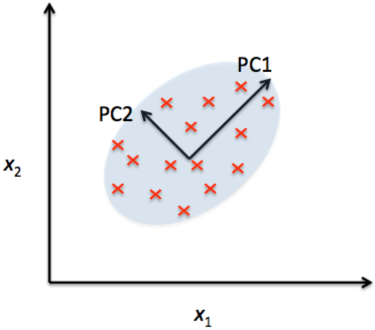
# Chapter 5

## Dimensionality Reduction

July 20, 2016

# Principal Component Analysis (PCA)

- Find the directions of maximum variance
- Project data onto the lower-dimensional space
- Original features: $x_1$ and $x_2$
- Principal components: **PC1** and **PC2**



# Mapping to a low-dimensional space

When we use PCA for dimensionality reduction, we construct a $d \times k$ transformation matrix $\mathbf{W}$. We then map a sample vector $\mathbf{x}$ onto a new $k$-dimensional feature subspace ($k << d$)

$$\mathbf{x} = [x_1, x_2, \ldots, x_j], \mathbf{x} \in \mathbb{R}^d$$

$$\downarrow \mathbf{xW}, \quad \mathbf{W} \in \mathbb{R}^{d \times k}$$

$$\mathbf{z} = [z_1, z_2, \ldots, z_k], \quad \mathbf{z} \in \mathbb{R}^k$$

# Principal components

- Transforming $d$-dimensional data to $k$ dimensions
- First principal component will have the largest variance
- Second principal component will have next largest variance
- And so on...
- PCA sensitive to data scaling, so need to standardize features

# Algorithm

1. Standardize the $d$-dimensional dataset.
2. Construct the covariance matrix.
3. Decompose the covariance matrix into its eigenvectors and eigenvalues.
4. Select $k$ eigenvectors that correspond to the $k$ largest eigenvalues, where $k$ is the dimensionality of the new feature subspace ($k \leq d$).
5. Construct a projection matrix $\mathbf{W}$ from the "top" $k$ eigenvectors.
6. Transform the $d$-dimensional input dataset $\mathbf{X}$ using the projection matrix $\mathbf{W}$ to obtain the new $k$-dimensional feature subspace.

# Variance-covariance matrix

- Symmetric $d \times d$ -dimensional matrix ($d$ - number of dimensions)
- Pairwise covariances between the different features
- Covariance between two features $\mathbf{x}_j$ and $\mathbf{x}_k$:

$$\sigma_{jk} = \frac{1}{n} \sum_{i=1}^{n} \left(x_j^{(i)} - \mu_j\right)\left(x_k^{(i)} - \mu_k\right)$$

Where $\mu_j$ and $\mu_k$ are the sample means of feature $j$ and $k$

# What is covariance?

- Measure of how much two random variables change together
- Positive covariance
  - Features increase together
  - Features decrease together
  - E.g. As a balloon is blown up it gets larger in all dimensions
- Negative covariance
  - Features vary in opposite directions
  - Large values of one variable correspond to small values of the other
  - E.g. if a sealed balloon is squashed in one dimension then it will expand in the other two
- The magnitude of the covariance is not easy to interpret
- The normalized version of covariance (*correlation coefficient*) indicates the strength of the linear relation.

# Variance-covariance matrix

- For three features, covariance matrix will look like this:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix}$$

- The eigenvectors of $\Sigma$ represent the principle components
- The corresponding eigenvalues represent their magnitude
  - Principle components: the directions of maximum variance
- E.g. Wine dataset (13 dimensions)
  - 13$x$13 covariance matrix
  - 13 eigenvectors
  - 13 eigenvalues

# Eigenpairs

- An Eigenvector $\mathbf{v}$ satisfies the condition:

$$\Sigma \mathbf{v} = \lambda \mathbf{v}$$

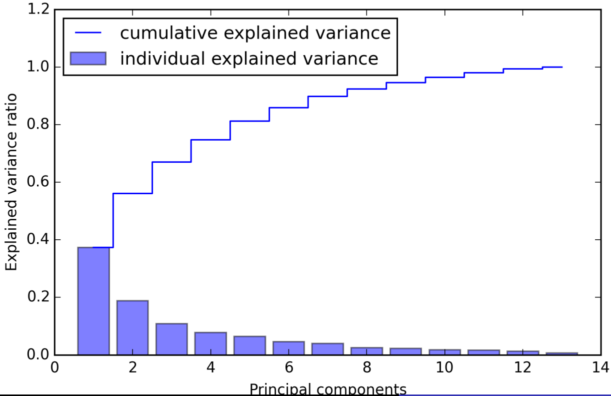
  Where $\lambda$ is the eigenvalue (scalar)
- NumPy has a function to compute eigenpairs
- We want to reduce the dimensionality
- So, we select a subset of $k$ most informative eigenvectors

# Variance explained ratio

- Variance explained ratio of an eigenvalue $\lambda_j$:

$$\frac{\lambda_j}{\sum_{j=1}^{d} \lambda_j}$$

- First two principal components explain about 60 percent of the variance in the data



# Feature transformation

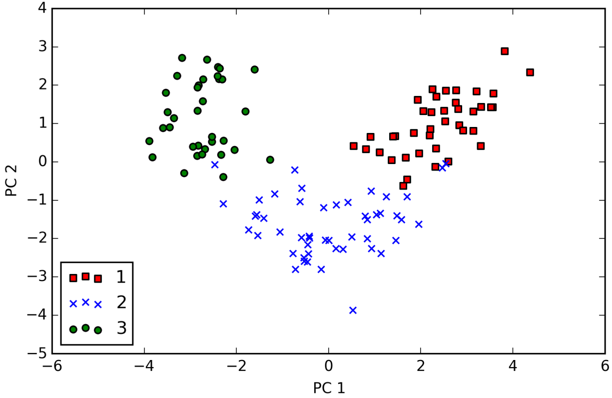
- We decomposed the covariance matrix into eigenpairs
- Now need to project to new space defined by principle component axes
- Construct a $13 \times 2$ projection matrix from top two eigenvectors
- Transform a sample $\mathbf{x}$ onto the PCA subspace obtaining $\mathbf{x}'$
- Which is a two-dimensional vector consisting of two new features:

$$\mathbf{x}' = \mathbf{x}\mathbf{W}$$

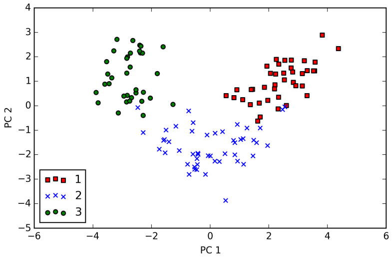
- Transform entire Wine dataset $(124 \times 13)$

$$\mathbf{X}' = \mathbf{X}\mathbf{W}$$

# Visualize *Wine* dataset in two dimensions



# Visualize *Wine* dataset in two dimensions




- Can now visualize a 13-dimensional dataset
- Data more spread along first principal component, which explained 40 percent of the variance
- A linear classifier should be able to do a good job separating the classes
- Keep in mind that PCA is an *unsupervised* algorithm