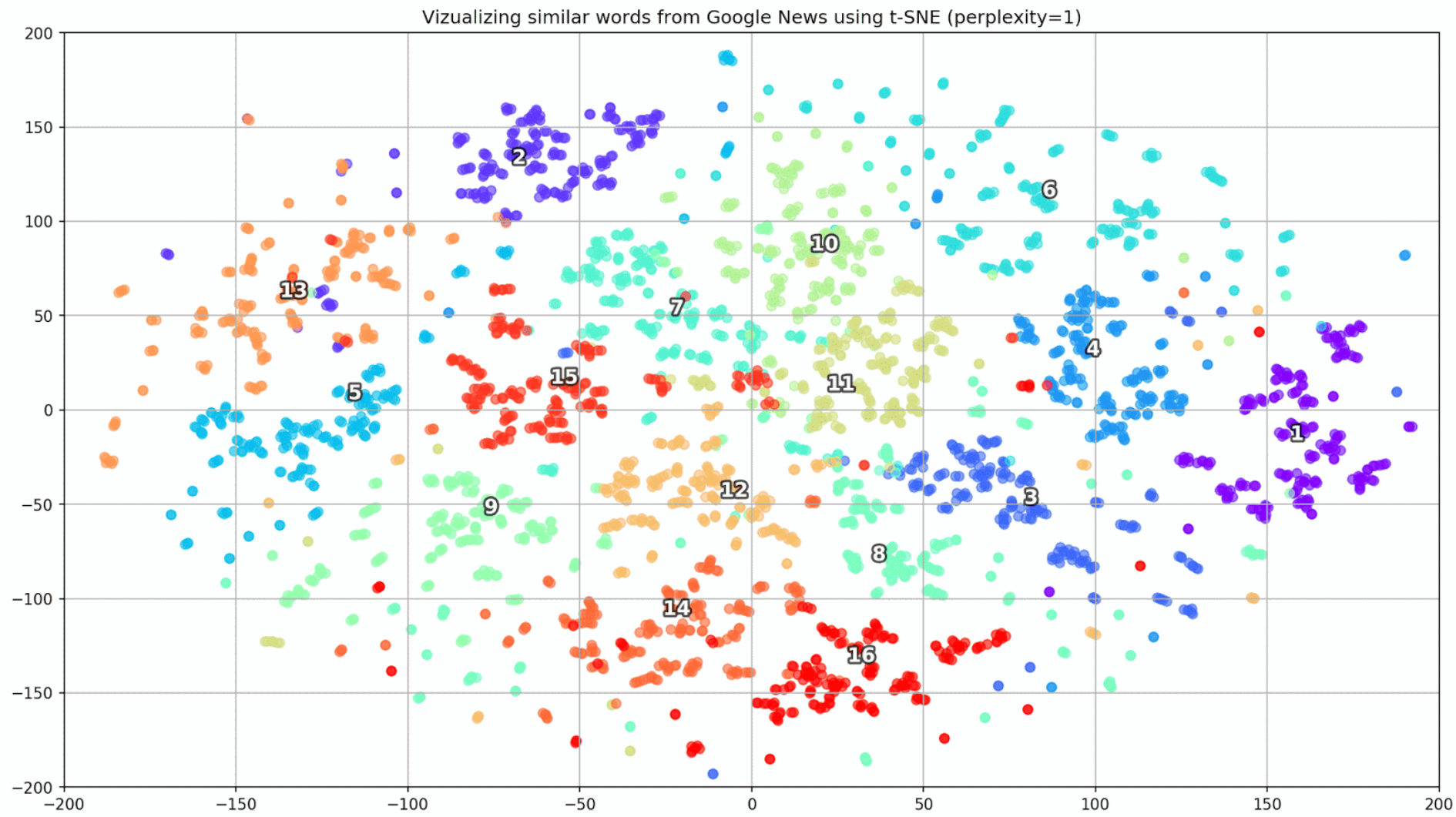
t-distributed stochastic neighbor embedding (t-SNE)

- The t-SNE is a nonlinear dimensionality reduction method well-suited for 2D or 3D visualization of higher-dimensional data.
- Similar items end up as close together points and dissimilar items as distant points.
- Sounds like clustering, however:
  - The t-SNE "clusters" can change considerably with a small change in parameters.
  - Tends to generate "clusters" even when the data doesn't support this.
  - Suitable mostly for visualization, usually not for clustering!

t-distributed stochastic neighbor embedding (t-SNE)

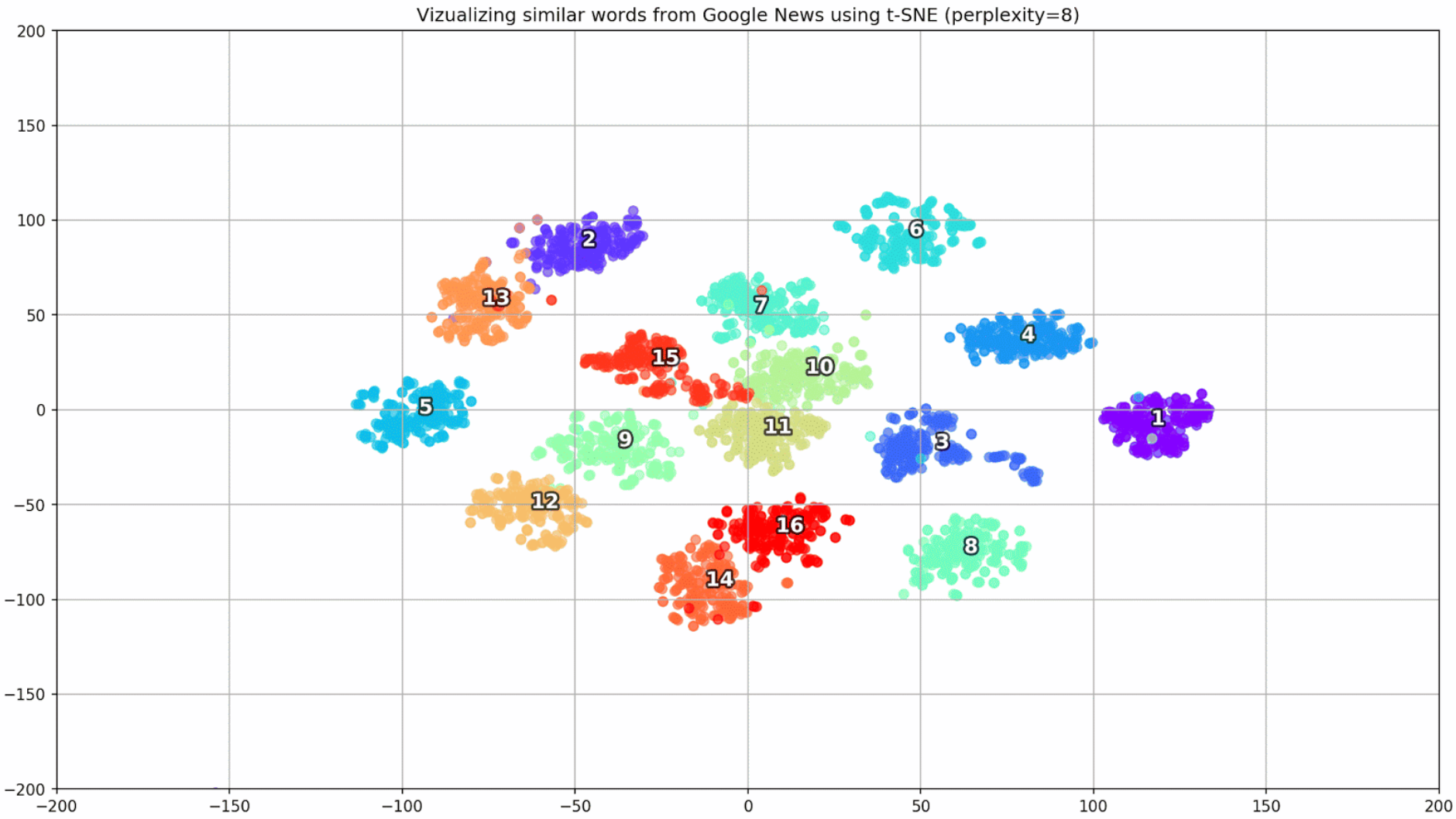
- The t-SNE algorithm consists of two stages.
- First, a probability distribution over pairs of high-dimensional objects is constructed so that similar objects receive a higher probability while dissimilar points receive a lower probability
- Second, a similar probability distribution is generated for the points in the low-dimensional map, and the Kullback–Leibler divergence (KL divergence) is minimized between the two distributions regarding the locations of the points in the map.
- Usually Euclidean distance is used as the similarity metric between the items.

t-distributed stochastic neighbor embedding (t-SNE)



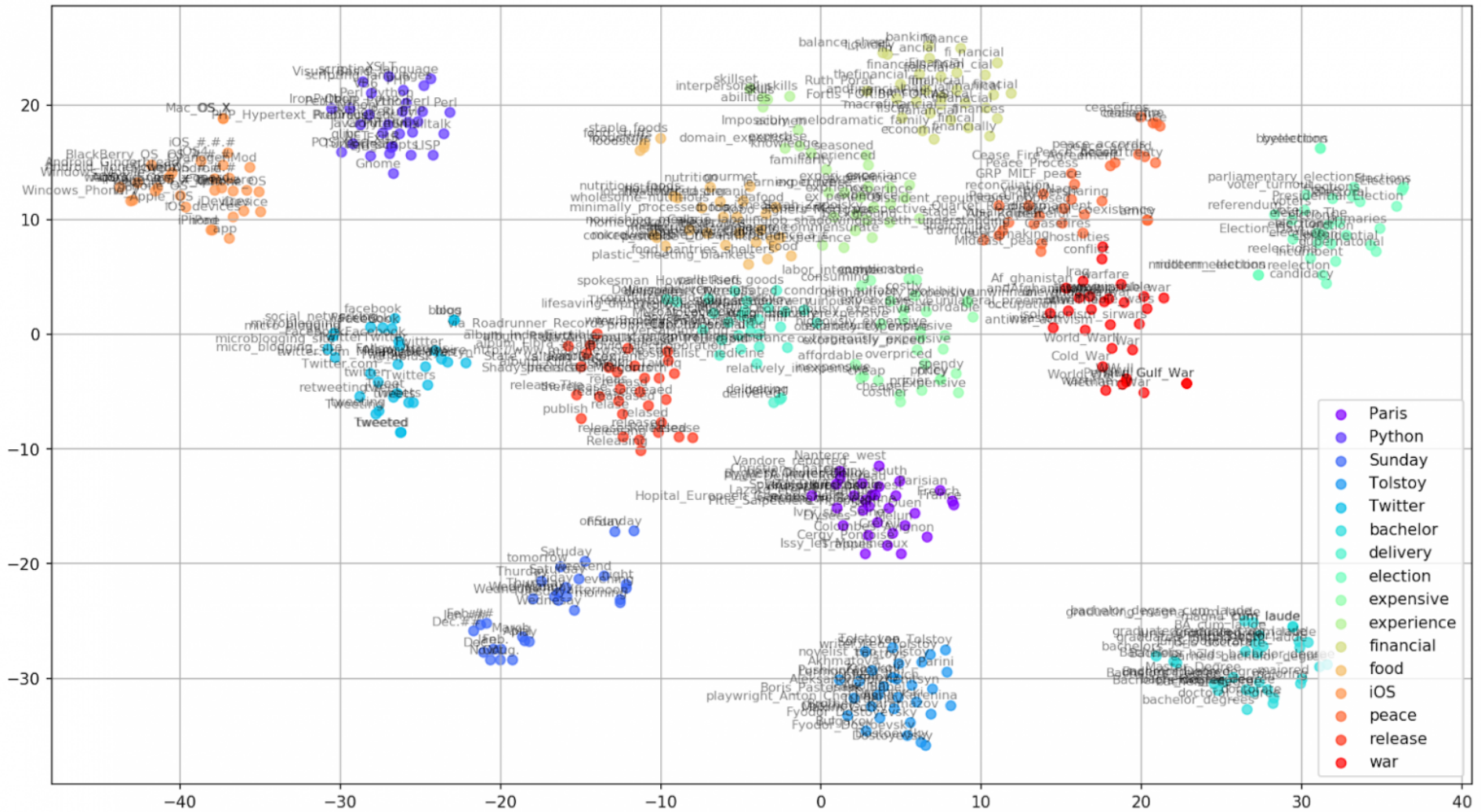
Source: Towards Data Science, Sergey Smetanin (`https://towardsdatascience.com/google-news-and-leo-tolstoy-visualizing-word2vec-word-embeddings-with-t-sne-11558d8bd4d`)

t-distributed stochastic neighbor embedding (t-SNE)



Source: Towards Data Science, Sergey Smetanin (`https://towardsdatascience.com/google-news-and-leo-tolstoy-visualizing-word2vec-word-embeddings-with-t-sne-11558d8bd4d`)

t-distributed stochastic neighbor embedding (t-SNE)



Source: Towards Data Science, Sergey Smetanin (`https://towardsdatascience.com/google-news-and-leo-tolstoy-visualizing-word2vec-word-embeddings-with-t-sne-11558d8bd4d`)

Dimensionality reduction methods

- In addition to PCA and t-SNE many other dimensionality reduction methods exist and preserve different properties of the data
- Nonlinear variant: Kernel principal component analysis
- Multidimensional scaling: find a mapping that preserves the (e.g. Euclidean) *distance* between any two data points
- Linear discriminant analysis: used in classification problems, finds a low dimensional-representation of the data such that separates the classes well
- Feature selection: select an informative subset of the features to represent the data
- Countless other approaches...

Section 6

A checklist for data understanding

A checklist for data understanding

- There are general and specific goals for data understanding
- One important part of data understanding is to get an idea of the data quality.
  - There are standard data quality problems like syntactic accuracy which are easy to check.
- There are various methods to support the identification of outliers:
  - Methods exclusively designed for outlier detection
  - Visualization techniques like boxplots, histograms, scatter plots, projections based on PCA and MDS (multidimensional scaling) that can help to find outliers, but are also useful for other purposes.

A checklist for data understanding

- Missing values are another concern of data quality.
- When there are explicit missing values, i.e. entries that are directly marked as missing, then try to find out of which type – MCAR (Missing Completely at Random), MAR (Missing at Random) or non-ignorable – they are.
- Use domain knowledge, but also classification methods can be applied.
- Be aware of the possibility of hidden missing values that are not explicitly marked as missing. The simplest case might be hidden missing values that have a default value.
- Histograms might help to identify candidates for such hidden missing values when there are unusual peaks.
- However, there is no standard test or technique to identify possible hidden missing values.
    - Therefore, whenever we see something unexpected in the data, hidden missing values of a specific type might be one explanation.

A checklist for data understanding

- Data understanding should also help to discover new or confirm expected dependencies or correlations between attributes.
  - Correlation analysis to solve this task.
  - Scatter plots can show correlations between pairs of attributes.
- Specific application dependent assumptions should be checked
  - For instance, the assumption that a specific attribute follows a normal distribution
- Representativeness of the data cannot always be checked just based on the data, but we have to compare the statistics with our expectations.
  - If we suspect that there is a change in a numerical attribute over time, we can compare histograms or boxplots for different time periods.
  - We can do the same with bar charts for categorical attributes.

A checklist for data understanding

- Check the distributions for each attribute whether there are unusual or unexpected properties like outliers.
  - Are the domains or ranges correct?
  - Do the medians of numerical attributes look correct?
  - Histograms and boxplots for continuous attributes
  - Bar charts for categorical attributes.
- Check correlations or dependencies between pairs of attributes with scatter plots which should be density-based for larger data sets.
  - For small numbers of attributes, inspect scatter plots for all pairs of attributes.
  - For higher numbers of attributes, do not generate scatter plots for all pairs, but only for those ones where independence or a specific dependency is expected.
  - Generate in addition scatter plots for some randomly chosen pairs.