TEXAS McCombs

MASTER OF SCIENCE IN BUSINESS ANALYTICS

Principal component analysis

Outline

Introduction to PCA

Detour into linear algebra

Key ideas and representation as an optimization problem

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 in that ...

- 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

Think about recipes

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

Think about recipes

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 among baked goods:

→ For example, 225g sugar for sponge cake versus 15g 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).

Hence, another unsupervised task!

The output of PCA

Before: Raw survey data on a bunch of TV shows

	Entertaining	Engaged	Original	Confusing	Funny
30 Rock	4.2	3.7	4.0	2.2	3.7
Next Top Model	4.2	3.8	3.8	2.0	3.5
American Chopper	4.2	3.6	3.9	2.1	3.5
Bones	4.3	4.1	3.8	1.9	3.4
Close to Home	4.1	3.8	3.8	1.9	2.9
Cold Case	4.2	3.9	4.0	1.9	3.0

+ 15 other columns

The output of PCA

After: Survey data on a bunch of TV shows run through PCA

	PC1	PC2
30 Rock	-2.64	0.12
Next Top Model	-0.78	-0.19
American Chopper	-2.31	0.68
Bones	3.24	2.22
Close to Home	-0.63	2.86
Cold Case	1.62	2.88

from 20 raw variables to 2 summaries. PC1 and PC2 are like the “ingredients.” The numbers are the “amount” of each “ingredient.”

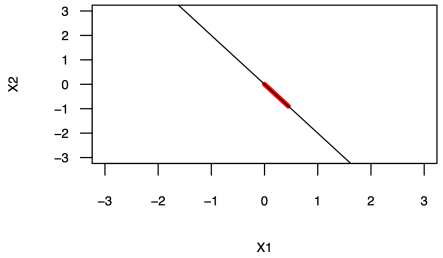
The trick to PCA is interpreting the summaries!

A brief detour into some linear algebra

Unfortunately, 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 $\mathbb{R}^P$:

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



A brief detour into some linear algebra

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

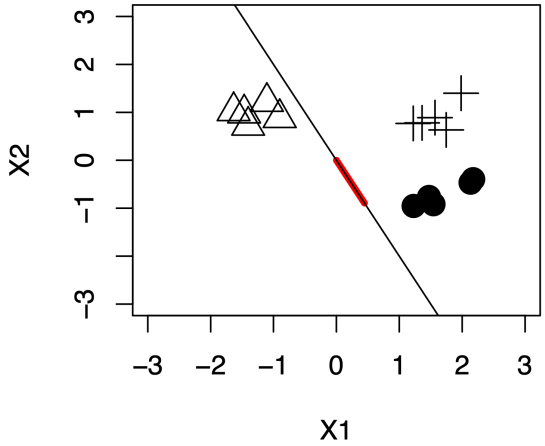
Suppose we **project** each x_i^T in our data matrix onto the subspace $\mathcal{V}$. The scalar location of this projected point in this subspace $\mathcal{V}$ is

$$\alpha_i = x_i \cdot v = x_i^T v$$

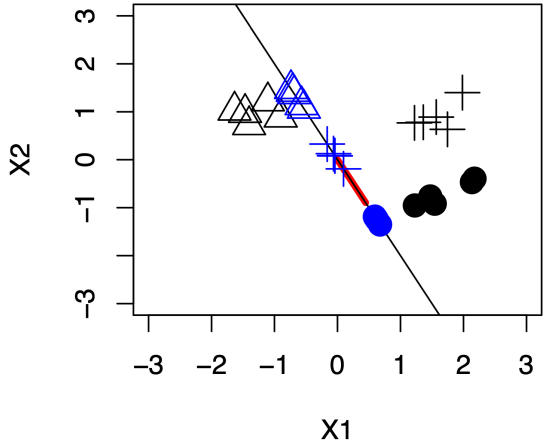
This is a one-number summary of our original point x_i. A different choice of v gives a different summary.

Let's see a picture.

The original points



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 ideas

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

- Intuition: we're already trying to crowd these P-dimensional points into 1-D. We should give them "room to breathe" by crowding them on top of each other as little as possible *within* that 1-D space!
- More variance in α_i means more "spread out" summaries in 1-D.
- And the more they're spread out, the more we have preserved differences between the projected points that were present in the original points. See `pca_intro.R`.

PCA as an optimization problem

Given data points $x_1, \ldots, x_N$, 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:

$\rightarrow$ is called the first principal component (synonyms: loading, rotation.)

$\rightarrow$ 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**.

Higher order principal components

We define principal components 2 and up by the directions in the data space not explained by the previous components

- 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 $M - 1$. In practice, we often stop with far fewer than P principal components.