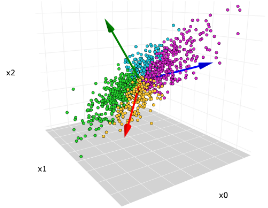
Dimmensionality Reduction and Cross Validation

Dana Golden, Lilia Maliar



Data Science and Machine Learning - **November 30, 2024**

Presentation Outline

Value of Dimensional Reduction

- Dimensionality reduction reduces data to its dimensions of highest
- It can allow datasets with thousands of variables

Singular Value Decomposition

- Factorization of matrix into three components
- Generalizes diagonalization to non-square and singular matrices

SVD Visualized

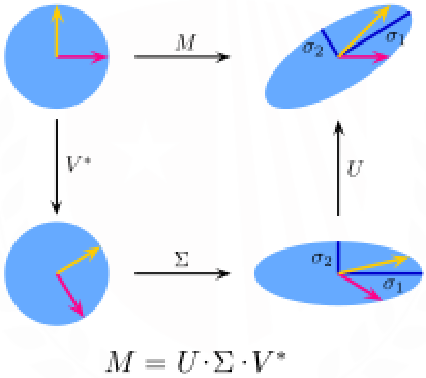


Figure 1: SVD Visualized

SVD Example

$$A = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix} \quad (1)$$

- Start by finding eigenvalues of AA^T

$$AA^T = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix} * \begin{bmatrix} 3 & 2 \\ 2 & 3 \\ 2 & -2 \end{bmatrix} = \begin{bmatrix} 17 & 8 \\ 8 & 17 \end{bmatrix} \quad (2)$$

SVD Example: Characteristic Equation

$$AA^T - \lambda I = 0 \tag{3}$$

SVD Example: Characteristic Equation

$$AA^T - \lambda I = 0 \tag{3}$$

$$\lambda^2 - 34\lambda + 225 = 0 \tag{4}$$

SVD Example: Characteristic Equation

$$AA^T - \lambda I = 0 \tag{3}$$

$$\lambda^2 - 34\lambda + 225 = 0 \tag{4}$$

$$(\lambda - 25)(\lambda - 9) = 0 \tag{5}$$

- $\sigma_1 = \sqrt{25}$, $\sigma_2 = \sqrt{9}$
- 3, 5, and 0 are our singular values σ_i

Finding V

- Find unit-length vector in kernel of matrix $A^T A - \sigma_i^2 I$

$$A^T A - 25I = \begin{bmatrix} -12 & 12 & 2 \\ 12 & -12 & -2 \\ 2 & -2 & -17 \end{bmatrix} \tag{6}$$

- This row reduces to:

Finding V

- Find unit-length vector in kernel of matrix $A^TA - \sigma_i^2 I$

$$A^TA - 25I = \begin{bmatrix} -12 & 12 & 2 \\ 12 & -12 & -2 \\ 2 & -2 & -17 \end{bmatrix} \tag{6}$$

- This row reduces to:

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{7}$$

- The unit-length vector in the kernel is:

$$v_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} \tag{8}$$

Finding V

- Find unit-length vector in kernel of matrix $A^T A - \sigma_i^2 I$

$$A^T A - 9I = \begin{bmatrix} 4 & 12 & 2 \\ 12 & 4 & -2 \\ 2 & -2 & -1 \end{bmatrix} \tag{9}$$

- This row reduces to:

$$\begin{bmatrix} 1 & 0 & \frac{-1}{4} \\ 0 & 1 & \frac{1}{4} \\ 0 & 0 & 0 \end{bmatrix} \tag{10}$$

- The unit-length vector in the kernel is:

$$v_2 = \begin{bmatrix} \frac{1}{\sqrt{18}} \\ \frac{-1}{\sqrt{18}} \\ \frac{4}{\sqrt{18}} \end{bmatrix} \tag{11}$$

Finding V

- Final vector can be found by computing the kernel of $A^T A$ or by finding a unit-length vector perpendicular to v_1 and with transpose perpendicular to V_2

$$v_3 = \begin{bmatrix} \frac{2}{\sqrt{3}} \\ \frac{-2}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix} \tag{12}$$

Finding U

$$A = U\Sigma V^T = U \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix} * \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{18}} & \frac{-1}{\sqrt{18}} & \frac{4}{\sqrt{18}} \\ \frac{2}{3} & \frac{-2}{3} & \frac{-1}{2} \end{bmatrix} \quad (13)$$

- $\sigma u_i = Av_i$ or $u_i = \frac{1}{\sigma} Av_i$

$$A = U\Sigma V^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} * \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix} * \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{18}} & 3 & 0 \end{bmatrix} \quad (14)$$

Principal Component Analysis Overview

- PCA is the most commonly used dimensionality reduction technique
- PCA is used to reduce the data to a combination of variables representing maximum variance

PCA Steps

1. Standardize Data
2. Compute Covariance Matrix
3. SVD or eigendecomposition
4. Select top Eigenvectors

Standardize Data

$$CenteredData = \frac{OriginalData - colMean}{ColstandardDeviation} \tag{15}$$

Compute Covariance Matrix

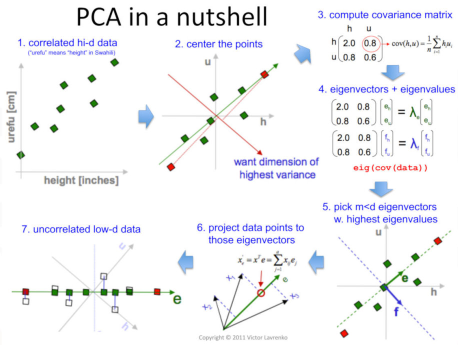
$$\text{Cov}(A) = \begin{bmatrix} \frac{\sum (x_i - \bar{X})(x_i - \bar{X})}{N} & \frac{\sum (x_i - \bar{X})(y_i - \bar{Y})}{N} \\ \frac{\sum (x_i - \bar{X})(y_i - \bar{Y})}{N} & \frac{\sum (y_i - \bar{Y})(y_i - \bar{Y})}{N} \end{bmatrix}$$

$$= \begin{bmatrix} \text{Cov}(X,X) & \text{Cov}(Y,X) \\ \text{Cov}(X,Y) & \text{Cov}(Y,Y) \end{bmatrix}$$

Select top Eigenvectors of covariance matrix

- Perform Eigenvector decomposition
- Order eigenvectors by eigenvalues
- Highest eigenvalues correspond to eigenvectors of "principal components" explaining most variance
- If not a square matrix, do SVD

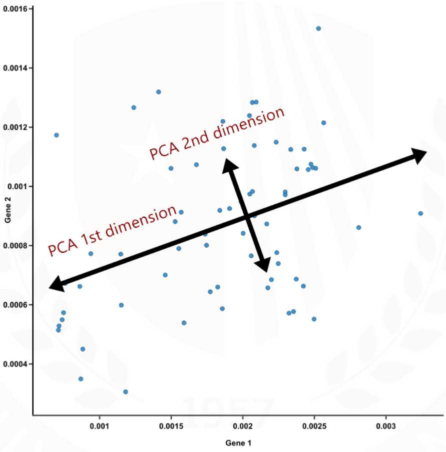
PCA Example


PCA in a nutshell
1. correlated hi-d data
("urefu" means "height" in Swahili)
urefu [cm]
height [inches]
2. center the points
u
h
want dimension of
highest variance
3. compute covariance matrix
4. eigenvectors + eigenvalues
eig(cov(data))
5. pick m<d eigenvectors
w. highest eigenvalues
e
f
6. project data points to
those eigenvectors
7. uncorrelated low-d data
Copyright © 2011 Victor Lavrenko

PCA Regression

- PCA regression often used to avoid multicolinearity problem in regression
- PCA is also useful in situations with high-dimensional covariates
- If using to forecast, make sure to reconvert to standard coordinates

PCA Visualization



LDA

- PCA is unsupervised while LDA is supervised
- PCA attempts to find principal components that maximize variation while LDA finds variables that maximize separability between groups
- LDA common in topic modelling

LDA Visualized

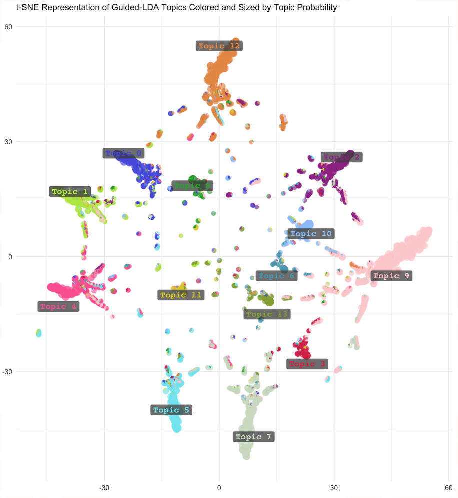


Figure 2: LDA Visualization

Bias-variance Tradeoff

$$MSE = bias^2 + variance + baselineError^2 \tag{16}$$

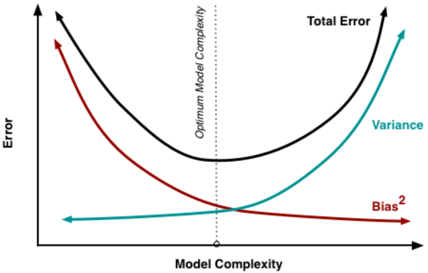


Figure 3: Bias-Variance Tradeoff

Bullseye Picture

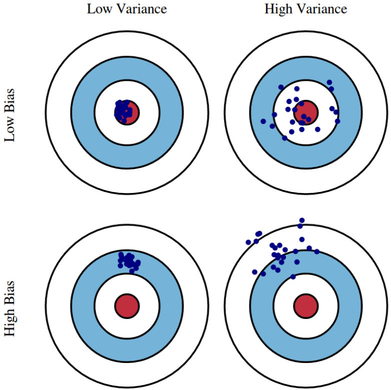


Figure 4: Precision vs. Accuracy

Overfitting vs. Underfitting

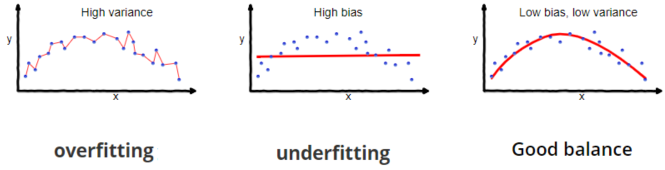


Figure 5: Overfitting vs. Underfitting

Splitting Data

- Data can be split into two or three datasets: training, testing, and cross-validation
- Data are split to make sure models fit out-of-sample data correctly
- Cross-validation dataset is used for parameter

Cross-validation

- How did you tune k?

Cross-validation

- How did you tune k?
- Cross-validation!

Cross-validation

- How did you tune k?
- Cross-validation!
- Cross-validation allows you to choose model parameters by testing the model on data other than the test set with a range of different parameters
- Can use separate dataset or split training set

K-Fold Cross-validation

- Divide the dataset into k folds
- Train with parameter on k-1 data
- Test for parameter on last data
- Repeat and take average error
- Use parameter with minimal error

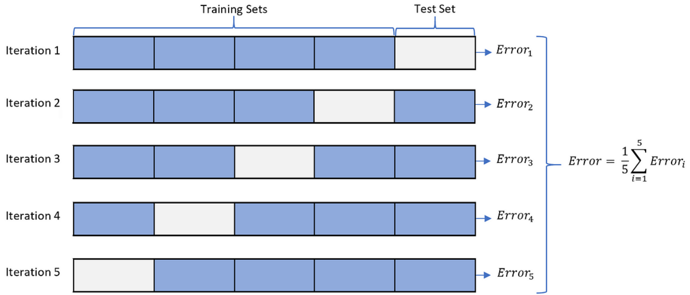


Figure 6: K-fold Cross validation

Leave-one-out Cross-validation

- Leave one out is a special form of K-fold cross validation in which one observation is used as the cross-validation set over all observations, and the average error is cross-validation error
- LOOCV is computationally expensive but good with few observations

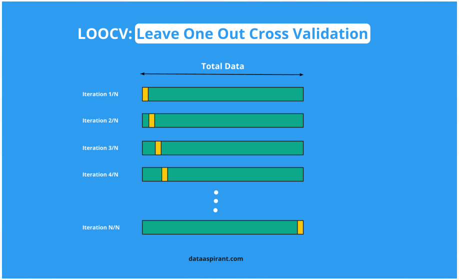


Figure 7: Leave-one-out Cross-validation

Stratified K-Fold CV

- Used most commonly for classification tasks
- Attempts to use stratified random sampling to match the proportions of observations in the training data
- Used to prevent bad batches of folds from messing up training error

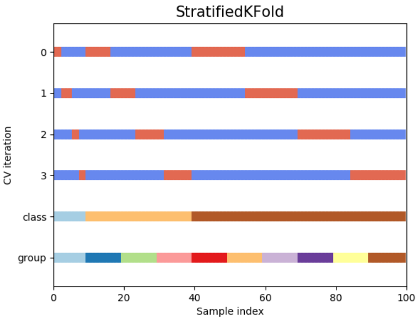


Figure 8: Stratified K-fold Cross-validation

Thank You So Much!

List of References