Business Analytics

Session 5b. Principal Component Analysis and Feature Selection

Renyu (Philip) Zhang

New York University Shanghai

Spring 2019

Dimensionality Reduction

- We have access to very rich data set of high dimensinality.
  - Requires intensive computational resource.
  - May contain redundant information.

- Dimensionality Reduction: Build low-dimensional representations for high-dimensional data.
  - Principal Component Analysis (PCA)
  - Manifold Learning

- Examples:
  - How to represent the ratings I gave to every product I've purchased?

    A (sparse) vector including all products vs. A low-dimensional vector describing my preferences.
  - How to represent the complete text of a document?

    A vector counting all words in the text vs. A low-dimensional vector describing the topics in the document.
  - How to represent connections in a social network?

    Adjacency matrix vs. Each node or user in terms of the communities they belong to.

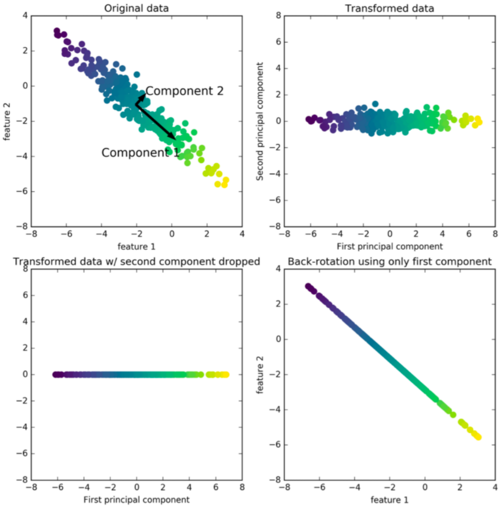
Principal Component Analysis

Breast Cancer Diagnosis

- Features computed from a digilized image of a fine needle aspirate of a breast mass.
  - Characteristics of the cell nuclei present in the image.
  - 30-dimensinal (radius, texture, smoothness, compactness, . . .)
- Questions:
  - Which dimensions are highly correlated (and how)?
  - Which dimensions could we "throw away" without losing much information?
  - How can we find which dimensions can be thrown away automatically?
  - In other words, how could we come up with a "compressed representation" of the cell nuclei's 30-d information into (say) 5-d?
- Pricipal Component Analysis
  - Select a few important features.
  - Compress the data by ignoring components which aren't meaningful.

PCA Objective

$$\min_{X', \text{rank}(X')=r<p} ||X - X'||$$

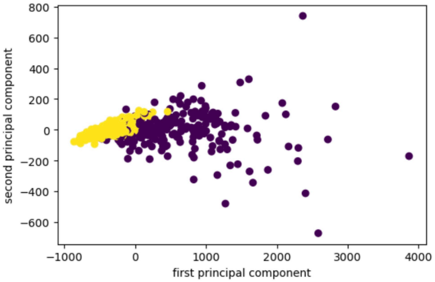


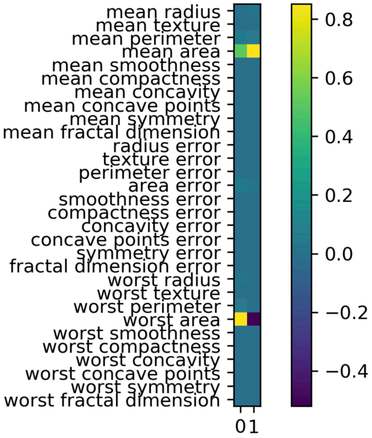
Find Important Dimensions

- Rotate the data such that
  - Most of the variance is along $\vec{x}_1$;
  - Most of the leftover variance (not explained by $\vec{x}_1$) is along $\vec{x}_2$;
  - Most of the leftover variance (not explained by $\vec{x}_1, \vec{x}_2$) is along $\vec{x}_3$;
  - ...
  - Leftover variance (not explained by $\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_{r-1}$) is along $\vec{x}_r$.
- Maximize the “randomness” that gets preserved.
- Each original covariate i is represented as

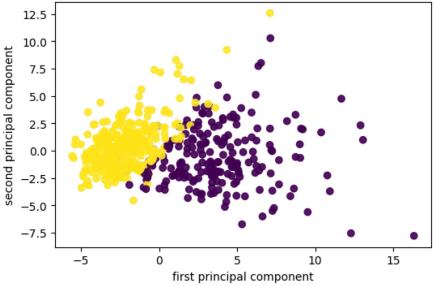
$$k_{i,1}\vec{x}_1 + k_{i,2}\vec{x}_2 + \ldots + k_{i,r}\vec{x}_r$$

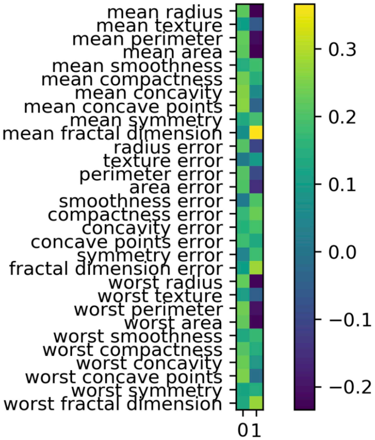
PCA for Breast Cancer Data





PCA for Breast Cancer with Scaling





PCA for Classification using Regularized Logistic Regression

- Original data: Overall Acuracy = 0.937
- PCA with 2 components: Overall Acuracy = 0.923.
- PCA with 2 components: Overall Acuracy = 0.958.

Feature Selection

Why Select Features?

- Avoid overfitting.
- Faster training and prediction.
- Less storage for model and dataset.
- More interpretable model.

Unsupervised Feature Selection

- May discard important information.
- Varianced-based: Remove features of 0 variance or very few values.
- Covariance: Remove correlated features.
- PCA: Remove linear spaces.

Supervised Feature Selection

- Univariate statistics: $p-$value, $F-$value.
  - f_regression, f_classif, chi2 in scikit-learn.

- Multual information between outcome and covariates.
  - mutual_info_regression, mutual_info_classif in scikit-learn.

Model-Based Feature Selection

- Get best fit for a particular model.
- Ideally: Exhaustive search over all possible combinations.
  - Exaustive search is infeasible.
- Use heuristics in practice.
  - Select the features with "highest importance" (largest coefficients in linear models, closest to root in trees).
  - Iterative model-based selection:

    Fit model, find least important feature, remove, iterate.

    Or: Start with single feature, find most important feature, add, iterate.

Recursive Feature Elimination

- Train the model with all features.
- Iteratively remove features and re-train the model based on feature “importance”.
- RFE function in sklearn.feature_selection

Wrapping Method

- Flexible enough to be applied for ANY model!
- Shrink/grow feature set by greedy search.
  - Called forward or backward selection.
- Run CV/train-validate split per each feature.
- Computational complexity: $p(p+1)/2$.
- Implemented in mlxtend.
  - pip install mlxtend

What We Do Not Have Time to Cover

- Data Visualization
  - See NYU Classes, Python&R Resources for more references.
- Support Vector Machines
  - A simple yet powerful classifier.
  - More generally, kernel methods (transform data into another dimension so that a clear division of the data exists).
- Bagging (aka bootstrap aggregating).
  - Generalize the idea of random forests.
- Boosting.
  - Build a stronger model with multiple weak models.
- Gaussian Mixture Models.
  - Soft-boundary version of $k-$means.
- Manifold learning.
  - Learn low-dimensional structure from high-dimensinal data.
- Natural language Processing.
  - Handling and predictive analytics with text data.
- Neural networks.
  - Powerful but less interpretable.
- Any many more other interesting stuffs...

Homework

- Finish Homework 5 (NO need to submit it).
- A bonus assignment (5% extra credits).
  - Due at 10:00pm on May 20.
- Read the assigned reading.