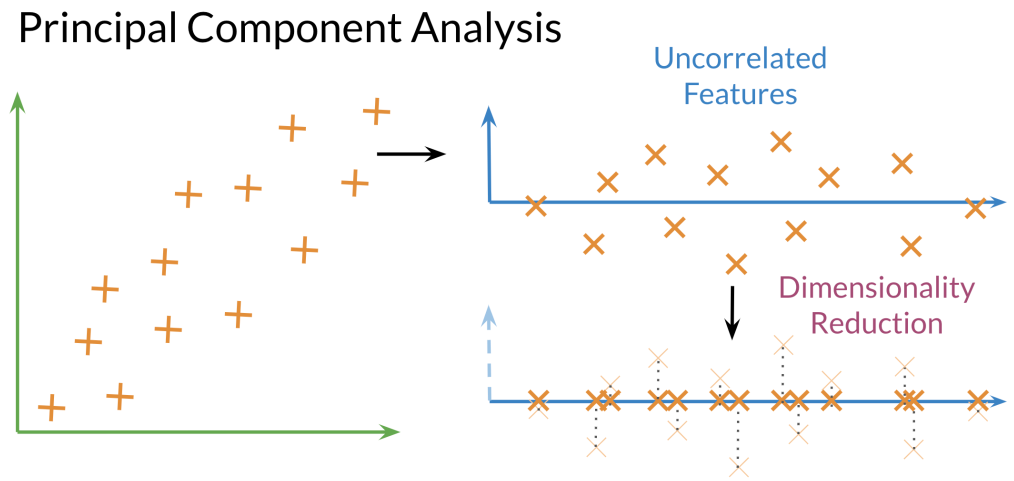
Hide menu

Lecture: Vector Space Models
- **Video:** Week Introduction 47 sec
- **Video:** Vector Space Models 2 min
- **Reading:** Vector Space Models 10 min
- **Video:** Word by Word and Word by Doc. 4 min
- **Reading:** Word by Word and Word by Doc. 10 min
- **Lab:** Linear algebra in Python with Numpy 1h
- **Video:** Euclidean Distance 3 min
- **Reading:** Euclidian Distance 10 min
- **Video:** Cosine Similarity: Intuition 2 min
- **Reading:** Cosine Similarity: Intuition 10 min
- **Video:** Cosine Similarity 3 min
- **Reading:** Cosine Similarity 10 min
- **Video:** Manipulating Words in Vector Spaces 3 min
- **Reading:** Manipulating Words in Vector Spaces 10 min
- **Lab:** Manipulating word embeddings 1h
- **Video:** Visualization and PCA 3 min
- **Reading:** Visualization and PCA 10 min
- **Video:** PCA Algorithm 3 min
- **Reading:** PCA algorithm 10 min
- **Lab:** Another explanation about PCA 1h
- **Reading:** The Rotation Matrix (Optional Reading) 10 min
- **Video:** Week Conclusion 46 sec

Lecture Notes (Optional)
Practice Quiz
Assignment: Vector Space Models

# PCA algorithm

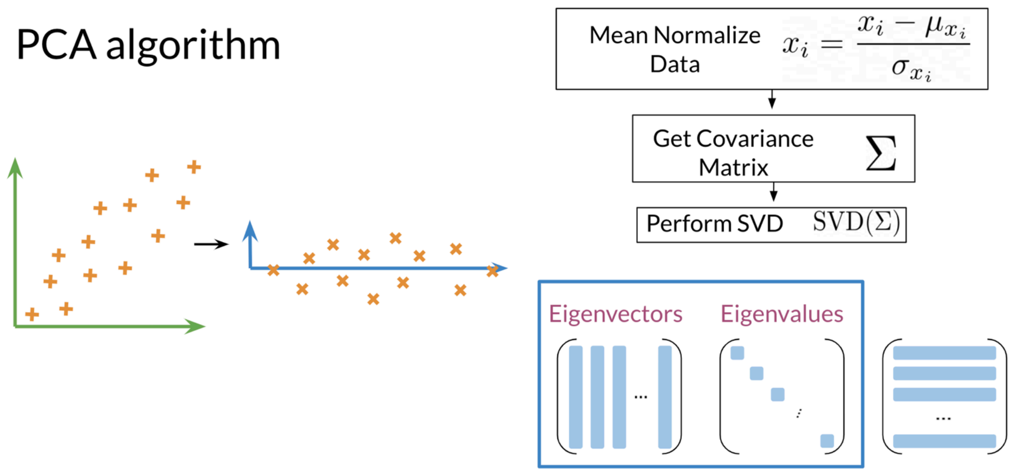
PCA is commonly used to reduce the dimension of your data. Intuitively the model collapses the data across principal components. You can think of the first principal component (in a 2D dataset) as the line where there is the most amount of variance. You can then collapse the data points on that line. Hence you went from 2D to 1D. You can generalize this intuition to several dimensions.



**Eigenvector**: the resulting vectors, also known as the uncorrelated features of your data

**Eigenvalue:** the amount of information retained by each new feature. You can think of it as the variance in the eigenvector.

Also each **eigenvalue** has a corresponding eigenvector. The eigenvalue tells you how much variance there is in the eigenvector. Here are the steps required to compute PCA:



**Steps to Compute PCA:**

- Mean normalize your data
- Compute the covariance matrix
- Compute SVD on your covariance matrix. This returns $[USV] = svd(\Sigma)$. The three matrices U, S, V are drawn above. U is labelled with eigenvectors, and S is labelled with eigenvalues.
- You can then use the first n columns of vector $U$, to get your new data by multiplying $XU[:, 0:n]$.