Principal Component Analysis (PCA)

Chris Reger (Thanks to F. Burkholder, A. Richards, R. Henning)

Objectives

galvanize

- PCA -> **P**rincipal **C**omponent **A**nalysis
- Define PCA conceptually
- Describe motivation(s) for PCA in machine learning
- Be able to apply PCA in numpy
    - Understand and apply PCA's mathematical definition
- Discuss methodology for picking the number of principal components
- Perform PCA in sklearn

Principal **C**omponent **A**nalysis -> PCA

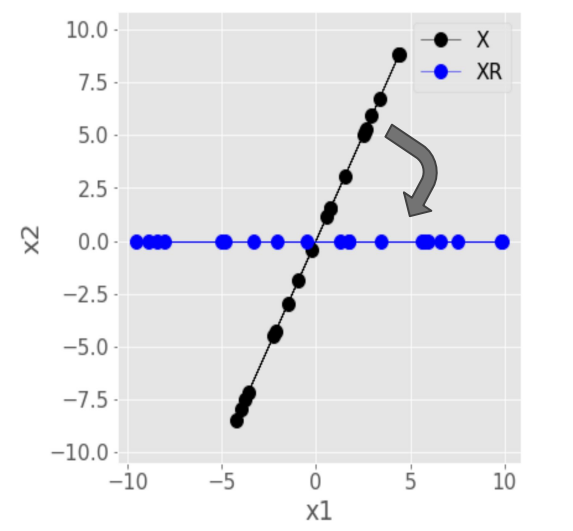
galvanize

Definition #1 (of 3):

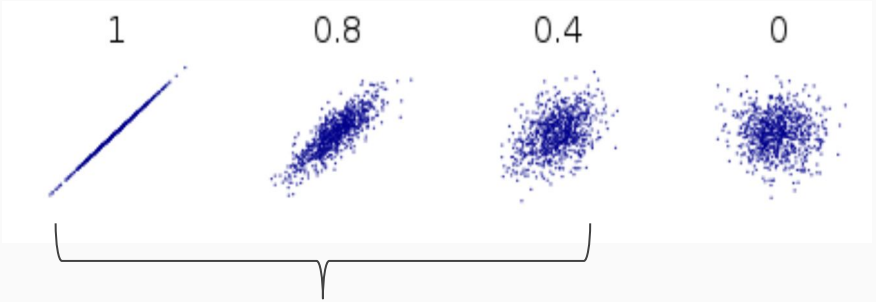
PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated features into … linearly uncorrelated features called principal components.

adapted from Wikipedia

Principal Component Analysis -> PCA



Example of an orthogonal transformation



Correlated features, candidates for PCA

Principal **C**omponent **A**nalysis -> PCA

galvanize

Definition #2 (of 3):

In PCA, the dataset is transformed from its original coordinate system to a new coordinate system. The new coordinate system is chosen by the data itself. The first new axis (a.k.a. principal component) is chosen in the direction of the most variance of the data. The second axis is orthogonal to the first axis and is in the direction of the next most variance in the data.

More and more orthogonal axes (PCs) can be added to describe all the variance in the data.

Usually the majority of the variance of the data is contained in the first few principal components. So the rest of the axes can be ignored, reducing dimensionality in the data.

adapted from Machine Learning in Action

Principal **C**omponent **A**nalysis -> PCA

galvanize

Definition #3 (of 3):

When faced with a large set of correlated variables, principal components allow us to summarize this set with a smaller number of representative variables that collectively explain most of the variability in the original set.

Principal component directions are … directions in feature space along which the data are highly variable.

PCA refers to the process by which principal components are computed, and the use of these components in understanding the data.

adapted from Introduction to Statistical Learning

PCA in machine learning

galvanize

PCA defines orthogonal axes of variance that can be used to describe the variance in the data.

This does two helpful things at once:

- Dimensionality reduction
- Remove collinearity of features

With fewer dimensions you can:

- Better visualize your data
- Make computations for your algorithms easier
- Identify structure for supervised learning

Disadvantages to PCA:

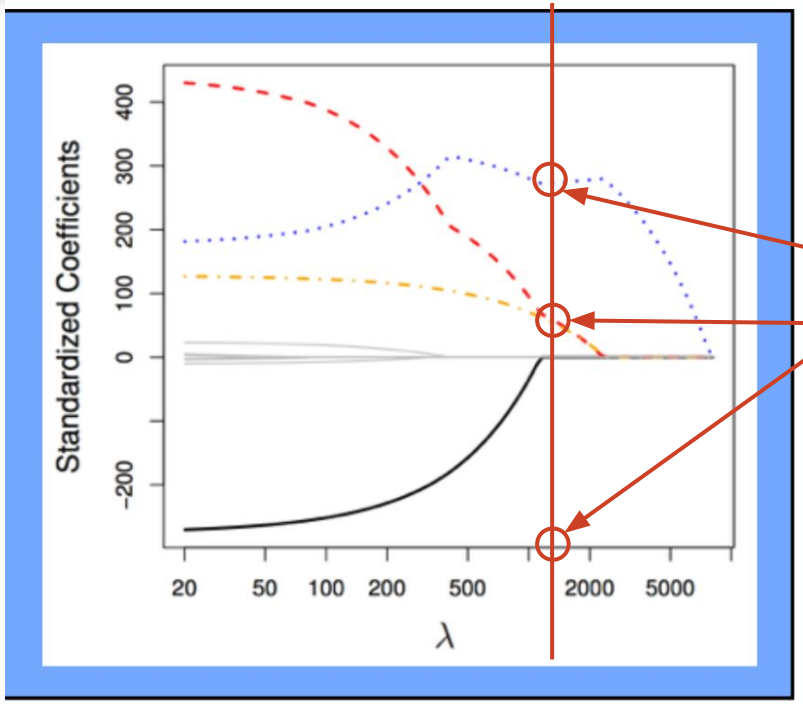
- Principal components are linear combinations of the features (less interpretable)

Other dimensionality reduction techniques

galvanize

- LASSO regularization
- “Relaxed” LASSO (next slide)
- sklearn’s Feature Selection page
  - `VarianceThreshold`
  - `SelectKBest`
  - `Recursive Feature Elimination (RFE)`
  - `SelectFromModel`
- Decision Tree Based
  - .feature_importances_

Relaxed LASSO



1. Use cross-validation and LASSO regression; find the best value for lambda.

2. Keep only the features with non-zero coefficients.

3. Re-fit using ordinary least squares (OLS) regression. (I.e. Re-fit using no regularization!)

Source: Meinshausen, 2006

More about principal components

galvanize

$$\text{features } X_1, X_2, \ldots, X_p$$

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p$$

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

Principal component Z_1 is a normalized linear combination of the features in X.

$\phi_{11}, \ldots, \phi_{p1}$ are the loadings of Z_1 on the features in X.

How-to-PCA

galvanize

- Accomplished with one of two methods:
  - Eigen-decomposition of the X covariance or correlation matrix (overview)
  - Singular value decomposition (covered later)

Eigenvectors

galvanize

See Visualization of Eigenvectors
Another Visualization

High-Level: How-to-PCA

galvanize

Steps (eigen-decomposition):

- Create "design matrix" $\boldsymbol{X}$ by mean-centering and (usually) by dividing by the standard deviation. (Yes, this is just standardizing the columns.)
- Compute the covariance (if only mean-centered) or correlation matrix: $\frac{1}{N}X^TX$
- Find the eigenvectors ($\mathbf{v}$) and eigenvalues($\boldsymbol{\lambda}$) of the covariance/correlation matrix ($\mathbf{A}$):

$$\mathbf{A} = \frac{1}{N}X^TX \qquad \mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

Example how to do this by hand.

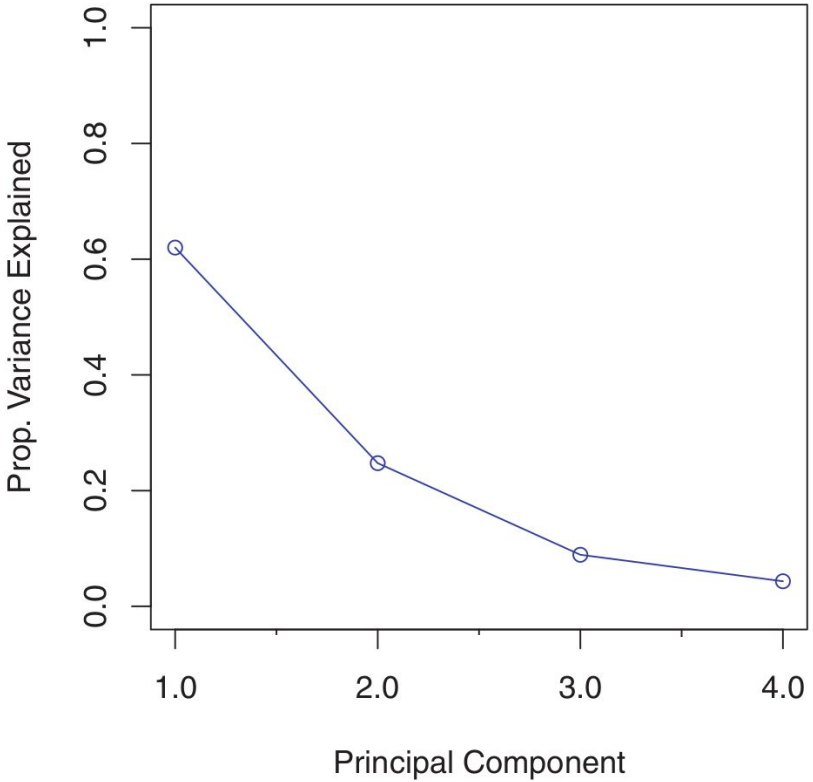
- The **eigenvectors are the principal components**.
- Further Reading

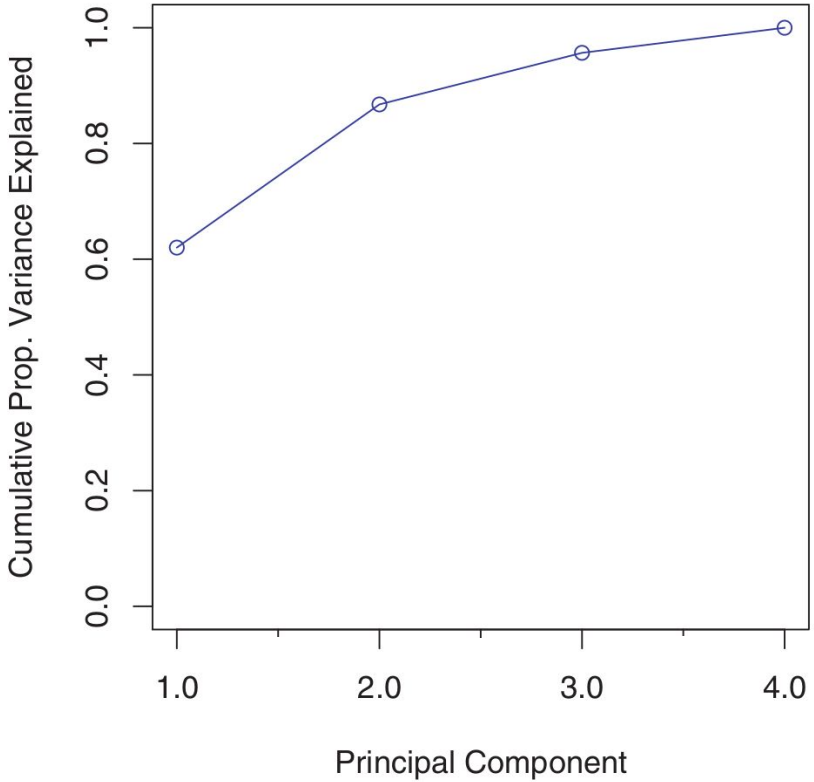
Picking the number of principal components

galvanize

- An ***n*** x ***p*** matrix (***n*** rows, ***p*** columns) has min(***n-1***, ***p***) principal components, but we're usually not interested in all of them.
- Just want the "first few" that "capture most" of the variance.
- "Capture most" is usually about 90-95%.
- The eigenvalues are measures of variance.
- The cumulative sum of the eigenvalues up to and including a certain principal component, divided by the sum of all the eigenvalues, **is** the explained variance.
- A scree plot is often used to visualize this.

Scree plots





See Scree demo in PCA_notebook.ipynb

Source: ISLR

Breakout

galvanize

- Groups - come back ready to explain to a non-technical person:
  - What does PCA stand for?
  - What is it used for?
  - What is a principal component?
  - What are Eigenvectors? Eigenvalues?
  - What is eigendecomposition?

Objectives

galvanize

- PCA ->
- Define PCA conceptually
- Describe motivation for PCA in machine learning
- Be able to apply PCA in numpy
  - Understand and apply PCA's mathematical definition
  - Review some statistical concepts
- Provide a methodology for picking the number of principal components
- Perform PCA in sklearn