Multivariate Verfahren

7.2 dimensionality reduction: a Motivation

Hannah Schulz-Kümpel

Institut für Statistik, LMU München

Sommersemester 2024

Contents

1. Introduction
   - Euclidean spaces
   - The Curse of dimensionality

2. Feature extraction vs. feature selection

3. Intuitive examples of reducing dimensions
   - Simplest approaches in practice
   - Projections

4. Outlook and topical classification

What is dimensionality reduction

- Remember that we can write data as a matrix.

- In this context, *high dimensional data* refers to very large matrices, specifically one with many variables (i.e. **columns** in matrix-notation, also called **features**).

- In the simplest of terms, **dimensionality reduction** is the compression of data from a higher dimensional matrix to a lower dimensional matrix.

Why would one want to perform dimensionality reduction?

- "Compressing the data matrix" can be very helpful for both data analysis and visualization.
- In the context of visualization, lower dimensional data is obviously easier to plot in a meaningful and comprehensible way.
- In the context of analysis, reducing the dimension often also reduces the computational cost.

  Additionally, it helps us avoid the *Curse of dimensionality*.

What is the *Curse of dimensionality*??

First, let's to take a closer look at Euclidean spaces.

Euclidean spaces I

- Generally speaking, **Euclidean spaces** refer to the spaces $\mathbb{R}^n$, $n \in \mathbb{N}$.
- They are sometimes also referred to as *(Euclidean) n-spaces* or *Cartesian spaces*.
- At this point, we could get into formal algebraic theory where $\mathbb{R}^n$, $n \in \mathbb{N}$ is a **vector space**, specifically a **inner product space with** inner product defined by

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \sum_{i=1}^{n} x_i y_i \,, \quad \forall \; \boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n \,, \; n \in \mathbb{N} \,.$$

Euclidean spaces II

- Instead, we will focus on the more superficial understanding of the Euclidean space $\mathbb{R}^n$, $n \in \mathbb{N}$, which is simply the space of all n-tuples of real numbers:

$$\left\{(x_1, \ldots, x_n) \middle| x_i \in \mathbb{R} \quad \forall i \in \{1, ..., n\}\right\}.$$

- We usually refer the elements of this space as vectors and write $\boldsymbol{x} = (x_1, \ldots, x_n)^\top \in \mathbb{R}^n$, but we can also refer to n-dimensional vectors as *points in* $\mathbb{R}^n$ - just as any element of $\mathbb{R}^n$ may be visualized as both a **point** and a directed vector (i.e. **line**).

- Finally, as vector spaces, Euclidean spaces have two things which are also of interest to us:

Euclidean spaces III

1. **Bases**: For $n \in \mathbb{N}$, any set $B = \{\boldsymbol{b}_1, \dots, \boldsymbol{b}_n\}$ of n vectors in $\mathbb{R}^n$ is a basis of $\mathbb{R}^n$, iff every element of $\mathbb{R}^n$ may be written in a unique way as a finite linear combination of elements of B, i.e.

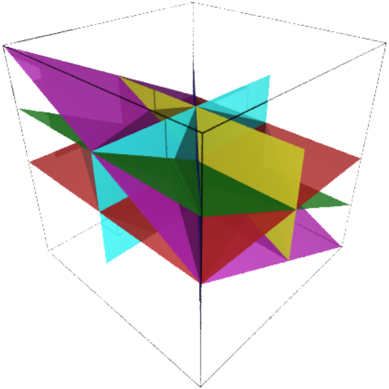
$$\forall \boldsymbol{u} \in \mathbb{R}^n \quad \exists c_1, \dots, c_n \in \mathbb{R} \; : \quad \boldsymbol{u} = \sum_{i=1}^{n} c_i \boldsymbol{b}_i \, .$$

$\longrightarrow$ Bases are not unique, but their elements are always linearly independent and their cardinality equal to the corresponding vector space's dimension.

2. **Subspaces:** For $n \in \mathbb{N}$, any nonempty subset V of $\mathbb{R}^n$ is a subspace, if it is itself a vector space, i.e. iff every linear combination of finitely many elements of V also belongs to V.
$\longrightarrow$ Each subspace of $\mathbb{R}^n$ has a basis with cardinality $< n$.

Example: All hyperplanes of $\mathbb{R}^3$ are subspaces

A **hyperplane** is a subspace whose dimension is one less than that of its ambient space.



By Kilom691 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=37508909

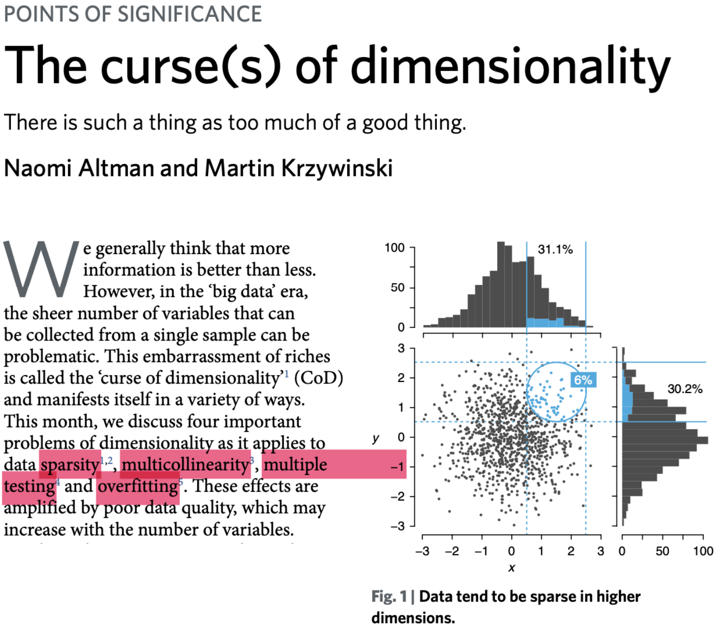
The Curse of dimensionality I

- The curse of dimensionality refers to the phenomena that occur when classifying, organizing, and analyzing high dimensional data that does not occur in low dimensional spaces, specifically the issue of data sparsity and “closeness” of data.

- Remember: High dimensional data refers to data with many variables (or features) → Why?

The Curse of dimensionality I

- The curse of dimensionality refers to the phenomena that occur when classifying, organizing, and analyzing high dimensional data that does not occur in low dimensional spaces, specifically the issue of data sparsity and “closeness” of data.
- Remember: High dimensional data refers to data with many variables (or features) $\rightarrow$ Why?
- Because we consider **each observation as a data point**, thereby each row is a vector in a Euclidean space whose dimension is determined by the number of variables/features.



POINTS OF SIGNIFICANCE

The curse(s) of dimensionality

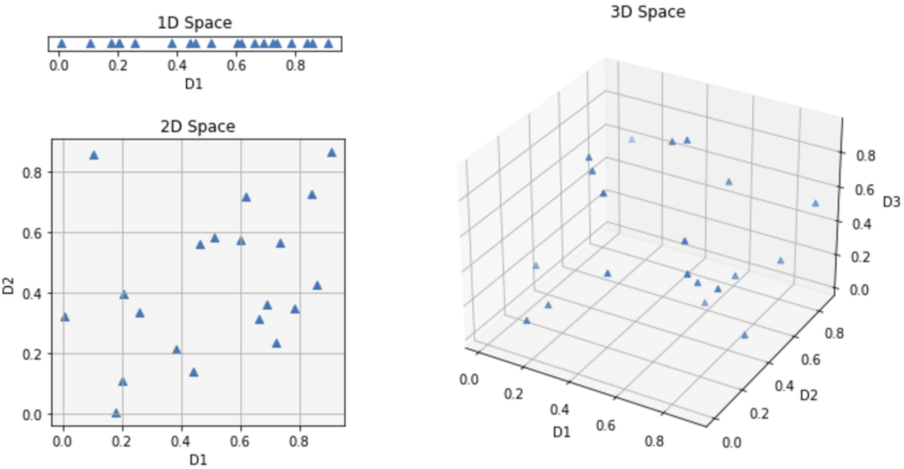
There is such a thing as too much of a good thing.

Naomi Altman and Martin Krzywinski

We generally think that more information is better than less. However, in the 'big data' era, the sheer number of variables that can be collected from a single sample can be problematic. This embarrassment of riches is called the 'curse of dimensionality'[1] (CoD) and manifests itself in a variety of ways. This month, we discuss four important problems of dimensionality as it applies to data sparsity[1,2], multicollinearity[3], multiple testing[4] and overfitting[5]. These effects are amplified by poor data quality, which may increase with the number of variables.

Fig. 1 | Data tend to be sparse in higher dimensions.

Sparse Data - an exemplary visualization

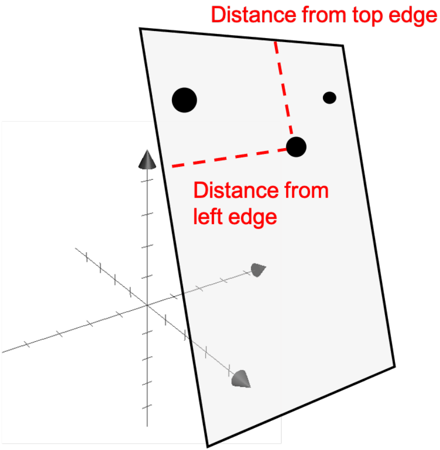


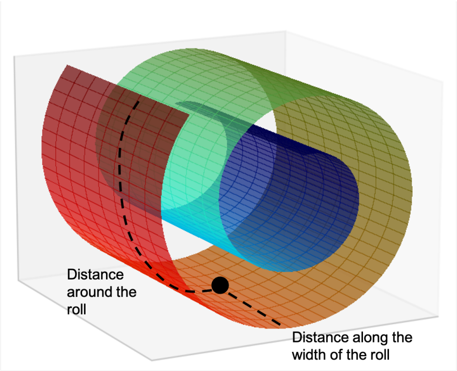
Source: Cunningham, Padraig & Kathirgamanathan, Bahavathy & Delany, Sarah. (2021). Feature Selection Tutorial with Python Examples.

The concept of "intrinsic dimension" I

- In the context of *signal processing*, the term **intrinsic dimension** has a formal definition.
- However, it is also often used more generally as the *number of variables "required" to describe a data point*.
- In some cases, the term "required" can be taken literally, for example:

The concept of "intrinsic dimension" II

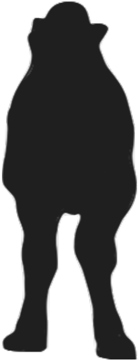




Source: https://mbernste.github.io/posts/intrinsic_dimensionality/

The concept of "intrinsic dimension" III

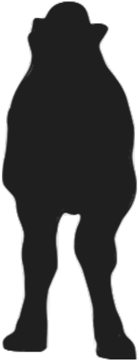
- However, "required number of variables" can also be taken to mean *reasonably 'best' number of variables considering all factors*.
- An example: What does the following depict?

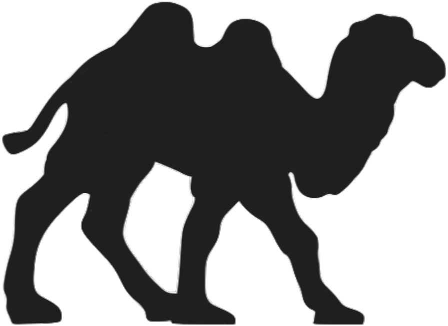


The concept of "intrinsic dimension" III

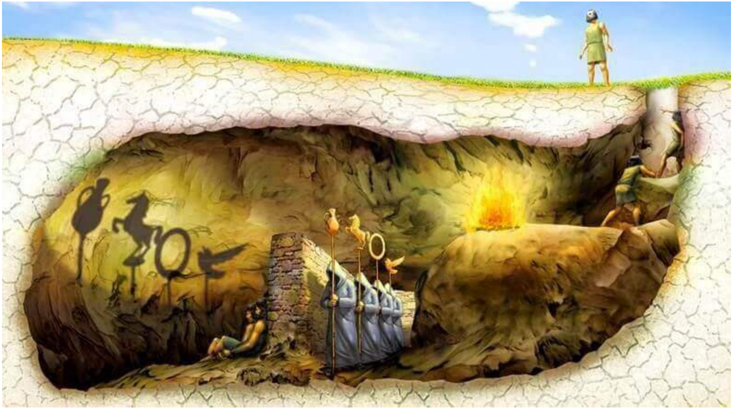
- However, "required number of variables" can also be taken to mean *reasonably 'best' number of variables considering all factors*.

- An example: What does the following depict?



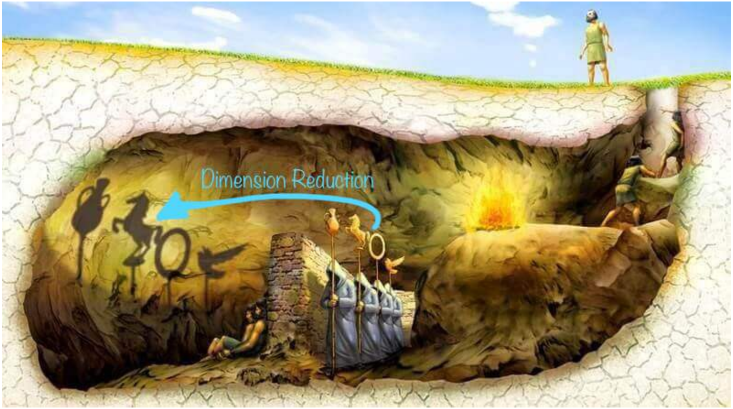


Intrinsic Dimension & Plato's Allegory of the Cave



- In the end the central questions is: **What is the *reasonably 'best' number of variables considering all factors*?**

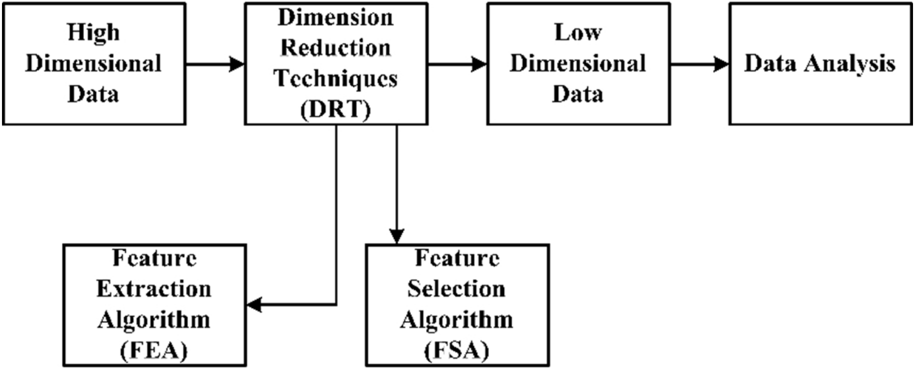
Intrinsic Dimension & Plato's Allegory of the Cave



- In the end the central questions is: **What is the *reasonably 'best' number of variables considering all factors***?

  → There are different methods to find out:

dimensionality reduction: an overview


High Dimensional Data
Dimension Reduction Techniques (DRT)
Low Dimensional Data
Data Analysis
Feature Extraction Algorithm (FEA)
Feature Selection Algorithm (FSA)

Source Ray, P., Reddy, S.S. & Banerjee, T. Various dimensionality reduction techniques for high dimensional data analysis: a review. *Artif Intell Rev* **54**, 3473–3515 (2021). https://doi.org/10.1007/s10462-020-09928-0

Contents

1 Introduction
- Euclidean spaces
- The Curse of dimensionality

2 Feature extraction vs. feature selection

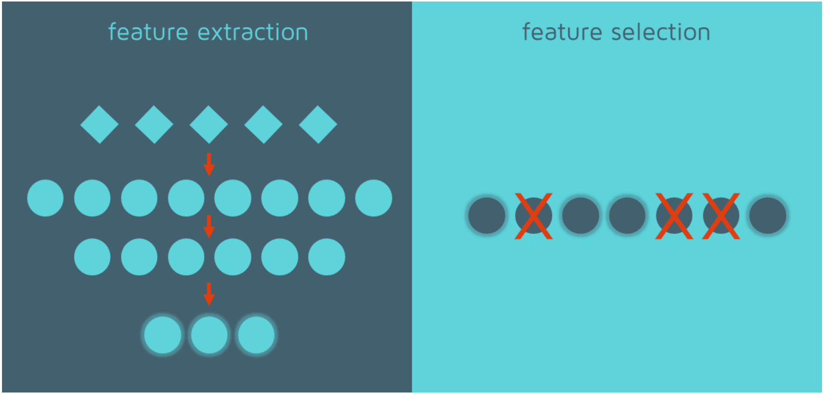
3 Intuitive examples of reducing dimensions
- Simplest approaches in practice
- Projections

4 Outlook and topical classification

Feature extraction vs. feature selection I

In a nutshell:

- **Extraction**: Getting useful features from existing data.
- **Selection**: Choosing a subset of the original pool of features.



Source: https://quantdare.com/what-is-the-difference-between-feature-extraction-and-feature-selection/

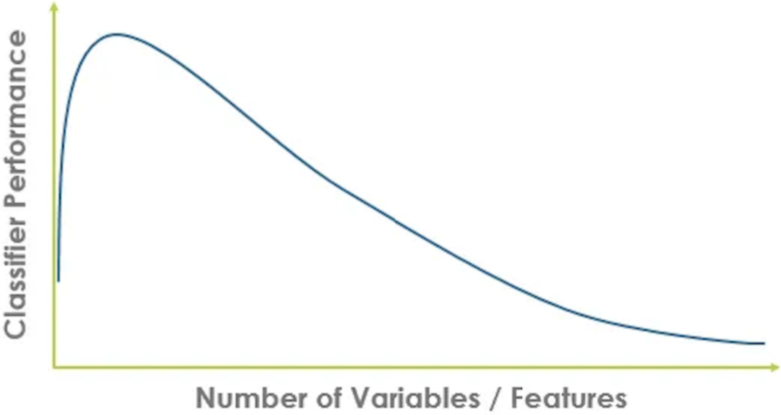
Feature extraction vs. feature selection II

The main difference:

- *Feature Extraction* transforms data, even from arbitrary formats such as text or images, into numerical features that can then be processed using statistical/ML models.
- *Feature Selection*, on the other hand, can only be applied to numerical features and merely eliminates potentially redundant/not strictly necessary features.

  Feature selection is, for example, applied in *stepwise regression*.

Often, the issue is simplified as follows...



Do you agree? ⟶ **Discuss**

Contents

1. Introduction
   - Euclidean spaces
   - The Curse of dimensionality
2. Feature extraction vs. feature selection
3. Intuitive examples of reducing dimensions
   - Simplest approaches in practice
   - Projections
4. Outlook and topical classification

Motivational example: consulting on a psychological study I

Consider the following situation: A group of psychologists have collected patient data, specifically

- A *target variable*, which they want to regress on the variables
- *age*, *weight*, *height* as well as
- 20 *metric scores*, each from a different psychological tests, measured **on different scales**.

Can you come up with some simple ways of reducing the number of independent variables in their planned regression?

Motivational example: consulting on a psychological study II

- Very simplest idea: Drop those variables that do not seem necessary content-wise (such as possibly *height*).

  ⟶ This would technically fall under **feature selection**.

- A very simple way of reducing the number of independent variables which would fall under **feature extraction**:
  - Summarizing all (or some of the) 20 metric scores into one new variable by calculating a weighted average.
  - The most obvious problem with this approach in our setting would be the different scales.

  ⟶ This could be avoided by first standardizing all scores and then summarizing them by calculating a weighted average.

Projections

- A more theoretical concept which may be used for dimensionality reduction are *projections*.
- Generally, the term **projection** refers to an *idempotent mapping* from a set (or other mathematical structure) into a subset (or sub-structure).
- **Idempotent**, in this context, means that projecting once is equal to projecting twice.

  I.e., if the projection is denoted by P, idempotence implies $P = P \circ P$.
- For statisticians, the most important projections are *linear projections*:

Linear Projection I

Definition (Linear Projection)

A *linear projection* (or simply *projection* in the context of linear algebra) on a vector space V is a linear operator $P : V \longrightarrow V$ so that $P^2 = P$.

Specifically,

(i) we define any linear projection via a **square** matrices P that is equal to its square, called *projection matrix* and

(ii) any square matrix P with $P^2 = P = P^\top$ defines an *orthogonal projection*.

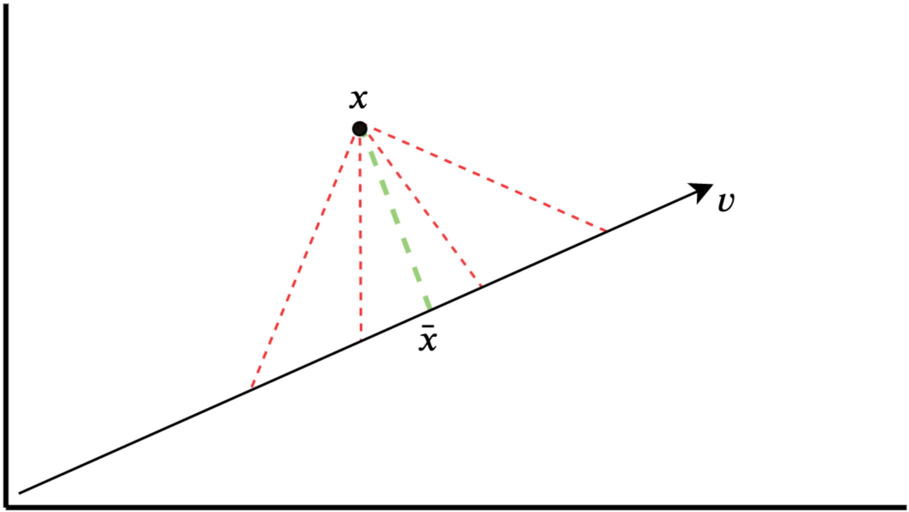
- A vector $v \in V$ is then linearly projected by multiplying the projection matrix P with it, i.e. $v \longmapsto Pv$.
- For example, the identity matrix $\boldsymbol{I}_p$, $p \in \mathbb{N}$, is an orthogonal projection matrix which projects any vector $v \in \mathbb{R}^p$ onto itself.

Linear Projection II

Intuitive Example: Consider the directed vector $\vec{v} \in \mathbb{R}^2$, drawn as a finite straight line pointing in a given direction, and the point $x \in \mathbb{R}$ not on this straight line but in the same two-dimensional space.

- The projection of x, i.e. Px for the appropriate **projection matrix** $P \in \mathbb{R}^{2\times 2}$, is a function that returns the point "closest" to x along the vector line $\vec{v}$.
- In most contexts, closest refers to Euclidean distance, i.e. the point $\bar{x} \in \mathbb{R}^2$ on the vector line $\vec{v}$ that minimizes $\sqrt{\sum_{i=1}^{2}(x_i - \bar{x}_i)^2}$.
- In the following, the green dashed line shows the orthogonal projection, and red dashed lines indicate other potential (non-orthgonal) projections that are further away from x than $\bar{x}$ in the Euclidean space:

Linear Projection III



Source: https://bookdown.org/ts_robinson1994/10_fundamental_theorems_for_econometrics/linear-projection.html

Linear regression as linear projection I

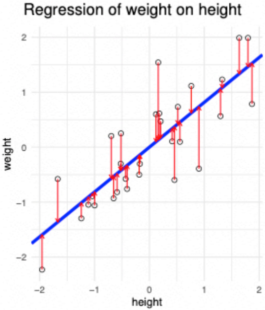
- Recall: The *frequentist* point estimates for the model coefficients can then be calculated via ordinary least squares (OLS): $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}\,.$

$\longrightarrow$ The point predictions are given by $\hat{\boldsymbol{y}} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-\mathbf{1}} \boldsymbol{X}^\top \boldsymbol{y}\,.$

- Note that $P := \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-\mathbf{1}} \boldsymbol{X}^\top$ is an orthogonal projection, because

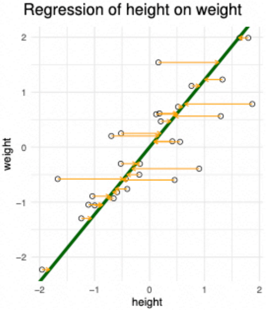
$$\begin{aligned} P^\top &= (\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top)^\top \\ &= \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \\ &= P\,, \end{aligned} \quad \text{and} \quad \begin{aligned} P^2 &= \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \\ &= \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \\ &= P\,. \end{aligned}$$

Linear regression as linear projection II



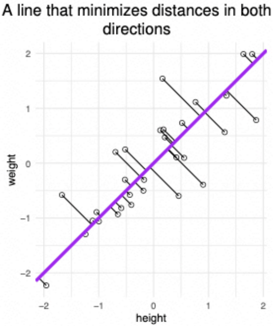
Regression of weight on height

The blue line minimizes the sum of squares of the vertical residuals (in red)



Regression of height on weight

The green line minimizes the sum of squares of the horizontal residuals (in orange)



A line that minimizes distances in both directions

The purple line minimizes the sums of squares of the orthogonal projections

Marginal distribution as projection I

- Given the joint distribution of a random vector $\boldsymbol{X} = (X_1, \ldots, X_p)^\top$, $p \in \mathbb{N}$, the marginal distribution of the ith entry $(F_{X_i}(x_i) = F_{X_1 \ldots X_p}(\infty, \ldots, \infty, x_i, \infty, \ldots, \infty))$ is the **projection** of the distribution $F_{X_1 \ldots X_p}$ of the random vector $\boldsymbol{X}$ onto the axis x_i, and is completely determined by the distribution of the original vector.
- Equivalently, the marginal distribution of several entries, i.e. ith – $(i+2)$th $(F_{X_i, X_{i+1}, X_{i+2}}(x_i, x_{i+1}, x_{i+2}) = F_{X_1 \ldots X_p}(\infty, \ldots, \infty, x_i, x_{i+1}, x_{i+2}, \infty, \ldots, \infty))$, is again a **projection**, just to a higher dimensional subspace.

Marginal distribution as projection II

Consider the specific example of two jointly continuously OR discretely distributed random variables X_1 and X_2 with a joint CDF $F_{X_1X_2}$.

Recall:

1. The marginal CDFs (which may be seen as projections) are given by $F_{X_1}(x_1) = F_{X_1X_2}(x_1, \infty)$ and $F_{X_2}(x_2) = F_{X_1X_2}(\infty, x_2)$.

2. We can estimate a **discrete** probability function for the random vector $\boldsymbol{X} = (X_1, X_2)^\top$ by using *relative frequency* : Given a sequence of data points $\{(x_{1i}, x_{2i})\}_{i=1,\dots,n}$, $(x_{1i}, x_{2i})^\top \in \mathbb{R}^2$ we get

$$\widehat{p} : \mathbb{R}^2 \longrightarrow \mathbb{R}\,, \quad x \longmapsto \frac{\#\, x \text{ appears in the sequence } \{(x_{1i}, x_{2i})\}_{i=1,\dots,n}}{n}\,.$$

Marginal distribution as projection III

- **Question**: How do we estimate the corresponding marginal distributions $\widehat{p}_{X_1}$ and $\widehat{p}_{X_2}$?

- **Answer**: Simply consider all observations of each variable as two separate sequences $\{x_{1i}\}_{i=1,\dots,n}$ and $\{x_{2i}\}_{i=1,\dots,n}$ $x_{1i}, x_{2i} \in \mathbb{R}$ and then define $\widehat{p}_{X_1}$ and $\widehat{p}_{X_2}$ w.r.t. just as $\widehat{p}$ above, but with regard to these sequences, respectively.

- As an intuition, the sequences $\{x_{1i}\}_{i=1,\dots,n}$ and $\{x_{2i}\}_{i=1,\dots,n}$ are equivalent to the non-zero entries of the vectors we get by *linearly projecting* each element of the sequence $\{(x_{1i}, x_{2i})\}_{i=1,\dots,n}$ using

$$P_{X_1} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad P_{X_2} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \text{ respectively.}$$

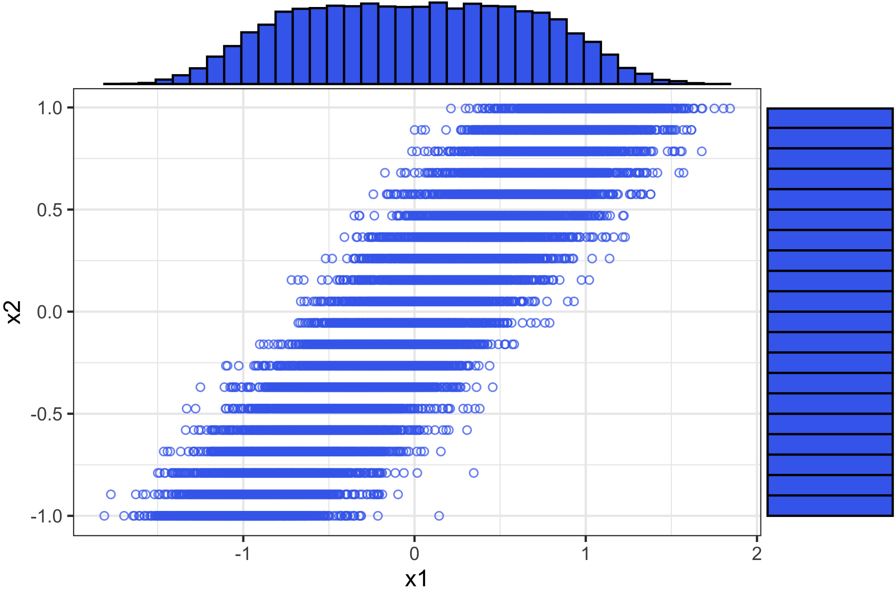
Empirical marginal distribution (rel.freq.) - example in R I

```
library(ggplot2)
library(ggExtra)

set.seed(2023)
data<-data.frame(x2=rep(seq(-1,1,by=0.105),each=1000),
                 x1=unlist(lapply(seq(-1,1,by=0.105),
                    function(x){rnorm(1000,x,sd=0.25)})))

ggMarginal(ggplot(data, aes(x=x1, y=x2)) +
  geom_point(fill = NA, colour = "royalblue2",shape=01,
  alpha=0.75) + theme(legend.position="none")+ theme_bw(),
  type="histogram",xparams = list(binwidth=c(0.1),
  fill="royalblue2"),
  yparams = list(binwidth=c(0.1),fill="royalblue2"))
```

Empirical marginal distribution (rel.freq.) - example in R II



Contents

1 Introduction
- Euclidean spaces
- The Curse of dimensionality

2 Feature extraction vs. feature selection

3 Intuitive examples of reducing dimensions
- Simplest approaches in practice
- Projections

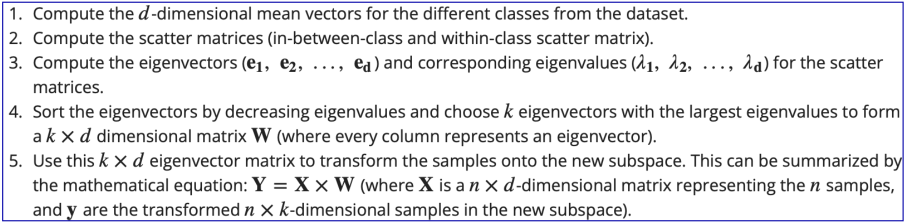
4 Outlook and topical classification

Note: There are many methods for dimensionality reduction!

- In fact, especially methods we have grouped as unsupervised learning are often also/primarily used for dimensionality reduction!
- Others popular methods, which we will not be seeing in the lecture, but address the issue of multicollinearity well:
  - *t-distributed Stochastic Neighbor Embedding (tSNE)*
  - *Uniform Manifold Approximation and Projection (UMAP)*
- We have already covered at least one supervised learning method used for dimensionality reduction though!

Example: LDA for dimensionality reduction

- **Linear discriminant analysis** (LDA) is both a classification (supervised learning) method *and* a classic feature extraction method!
- Specifically, we may utilize LDA for dimensionality reduction using the following algorithm:



1. Compute the d-dimensional mean vectors for the different classes from the dataset.
2. Compute the scatter matrices (in-between-class and within-class scatter matrix).
3. Compute the eigenvectors ($\mathbf{e_1}, \ \mathbf{e_2}, \ \ldots, \ \mathbf{e_d}$) and corresponding eigenvalues ($\lambda_1, \ \lambda_2, \ \ldots, \ \lambda_\mathbf{d}$) for the scatter matrices.
4. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $k \times d$ dimensional matrix $\mathbf{W}$ (where every column represents an eigenvector).
5. Use this $k \times d$ eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the mathematical equation: $\mathbf{Y} = \mathbf{X} \times \mathbf{W}$ (where $\mathbf{X}$ is a $n \times d$-dimensional matrix representing the n samples, and $\mathbf{y}$ are the transformed $n \times k$-dimensional samples in the new subspace).

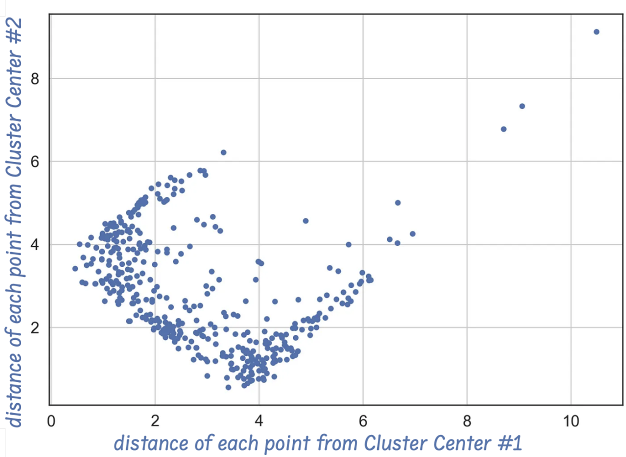
Source: mlxtend documentation

Back to unsupervised learning: Clustering for dimensionality reduction I

- Clustering algorithms, such as k-means, can also be used for dimensionality reduction by calculating the distance of each point to each cluster center.

$\longrightarrow$ Thereby, the number of features is reduced to the number of clusters.

Back to unsupervised learning: Clustering for dimensionality reduction I



2 Reduced Dimensions — using K-Means on the Boston Housing Data

Edited from:

https://medium.com/analytics-vidhya/less-known-applications-of-k-means-clustering-dimensionality-reduction-anomaly-detection-and-908f4bee155f

Outlook

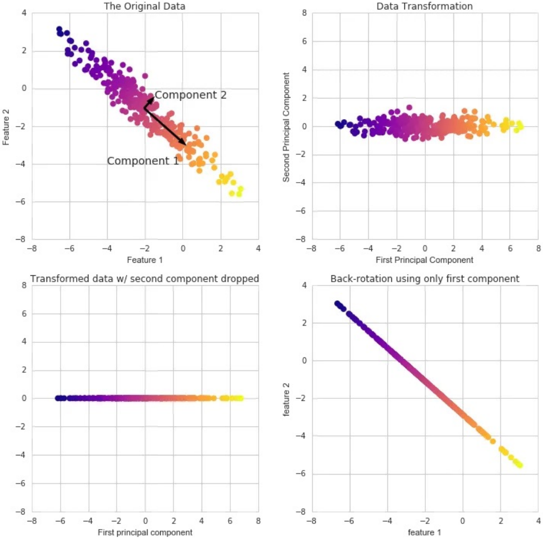
In the next 2 lectures, we will cover the topics of

- **More clustering methods, such as k-means**
- **Principal component analysis (PCA)**

PCA teaser I

- The main idea of PCA is to reduce the dimensionality of a data set consisting of many variables correlated with each other, while retaining the variation present in the dataset, up to the maximum extent.
- This is done by transforming the variables to a new set of variables, which are known as the principal components (PCs).
- Using the PCs, the projection onto a lower dimensional subspace then works similarly to LDA.
- *The difference is that PCA focuses on maximizing the variance in the data, while LDA aims to maximize the separability between different classes in a classification problem.*

PCA teaser II



Source: `https://medium.com/@mayureshrpalav/principal-component-analysis-feature-extraction-technique-3f480d7b9697`

PCA teaser III

→ PCA is a feature extraction method.

- Mathematically, we will require the following matrix decomposition methods to perform PCA:
  - Eigendecomposition
  - Singular value decomposition (SVD)