PCA Revealed

Part 5: Geometric Approach

Gaston **S**anchez

August 2014

Content licensed under CC BY-NC-SA 4.0

Readme

License:

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
http://creativecommons.org/licenses/by-nc-sa/4.0/

You are free to:

Share — copy and redistribute the material

Adapt — rebuild and transform the material

Under the following conditions:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.

NonCommercial — You may not use this work for commercial purposes.

Share Alike — If you remix, transform, or build upon this work, you must distribute your contributions under the same license to this one.

Reminder

PCA

Principal Components Analysis (PCA) allows us to study and explore a set of quantitative variables measured on a set of objects.

Core Idea

With PCA we seek to reduce the dimensionality (reduce the number of variables) of a data set while retaining as much as possible of the variation present in the data.

CC BY-NC-SA 4.0

Presentation

About

In these slides we cover PCA from a geometric perspective.

Working Principle

The underlying notion of this approach is that of Projected Inertia, and the intensive use of geometric principles.

Visually Intended

Visualization plays a notable role in the Geometric Approach of PCA. One of the main reasons to reduce dimensions is to obtain graphical representations of data.

Data considerations

 CC BY-NC-SA 4.0

Data Structure

Data

The analyzed data takes the form of a table (i.e. `matrix`) $\mathbf{X}$:

$$\mathbf{X}_{n,p} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

- n objects in the rows
- p quantitative variables in the columns

 CC BY-NC-SA 4.0

Data Considerations

Variables

The p variables in $\mathbf{X}$ are denoted by $X_1, X_2, \dots, X_p$

Mean centered

For convenience, we will assume that the data is centered, i.e. Variables with mean $= 0$:

$$\bar{X}_j = \sum_{i=1}^{n} x_{ij} = 0$$

 CC BY-NC-SA 4.0

Data from a geometric perspective

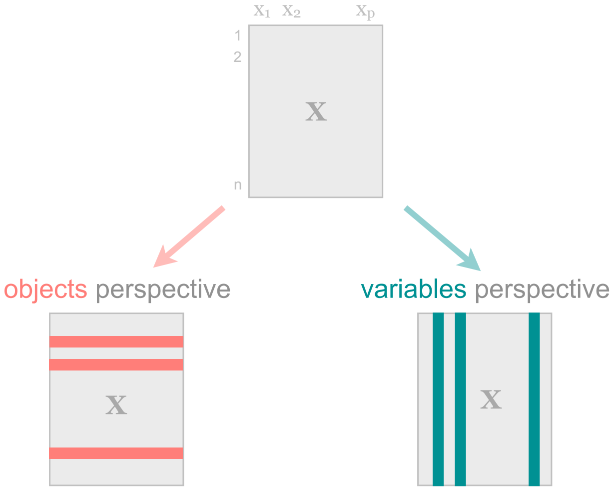
CC BY-NC-SA 4.0

Geometric Frame of Mind

Looking at PCA from a geometric standpoint requires you to think about data in terms of points living in a multidimensional space —both objects and variables—

Data Perspectives

looking at a data matrix from two perspectives



Objects and Variables Perspectives

Data Perspectives

We are interested in analyzing a data set from both perspectives: objects and variables

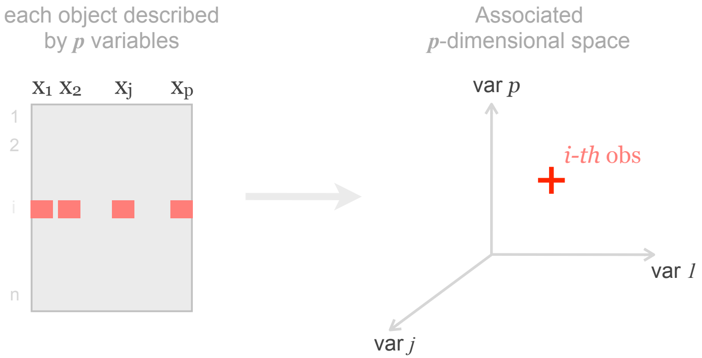
Main Interests

At its simplest we are interested in 2 fundamental purposes:

- Study (dis)similarities among objects
- Study relationships among variables

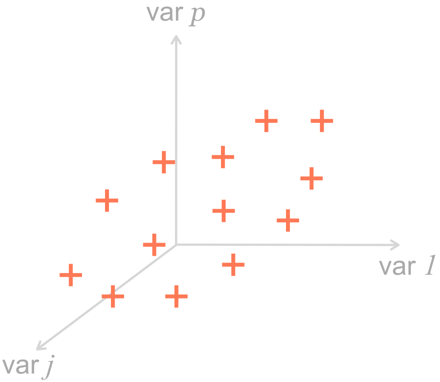
 CC BY-NC-SA 4.0

Objects in Multidimensional Space



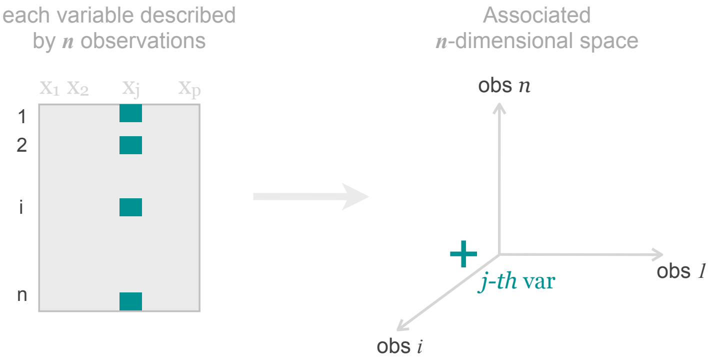
Cloud of objects

Objects as points in a p-dimensional space



 CC BY-NC-SA 4.0

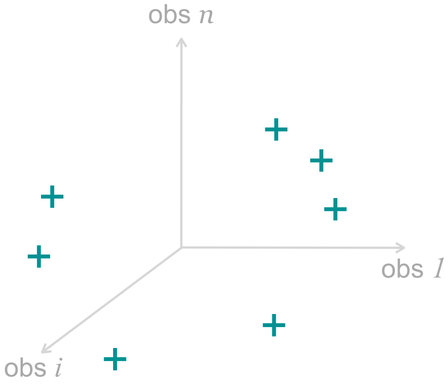
Variables in Multidimensional Space



 CC BY-NC-SA 4.0

Cloud of variables

Variables as points in a n-dimensional space



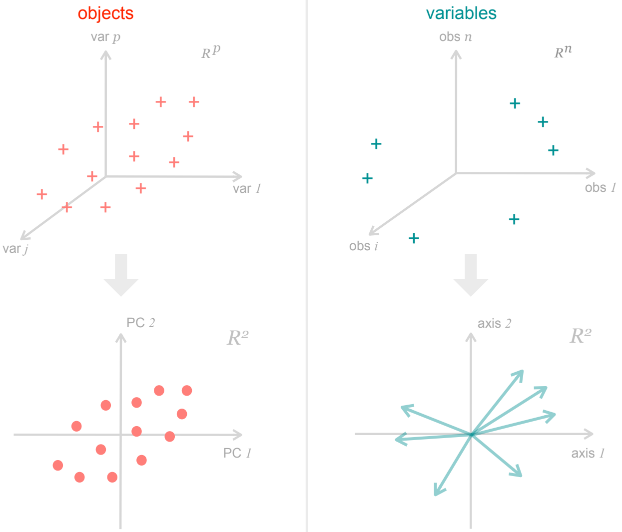
Overall Goal

PCA Visualization

We look for the "best" graphical representation that allows us to visualize the data in a low dimensional space (usually 2-dimensions).

CC BY-NC-SA 4.0

Best representation in low dimensional space



 CC BY-NC-SA 4.0

Low Dimensional Projections

Geometric mindset

To help you understand the main idea of PCA from a geometric standpoint, I'd like to begin showing you my *mug-data* toy example.

Key Message

The "name of the game" is **projection**: PCA involves projecting the data onto a low-dimensional space that best captures the original dispersion in the data.

 CC BY-NC-SA 4.0

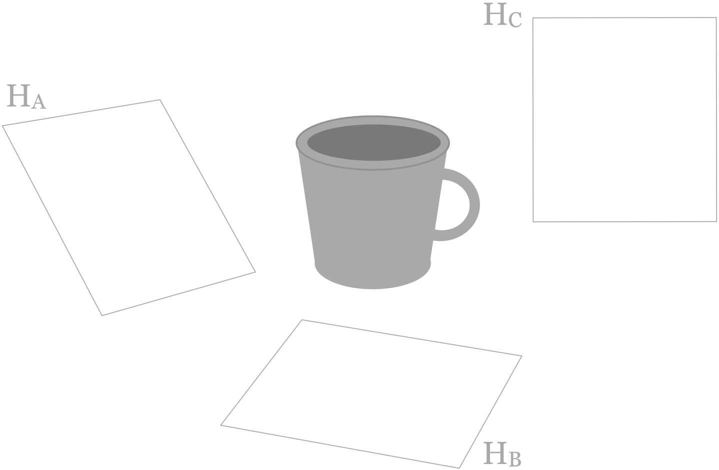
Illustrational View

Example

Imagine we have some data in a "high-dimensional space"

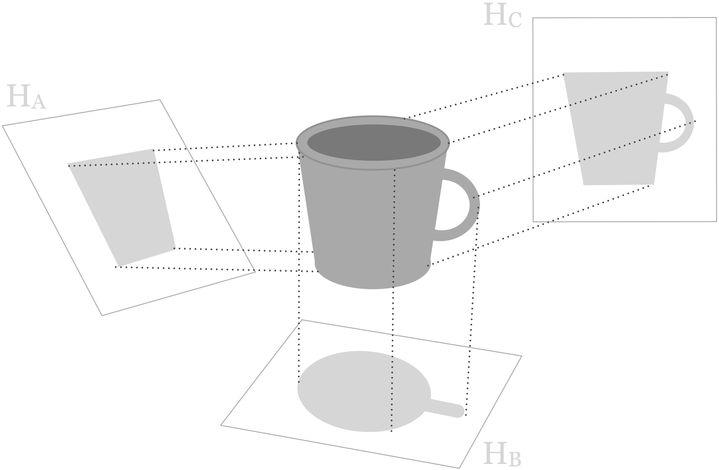
 CC BY-NC-SA 4.0

We are looking for Candidate Subspaces



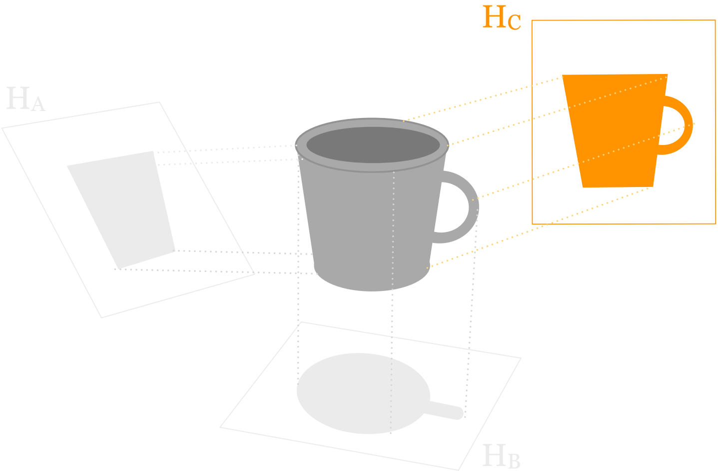
 CC BY-NC-SA 4.0

with the best low-dimensional representation



 CC BY-NC-SA 4.0

Best low-dimensional projection

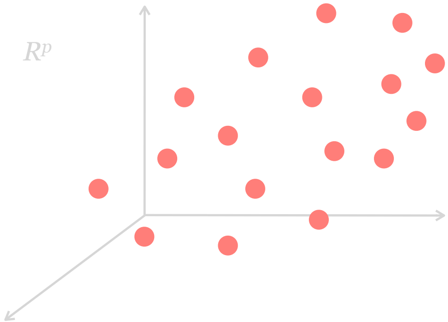


Projections!!!

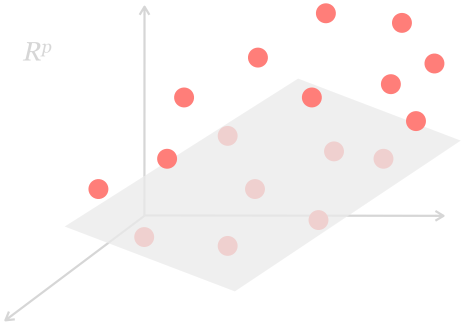
Projection

We want to find a subspace that provides us the best projection of the data

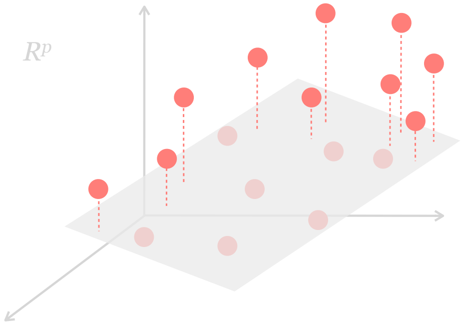
Objects in a high-dimensional space



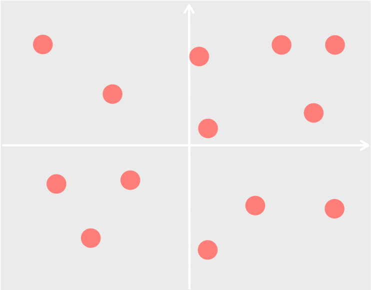
We look for a subspace such that



the projection of points on it



is the best low-dimensional representation



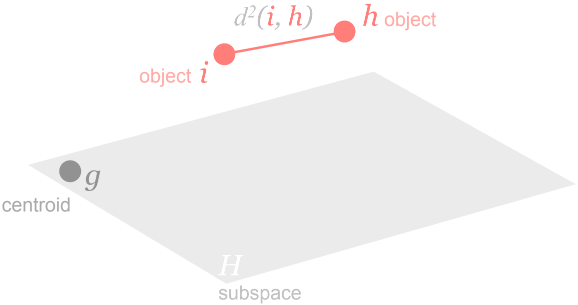
Projected Distances

Best Projection?

Looking for the best low-dimensional projection means that we want to find a subspace in which the projected distances among points are as much similar as possible to the original distances.

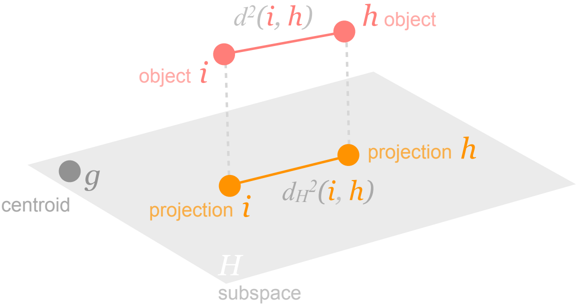
 CC BY-NC-SA 4.0

Focus on distances between objects



 CC BY-NC-SA 4.0

We want projected dists to preserve original dists



$d^2(i, h)$ as close as possible to $d_H{}^2(i, h)$

Distances and Dispersion

Dispersion of Data

Focusing on distances among all pairs of objects implicitly entails taking into account the **dispersion** (i.e. variation) of the data.

Data Configuration

The reason to pay attention to distances and dispersion is to summarize in a quantitative way the original configuration of the data points.

 CC BY-NC-SA 4.0

Sum of Square Distances

Pair-wise Square distances

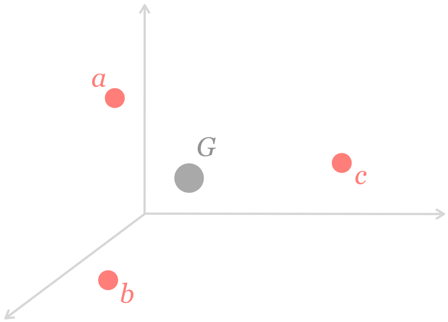
One way to consider the dispersion of data (in a mathematical form) is by adding the square distances among all pairs of points.

Square distances from centroid

Another way to measure the dispersion of data is by considering the square distances of all points around the center of gravity (i.e. centroid)

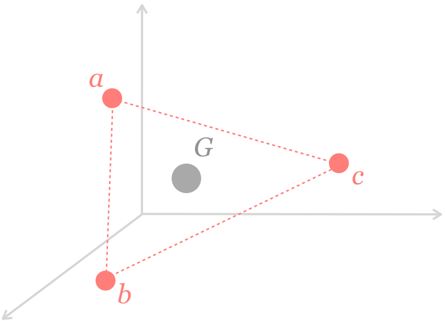
 CC BY-NC-SA 4.0

Imagine 3 points and its centroid



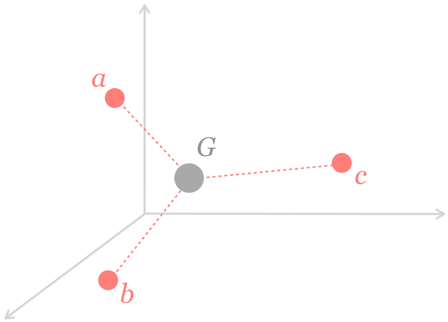
 CC BY-NC-SA 4.0

Sum of square distances



Sum of all square distances $= d^2(a,b) + d^2(a,c) + d^2(b,c)$

Sum of 2 $\times$ square distances w.r.t. centroid



Sum of all square distances $= 2d^2(a,G) + 2d^2(b,G) + 2d^2(c,G)$

 CC BY-NC-SA 4.0

Inertia

CC BY-NC-SA 4.0

Inertia

Inertia

To better take into account the dispersion of the data we must use the concept of **Inertia**.

Idea

Simply put, we use the term Inertia to convey the idea of dispersion or *information* (variation) contained in the data.

 CC BY-NC-SA 4.0

Inertia Concept (con't)

Moment of Inertia

Inertia is a term borrowed from the *moment of inertia* in mechanics.

Inertia in Multivariate Methods

In multivariate methods, the term **Inertia generalizes the notion of variance**. Think of Inertia as a "multidimensional variance"

 CC BY-NC-SA 4.0

Formula of Total Inertia

Formula

The Total Inertia, I, is a weighted sum of square distances among all pairs of objects:

$$I = \frac{1}{2n^2} \sum_{i=1}^{n} \sum_{h=1}^{n} d^2(i,h)$$

Formula of Total Inertia

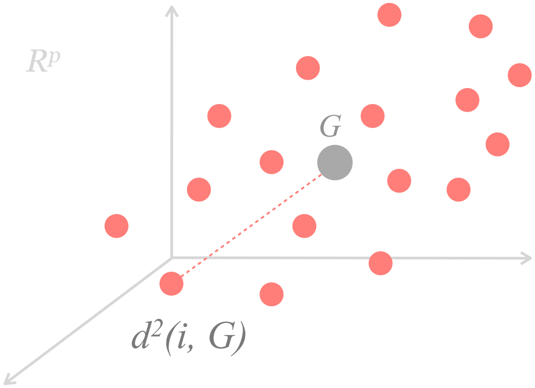
Equivalent Formula

Equivalently, the Total Inertia can be calculated in terms of the center of gravity G:

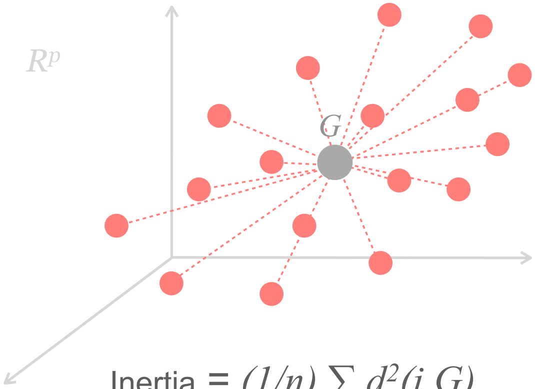
$$I = \frac{1}{n} \sum_{i=1}^{n} d^2(i, G)$$

The Inertia is an average sum of square distances around the centroid G

Data Points with their Centroid



Inertia around the Centroid



$$\text{Inertia} = (1/n) \sum d^2(i,G)$$

Computing Inertia

Notation

$x_{i.}$ i-th object $(i = 1, \dots, n)$

m_i mass of i-th object (usually $m_i = \frac{1}{n}$)

G center of gravity (if data is mean-centered then $G = 0$)

$d^2(i, G)$ distance between i-th object and centroid G

Inertia Formula

$$
\begin{aligned}
Inertia &= \sum_{i=1}^{n} m_i d^2(i, G) && (1) \\
&= \sum_{i=1}^{n} \frac{1}{n}(x_{i.} - G)'(x_{i.} - G) && (2) \\
&= \frac{1}{n} tr(X'X) && (3) \\
&= \frac{1}{n} tr(XX') && (4)
\end{aligned}
$$

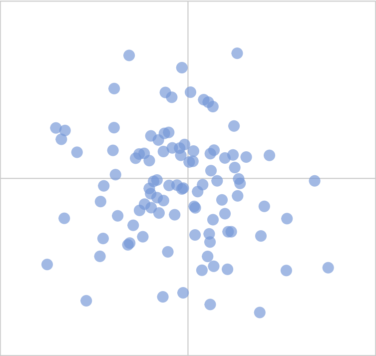
Inertia? What for?

What's Important?

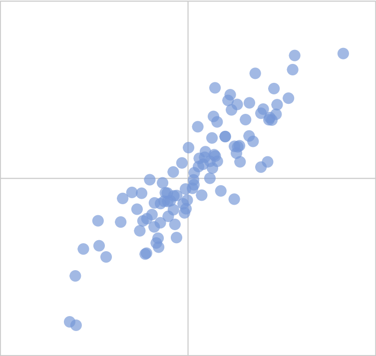
Two data sets can have the same inertia. The amount of dispersion is important, but it is also important the shape-form of that dispersion.

 CC BY-NC-SA 4.0

Two data sets with similar inertia but different shape



Inertia = 2.02



Inertia = 2

 CC BY-NC-SA 4.0

Projected Inertia and Dimension Reduction

 CC BY-NC-SA 4.0

Inertia Concept

Inertia and PCA

In PCA we look for a low-dimensional subspace having Projected Inertia as close as possible to the Original Inertia.

Criterion

The criterion used for dimensionality reduction implies that the inertia of a cloud of points in the optimal subspace is maximum, but that would still be less than that in the true space.

Subspace Preserving Information

Global Criterion

We want the subspace that maximices the total dispersion, that is, the total inertia (maximize projected inertia)

Interpretation

This means that we want to find the subspace that keeps the maximum amount of information of the original configuration

 CC BY-NC-SA 4.0

Criterion

Maximize Projected Inertia

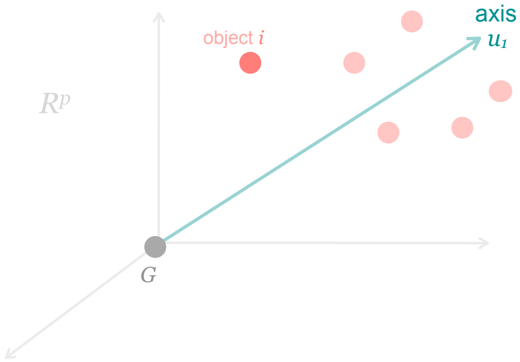
We want to maximize the Projected Inertia on subspace H:

$$max \text{ projected} \sum_i d_H^2(i, G)$$

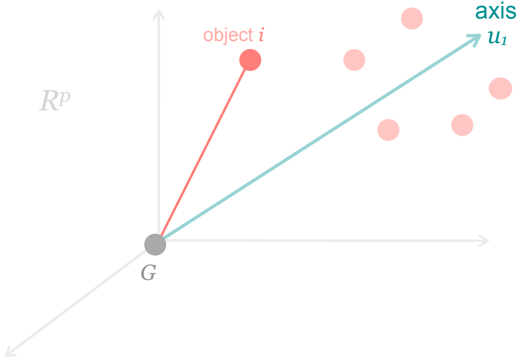
Axis of Inertia

To find the subspace H we can look for each of its dimensions (i.e. axes) $\mathbf{u_1}, \mathbf{u_2}, \dots, \mathbf{u_k}$

Projection 1

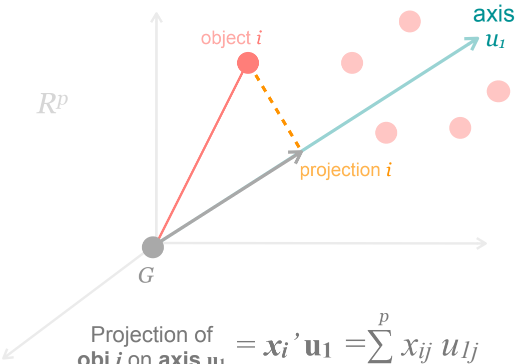


Projection



 CC BY-NC-SA 4.0

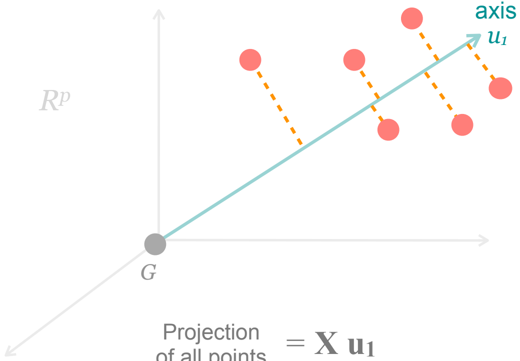
Projection



Projection of **obj** i on **axis** $\mathbf{u}_1$ $= \mathbf{x}_i' \mathbf{u}_1 = \sum_{j=1}^{p} x_{ij} \, u_{1j}$

CC BY-NC-SA 4.0

Projection



Projection of all points $= \mathbf{X}\,\mathbf{u_1}$

PCs as Data Projections

It turns out that:

Looking for all the projections $Z_1 = \mathbf{X}\mathbf{u_1}, Z_2 = \mathbf{X}\mathbf{u_2}, \dots, Z_k = \mathbf{X}\mathbf{u_k}$ implies looking for the Principal Components.

CC BY-NC-SA 4.0

Finding PCs

CC BY-NC-SA 4.0

How to find the 1st PC

In order to find the first principal component $Z_1 = \mathbf{X}\mathbf{u_1}$, we need to find $\mathbf{u_1}$ such that

$$\max_{\mathbf{u_1}} \; \mathbf{u_1}'\mathbf{X}'\mathbf{X}\mathbf{u_1}$$

subject to $\mathbf{u_1}'\mathbf{u_1} = 1$

What to do?

Being a maximization problem, the typical procedure to find the solution is by using the Lagrangian multiplier method.

 CC BY-NC-SA 4.0

Lagrangian Multiplier

Finding 1st PC

Using Lagrange multipliers we get:

$$\mathbf{u_1'X'Xu_1} - \lambda(\mathbf{u_1'u_1} - 1)$$

Differentiation with respect to $\mathbf{u_1}$ gives:

$$\mathbf{X'Xu_1} - \lambda_1 \mathbf{u_1} = \mathbf{0}$$

Rearranging some terms we get:

$$\mathbf{X'Xu_1} = \lambda_1 \mathbf{u_1}$$

 CC BY-NC-SA 4.0

Lagrangian Multiplier Solution

What does this mean?

$$\mathbf{X'Xu_1} = \lambda_1 \mathbf{u_1}$$

It means that

- λ_1 is an eigenvalue of $\mathbf{X'X}$
- and $\mathbf{u_1}$ is the corresponding eigenvector

 CC BY-NC-SA 4.0

Finding 2nd PC

How to find the 2nd PC

In order to find the second principal component $Z_2 = \mathbf{X}\mathbf{u_2}$, we need to find $\mathbf{u_2}$ such that

$$\max_{\mathbf{u_2}} \; \mathbf{u_2'}\mathbf{X'}\mathbf{X}\mathbf{u_2}$$

subject to $\quad \|\mathbf{u_2}\| = 1 \quad$ and $\quad Z_1' Z_2 = 0$

Finding 2nd PC

Another eigenvalue-eigenvector pair

Applying the Lagrange multipliers, it can be shown that the desired $\mathbf{u_2}$ is such that

$$\mathbf{X}'\mathbf{X}\mathbf{u_2} = \lambda_2 \mathbf{u_2}$$

In other words

- λ_2 is an eigenvalue of $\mathbf{X}'\mathbf{X}$
- and $\mathbf{u_2}$ is the corresponding eigenvector

 CC BY-NC-SA 4.0

Finding all PCs

Diagonalization

All Principal Components can be found simultaneously by **diagonalizing** $\mathbf{X}'\mathbf{X}$

Eigenvalue Decomposition (EVD)

Diagonalizing a matrix is nothing more than obtaining its eigenvalue decomposition (a.k.a. spectral decomposition)

Variables Perspectives

 CC BY-NC-SA 4.0

Inertia for Variables

PCA model

A similar Inertia approach can be conceived for the variables. In this case, we look for vectors $\mathbf{v_j}$ such that the projections $\mathbf{X'v_j}$ have maximum inertia.

It turns out that:

The solution consists in **diagonalizing** $\mathbf{XX'}$

 CC BY-NC-SA 4.0

Algebraic Perspective

CC BY-NC-SA 4.0

Data Decomposition

Algebraically

PCA involves a **Singular Value Decomposition** (SVD) of the data matrix $\mathbf{X}$, that is:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}'$$

- $\mathbf{U}$ is orthonormal (i.e. $\mathbf{U}'\mathbf{U} = \mathbf{I}$)
- $\mathbf{D}$ is a diagonal matrix
- $\mathbf{V}$ is orthonormal (i.e. $\mathbf{V}'\mathbf{V} = \mathbf{I}$)

SVD Decomposition

PCA by SVD

The data matrix $\mathbf{X}$ is decomposed as a product of three simpler matrices $\mathbf{U}$, $\mathbf{D}$ and $\mathbf{V}$

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}'$$

- $\mathbf{U}_{n,k}$ (information about the basic structure)
- $\mathbf{D}_{k,k}$ (information about the scale)
- $\mathbf{V}_{p,k}$ (information about orientation or correlations)

CC BY-NC-SA 4.0

SVD and PCA

The relationship between **SVD** and **PCA** is:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}' = \mathbf{Z}\mathbf{P}'$$

where:

$\mathbf{Z} = \mathbf{U}\mathbf{D}$ (PCs or scores)

$\mathbf{P} = \mathbf{V}$ (Loadings)

Note that:

$\mathbf{W} = \mathbf{V}$ since $\mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{D}\mathbf{V}'\mathbf{V}$

i.e. $\mathbf{Z} = \mathbf{X}\mathbf{W} = \mathbf{X}\mathbf{V}' = \mathbf{U}\mathbf{D}$

 CC BY-NC-SA 4.0

PCA and Data Decomposition

Computation of all PCs

We can obtain as many PCs as the rank of $\mathbf{X}$ (i.e. $k = rank(\mathbf{X})$)

$$\mathbf{X}_{n,p} = \mathbf{Z}_{n,k}\mathbf{P'}_{k,p}$$

Keeping just a few PCs

But usually we will only retain just a few PCs (i.e. $m \ll p$)

$$\mathbf{X}_{n,p} \approx \mathbf{Z}_{n,m}\mathbf{P'}_{m,p} + Residual$$

(just a few PCs will *optimally* summarize the main structure of the data)

CC BY-NC-SA 4.0