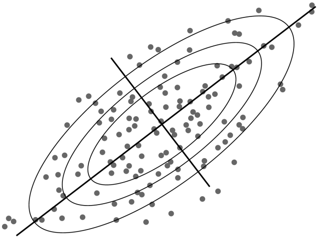
More About PCA

Foundations of Data Analysis

March 26, 2020

Review: Principal Component Analysis



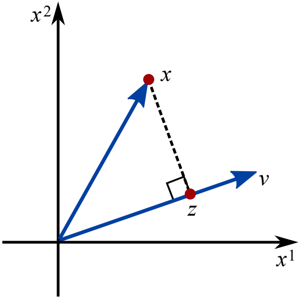
PCA is an eigenanalysis of the covariance matrix:

$$\Sigma = V \Lambda V^T$$

- Eigenvectors: $v_k = V_{\bullet k}$ are **principal components**
- Eigenvalues: λ_k are the **variance** of the data in the v_k direction

Maximizing Variance of Projected Data

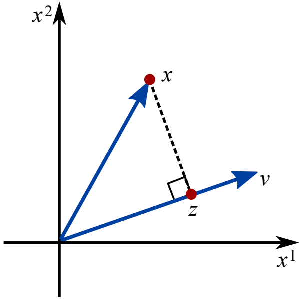
Fact: PCA finds dimensions that maximize variance



Given direction $v \in \mathbb{R}^d$, with $\|v\| = 1$,
project data point $x \in \mathbb{R}^d$ onto v:

$$z = \langle v, x \rangle$$

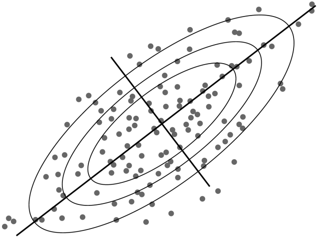
Maximizing Variance of Projected Data



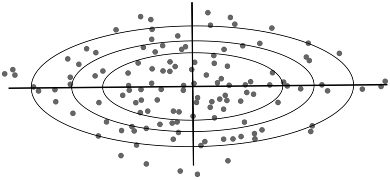
Given mean-centered data, x_i,
first principal component, v_1 maximizes variance:

$$v_1 = \arg\max_{\|v\|=1} \sum_{i=1}^{n} \langle v, x_i \rangle^2$$

PC's as Rotation



X



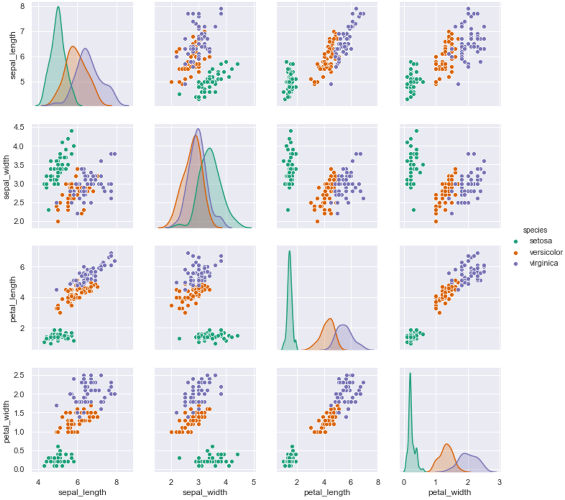
Z

The principal components matrix, V, acts as a rotation:

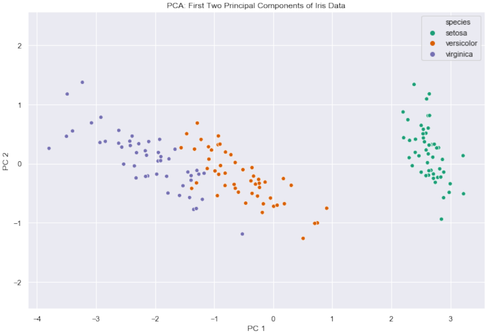
$$Z = XV$$

Columns of Z are new coordinates, called **loadings**

Example: Iris Data



Example: Iris Data PCA



	s. len.	s. wid.	p. len.	p. wid.
PC 1: $v_1 =$	-0.36	0.08	-0.86	-0.36
PC 2: $v_2 =$	0.66	0.73	-0.17	-0.08

These coefficients are called **weights**

Homework 4

- You may use a Python library for SVD and eigenanalysis (e.g., `numpy.linalg`)
- For part 1, you want to use SVD (e.g., `numpy.linalg.svd`)
- For part 2, you want to use eigenanalysis (e.g., `numpy.linalg.eigh`)
- You should not use a library function for PCA
- Finally, you may use a library (e.g., `matplotlib` or `seaborn`) for plotting hands