# Principal components analysis

Reference: Introduction to Statistical Learning Chapter 10.1-10.2

## Outline

1. Introduction to PCA
2. Brief reminder of some linear algebra notation
3. PCA as a projection
4. PCA as an optimization problem
5. PCA as a regression on latent variables

## Introduction to PCA

The goal of PCA is to find low-dimensional summaries of high-dimensional data sets.

This is useful for compression, for denoising, for plotting, and for making sense of data sets that initially seem too complicated to understand.

It differs from clustering:

- Clustering assumes that each data point is a member of one, and only one, cluster. (Clusters are mutually exclusive.)
- PCA assumes that each data point is like a combination of multiple basic “ingredients.” (Ingredients are not mutually exclusive.)

## Think about recipes:

Nestle Toll House Chocolate-chip cookies: 280 grams flour, 150 grams white sugar, 165 grams brown sugar, 225 grams butter, 2 eggs, 0 grams water…

Mary Berry's Victoria sponge cake: 225 grams flour, 225 grams white sugar, 0 grams brown sugar, 225 grams butter, 4 eggs, 0 grams water…

seriouseats.com old fashioned flaky pie dough: 225 grams flour, 15 grams white sugar, 0 grams brown sugar, 225 grams butter, 0 eggs, 115 grams water…

## Think about recipes:

Each baked good is constructed by following a recipe: a combination of the same basic ingredients.

- Each data point $x_i$ is like a baked good.
- In PCA, the **principal components** are like the ingredients.

The amounts of each ingredient differ from one baked good to the next:

- E.g. 225g sugar for sponge cake versus 15 grams sugar for pie dough.
- In PCA, the **scores** are like the amounts of each ingredient in a given baked good.

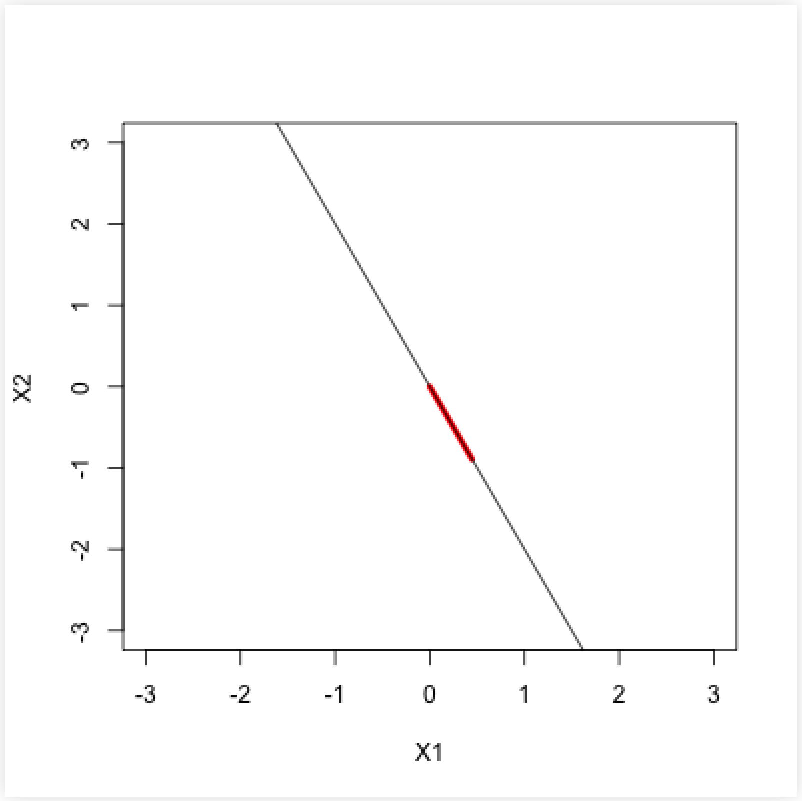
## Think about recipes:

Our goal is to reverse-engineer both the **ingredients** and the **amounts/recipes** from an observed set of “baked goods” (i.e. original data points).

## Some linear algebra reminders

Alas, PCA is less delicious than baking, and it uses more linear algebra. Say that $v \in \mathbb{R}^P$ is some vector. This defines a *subspace* of $\mathbb{R}^P$:

$$\mathcal{V} = \{z : z = \alpha_i v, \alpha_i \in \mathbb{R}\}$$



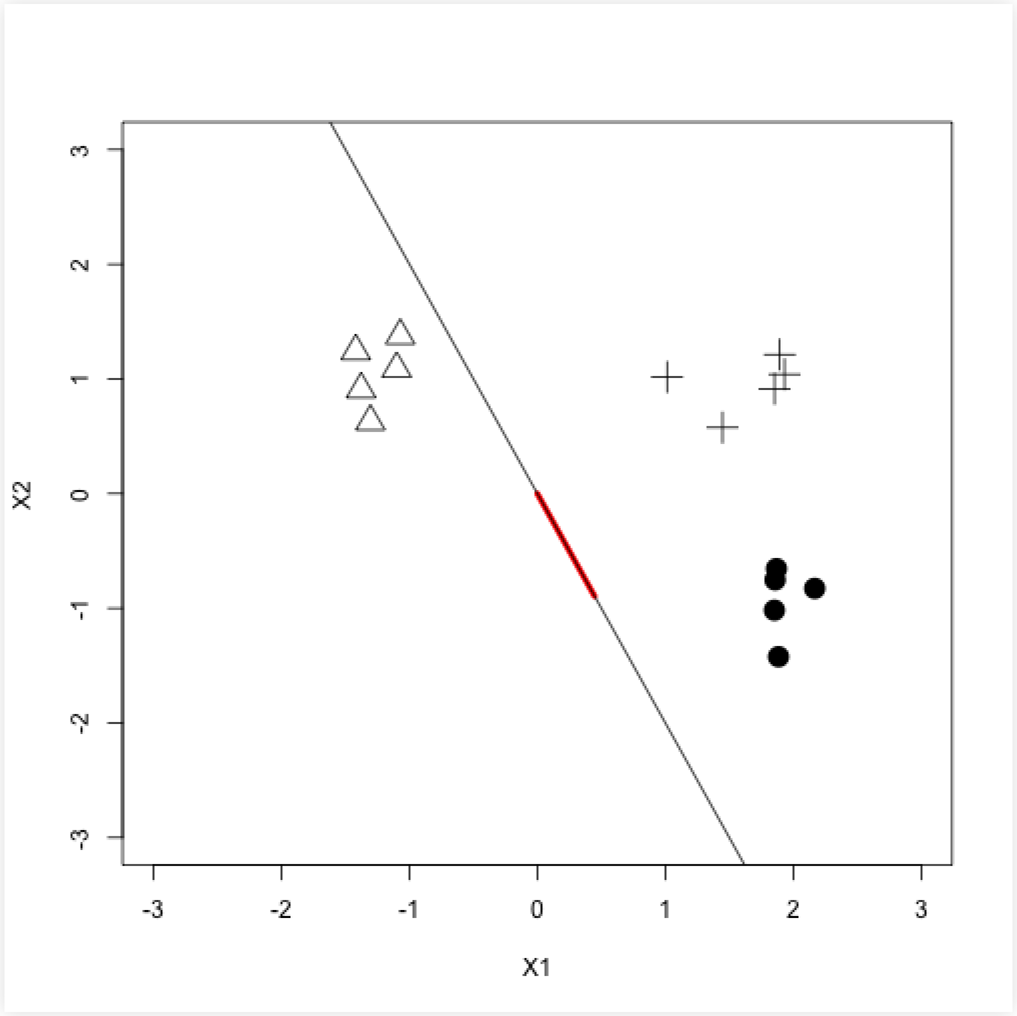
# Some linear algebra reminders

Now let $X$ be our usual $N \times P$ data matrix with rows $x_i^T$.

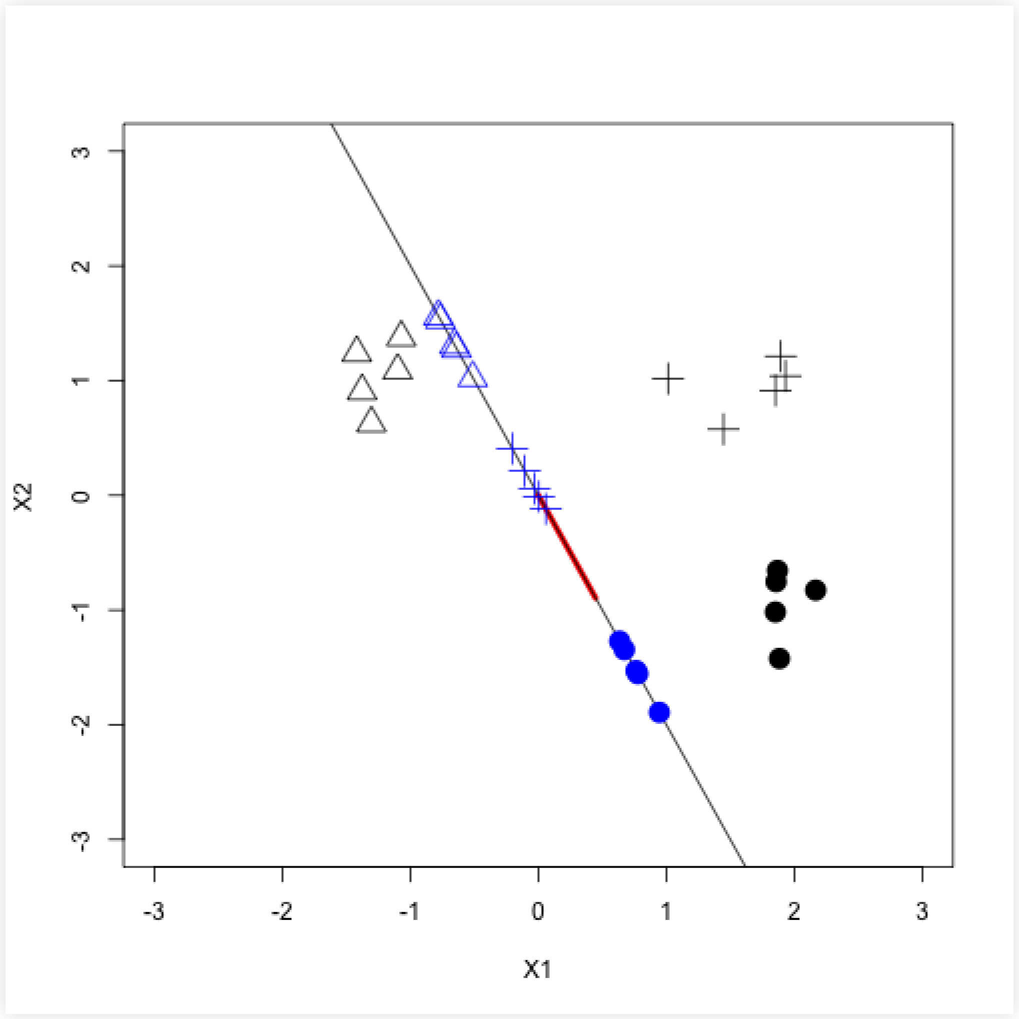
Suppose we *project* each $x_i^T$ in our data matrix onto the subspace $\mathcal{V}$.

- The scalar location along the vector $v$ is
$$\alpha_i = x_i \cdot v = x_i^T v$$
- The location of the projected point in $\mathcal{R}^P$ is
$$\hat{x}_i = \alpha_i v = (x_i \cdot v) v$$

## The original points



## With the projected points



## Key ideas

Key idea 1: projection = summary

- Each point's location along the subspace is a **one-number linear summary** of a $P$-dimensional data vector:
$$\alpha_i = x_i \cdot v = x_i^T v$$
- The goal of principal components analysis (PCA) is to find the "best" projection, i.e. the best linear summary of the data points.

Key idea 2: the "best summary" is the one that preserves as much of the **variance** in the original data points as possible. (Why?)

# PCA as an optimization problem

Given data points $x_1, \dots, x_N$, with each $x_i \in \mathbb{R}^P$, and a candidate vector $v_1$, the variance of the projected points is

$$\text{variance} = \frac{1}{n} \sum_{i=1}^{n} [\alpha_i - \bar{\alpha}]^2$$

where $\alpha_i x_i \cdot v_1$.

So we solve:

$$\underset{v_1 \in \mathbb{R}\ ,\ \|v\|_2 = 1}{\text{maximize}} \quad \sum_{i=1}^{n} \left[ x_i \cdot v_1 - \left( \frac{1}{n} \sum_{i=1}^{N} x_i \cdot v_1 \right) \right]^2$$

Note: we constrain $v_1$ to have length 1; otherwise we could blow up the variance of the projected points as large as we wanted to.

## PCA as an optimization problem

The solution $v_1$ to this optimization problem:

- is called the first principal component (synonyms: loading, rotation.)
- is the one-dimensional subspace capturing as much of the information in the original data matrix as possible.

The projected points $\alpha_i = x_i \cdot v$ are called the *scores* on the first principal component.

# PCA as regression on latent variables

We can think of the projected location of $x_i$ as an approximation to the original data point: $x_i \approx \hat{x}_i = \alpha_i v$.

Or to make the approximation error explicit:

$$\begin{aligned} x_{ij} &= \hat{x}_{ij} + e_i \\ &= \alpha_i v_j + e_i \end{aligned}$$

This is like a regression problem for the jth feature variable.

- The $\alpha_i$'s are like hidden (latent) features.
- $v_j$ is like a regression coefficient for observed variable $j$.

Thus PCA is like estimating $P$ regression coefficients $v_1 = (v_{11}, \ldots, v_{1P})$ all at once, where the feature variable is hidden.

## PCA as regression on latent variables

We can write the approximation for the whole matrix as follows:

$$
\begin{aligned}
X &\approx \begin{pmatrix} \alpha_1 v_{11} & \alpha_1 v_{12} & \cdots & \alpha_1 v_{1P} \\ \alpha_2 v_{11} & \alpha_2 v_{12} & \cdots & \alpha_2 v_{1P} \\ \vdots & \vdots & & \vdots \\ \alpha_N v_{11} & \alpha_N v_{12} & \cdots & \alpha_N v_{1P} \end{pmatrix} \\
&= \alpha v_1^T \quad \text{(outer product of } \alpha \text{ and } v_1\text{)}
\end{aligned}
$$

## PCA as regression on latent variables

And if we explicitly include the error term:

$$X = \alpha v_1^T + E$$

where $E$ is a residual matrix with entries

$$\begin{aligned} E_{ij} &= x_{ij} - \hat{x}_{ij} \\ &= x_{ij} - \alpha_i v_{1j} \end{aligned}$$

## Higher-order principal components

With this in place, we can now define principal components 2 and up!

- PC 2: run PCA on the residual matrix from PC 1.
- PC 3: run PCA on the residual matrix from PCs 1-2.
- …
- PC P: run PCA on the residual matrix from PCs 1-(P-1).

Thus principal component $M$ is defined recursively in terms of the fit from principal components 1 through $M - 1$.

## Higher-order principal components

Note: in practice we often stop with far fewer than $P$ principal components.

Let's see the examples in `congress109.R` and `NCI60.R`.