Felix Juefei Xu

Pattern Recognition Theory

Recitation 4: PCA- Recap

Topics To Be Covered

- PCA Basics
  - Motivation
- PCA Derivation
  - Eigen-decompostion on $\mathbf{X}\mathbf{X}^T$
  - Singular value decomposition (SVD) on $\mathbf{X}$
- Application and Caveat
- Further Reading
  - More component analysis (CA) methods

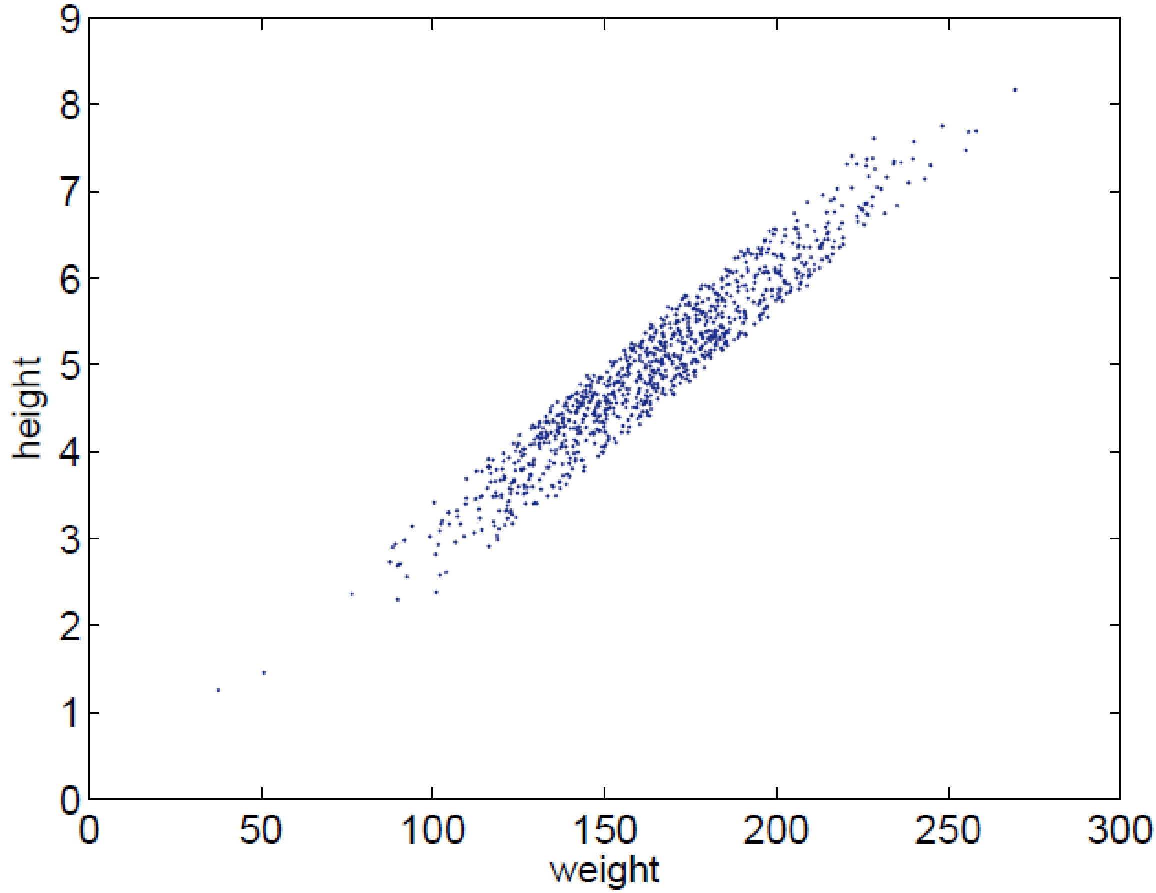
The PCA Basics

- PCA is a linear projection operation that maps a variable of interest to a new coordinate system where the axes represent maximal variability.

- Input data matrix **X** (D by N), D is the dimension, and N is the number of samples. Usually D >> N
- Output **Y** (D' by N), D' <= D
- $\mathbf{Y} = \mathbf{P}^T\mathbf{X}$
  - Where **P** is the projection matrix (D by D') of which each column is a principal component (PC)

Motivation

- Remove redundancy
  - If some dimensions are highly correlated (or perfectly correlated, a line), we could discard some dimensions while still capturing the full distribution.



Derivation

- Most commonly used technique is to apply eigen-decomposition on the covariance matrix as shown in class.
- Covariance matrix of **X** is: $\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \mu)(\mathbf{x}_n - \mu)^T$
- Optimization framework:

$$\mathbf{p}_1 \leftarrow \max F = \mathbf{p}_1^T \mathbf{C} \mathbf{p}_1 + \lambda_1 (1 - \mathbf{p}_1^T \mathbf{p}_1)$$

- The unit norm constraint ensures that the projection is purely rotational without any scaling.
- Eigen-decomposition: $\mathbf{C}\mathbf{p}_1 \;=\; \lambda_1 \mathbf{p}_1$

Derivation

- $\mathbf{P}_2, \ldots, \mathbf{P}_{D'}$ can be found by repeating the aforementioned process.
- A good exercise, manually eigen-decompose a simple 2-by-2 matrix.
- MATLAB: eigs(C,D') / eig(C)

- Full projection matrix **P** (D'=D)
  - Covariance matrix is diagonalized as follows:

$$\mathbf{C} = \mathbf{P}\Lambda\mathbf{P}^T$$

- The i-th eigenvalue indicates the variance explained by projecting the data onto the i-th PC

Detour - SVD

- Singular value decomposition of an m x n real or complex matrix $\mathbf{M}$ is a factorization of the form:

$$\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$$

- $\mathbf{U}$ is a *m x m* real or complex unitary matrix.
- $\boldsymbol{\Sigma}$ is a *m x n* rectangular diagonal matrix with nonnegative real numbers on the diagonal.
- $\mathbf{V}^*$ is the conjugate transpose of $\mathbf{V}$, which is an *n x n* real or complex unitary matrix.
- The left-singular vectors of M are eigenvectors of MM*.
- The right-singular vectors of M are eigenvectors of M*M.

Detour – Truncated SVD

- Approximating $\mathbf{M}$ using $\hat{\mathbf{M}}$ by considering only t largest singular values. The rest of the matrix is discarded.

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^*$$
$$\hat{\mathbf{M}} = \mathbf{U}_t\Sigma_t\mathbf{V}_t^*$$

- Only the *t* column vectors of U and *t* row vectors of V* corresponding to the *t* largest singular values $\boldsymbol{\Sigma}_t$.
- Of course, the truncated SVD is no longer an exact decomposition of the original matrix $\mathbf{M}$, $\hat{\mathbf{M}}$ is a low rank approximation.
- $\mathbf{U}_t$ is thus <u>*m x t*</u>, $\boldsymbol{\Sigma}_t$ is <u>*t x t*</u>, and $\mathbf{V}_t^*$ is <u>*t x n*</u>

Derivation

- Second most commonly used technique is to apply singular value decomposition (SVD) on data matrix **X**.

- SVD of **X** is: $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$
  - Where **U** and **V** are orthogonal bases for the column and row spaces of **X**, and Σ is a diagonal matrix of the singular values.
  - Singular values are “stretch factors” that help to match **u**’s with **v**’s

$$\sigma_i \mathbf{u}_i = \mathbf{X}\mathbf{v}_i \; (i = 1, \dots, D)$$

- Now let’s derive the covariance matrix of **X** (magic begins!)

$$\begin{aligned} \mathbf{X}\mathbf{X}^T &= \mathbf{U}\Sigma\mathbf{V}^T(\mathbf{U}\Sigma\mathbf{V}^T)^T \\ &= \mathbf{U}\Sigma\mathbf{V}^T\mathbf{V}\Sigma\mathbf{U}^T \\ &= \mathbf{U}\Sigma^2\mathbf{U}^T \end{aligned}$$

Derivation

- Covariance matrix of **X** now becomes:

$$\begin{aligned}\mathbf{X}\mathbf{X}^T &= \mathbf{U}\Sigma\mathbf{V}^T(\mathbf{U}\Sigma\mathbf{V}^T)^T \\ &= \mathbf{U}\Sigma\mathbf{V}^T\mathbf{V}\Sigma\mathbf{U}^T \\ &= \mathbf{U}\Sigma^2\mathbf{U}^T\end{aligned}$$

- This is identical to: $\mathbf{C} = \mathbf{P}\Lambda\mathbf{P}^T$
  - Only that **(singular value)^2 = eigenvalue**

- In other words, performing SVD on **X** is equivalent to performing eigen-decomposition on **XX**^{T}

Application and Caveat

- PCA is very useful if one has limited amount of data
  - e.g., face image (128x128=16384, huge dimension)

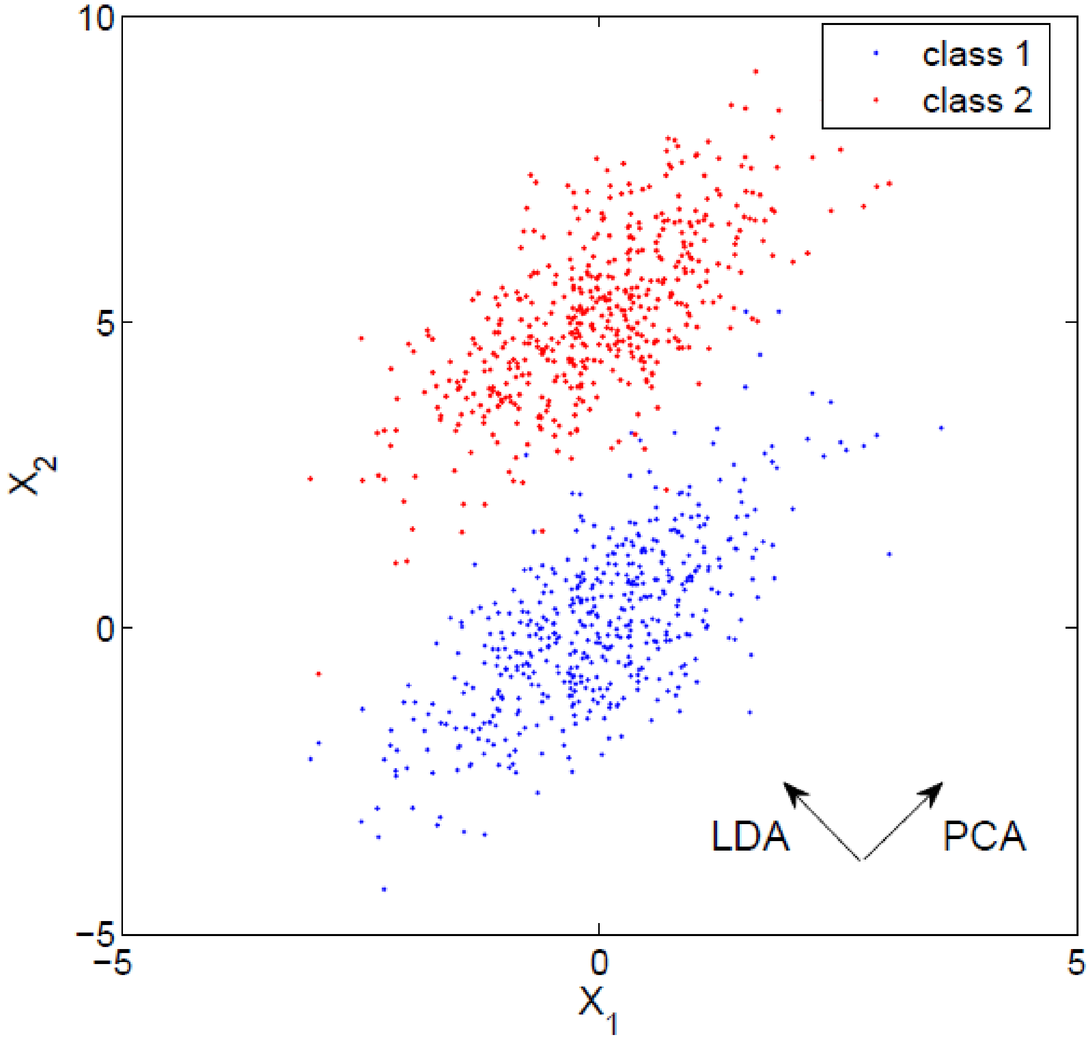
- After dimensionality reduction, the dataset becomes more applicable. (easier to handle in regression, classification, etc)

- Eigenface

- The truncated covariance matrix $\mathbf{C}' = \mathbf{P}\Lambda'\mathbf{P}^T$ is a low-rank approximation of C.

Application and Caveat

- Maximal variability **does not** imply maximal discriminability



Further Reading

- More component analysis (CA) methods
  - Linear discriminant analysis (LDA)
  - Canonical correlation analysis (CCA)
  - Laplacian eigenmaps (LE)
  - Spectral clustering (SC)
  - Independent component analysis (ICA)

References

- Jonathon Shlens, "A Tutorial on Principal Component Analysis," http://www.snl.salk.edu/~shlens/pca.pdf

- Chris Bishop, "Pattern Recognition and Machine Learning", Chapter 12.1