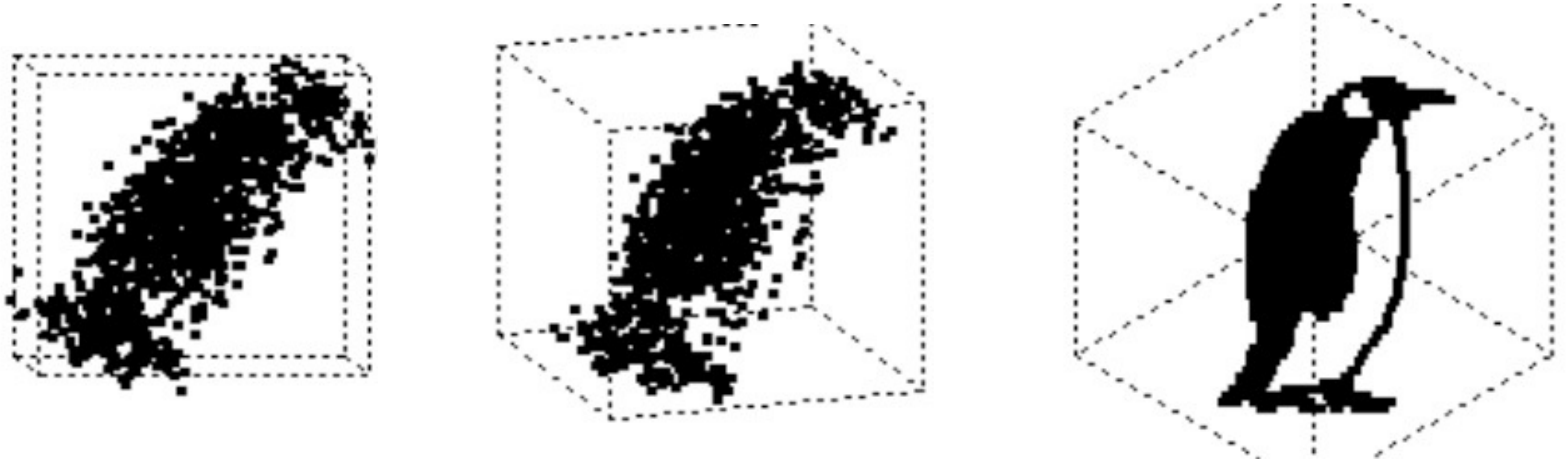
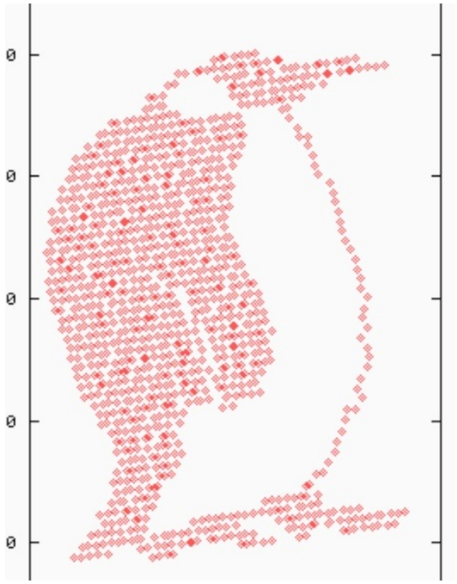
# CSI 436/536
# Introduction to Machine Learning

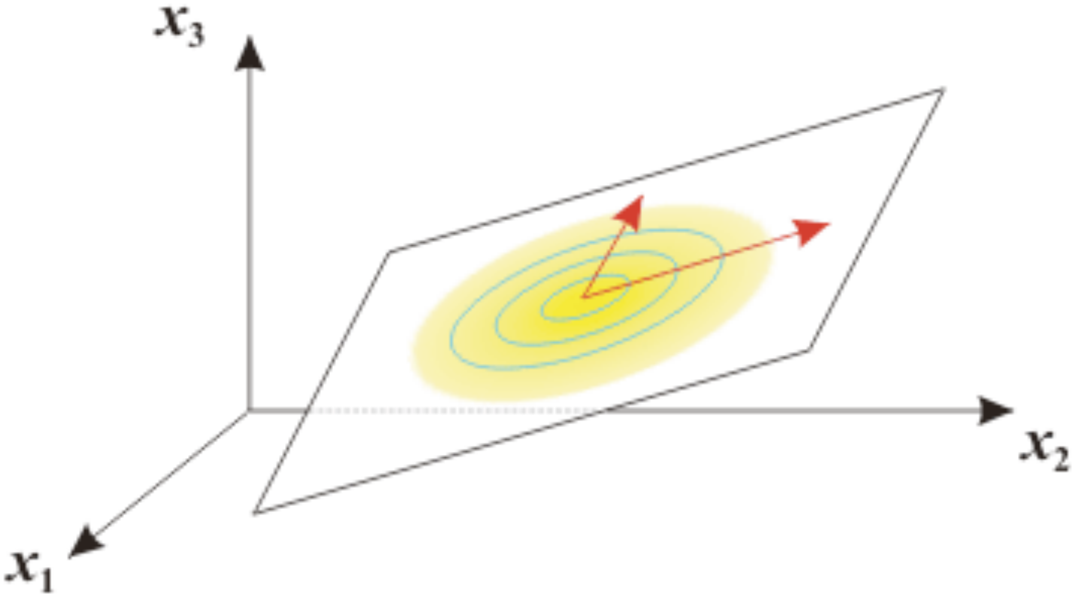
## Dimension reduction: PCA

Professor Siwei Lyu
Computer Science
University at Albany, State University of New York

# The need to reduce data dimension







# The need to reduce data dimension

- The curse of dimensionality
    - effective learning requires more training data as dimensionality increases
    - accuracy and efficiency of learning algorithms degrade rapidly as the dimension increases
- Data intrinsic dimension may be small, and using the original data representation could be wasteful
    - data compression
- Only a fraction of information in data are relevant to the learning task, and small variations may be due to noise
    - feature extraction and noise reduction
- Humans can get intuition in 2D or 3D data (visualization)

# Dimension reduction

- For an input high dimensional data source $x \in \mathscr{R}^d$, find a low dimensional representation $\tilde{x} \in \mathscr{R}^m$ with $m \ll d$ that "best" approximate the original data
  - Determine a pair of transforms $\phi : \mathscr{R}^d \mapsto \mathscr{R}^m$ (***encoder***) and $\psi : \mathscr{R}^m \mapsto \mathscr{R}^d$ (***decoder***) such that $\tilde{x} = \phi(x)$, and $L(x - \psi(\tilde{x})) = L(x - \psi(\phi(x)))$ is minimized, where $L$ is a loss function
  - Dimension reduction is an example of unsupervised learning problem (**self-supervised learning**)
- The dimensionality constraint is served as an ***information bottleneck***, filtering out less relevant information as discarded dimension

# Dimension reduction

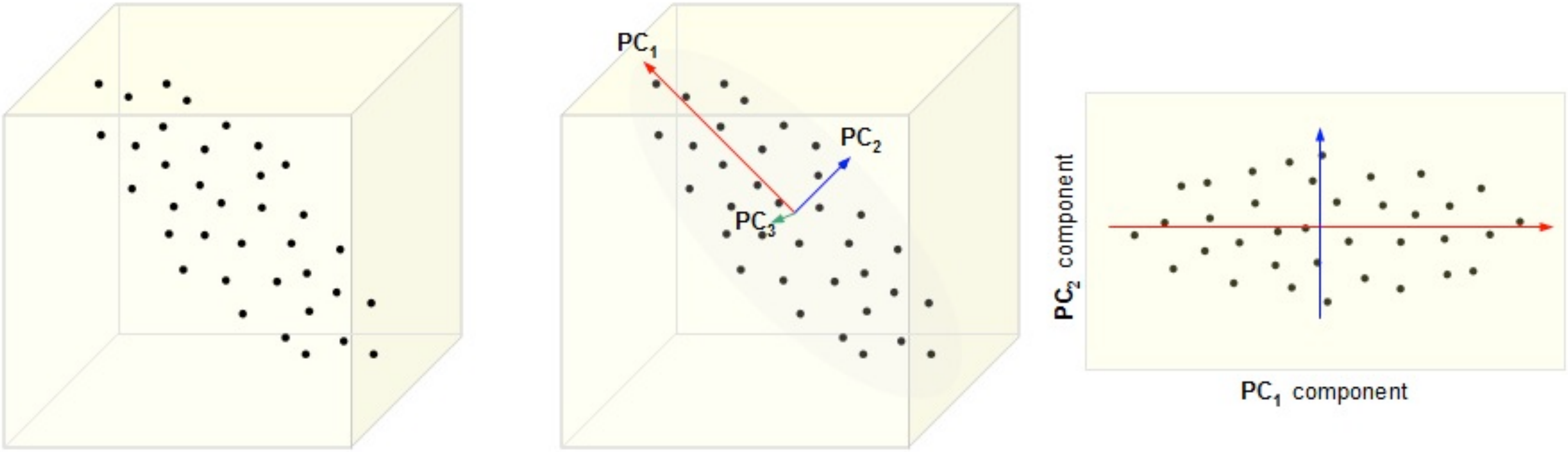
- Representation learning
  - The encoder facilitates the process by ordering the information content corresponding to different levels of relevancy for a more efficient representation
  - The decoder reconstruct the data from the low dimensional representation
- If the encoder and decoder are linear functions, then it looks for an optimal ***linear subspace***, otherwise, it seeks an optimal low dimensional ***manifold***
  - Learning low dimensional linear subspace: PCA
  - Learning low dimensional manifold: ISOMAP

# Linear dimension reduction — PCA

- Assume centered data matrix $X = \begin{pmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{pmatrix}$
- Choose encoder and decoder as orthonormal transforms $W \in \mathscr{R}^{d \times m}$, so the low dimensional representation will be $\tilde{X} = W^\top X \in \mathscr{R}^{m \times n}$, and the decoder as W itself, with $W^\top W = I_m$,
- Choose the metric as the squared matrix L$_2$ distance (also known as the Frobenius distance), the overall learning problem is a constrained optimization problem $\min_W \|X - W(W^\top X)\|_F^2$, s.t $W^\top W = I_m$
  - Solution: W corresponds to the eigenvectors of the top-m eigenvalues of matrix $XX^\top$ (principal components)

# PCA

- In the 1-D case, it is equivalent to the total least squares
- In the m-D case, ***principal values*** are the top-m eigenvalues of the data covariance matrix, while ***principal components*** are the corresponding eigenvectors
  - Principal values measure variance/level of uncertainty/ information in the direction of principal component

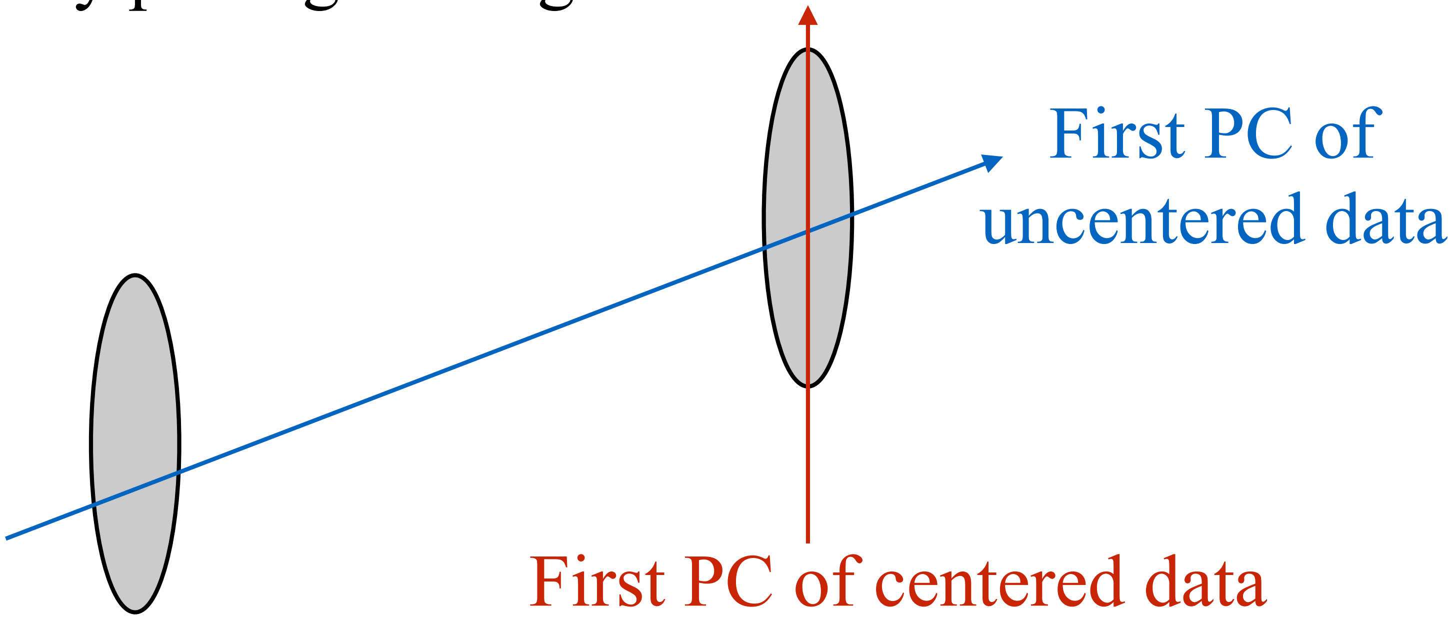


# The PCA algorithm — training

- Computing data mean $\mu = \frac{1}{n}X1_n$, where $1_n$ is all 1 vector
- Centering data matrix $\bar{X} = X\left(I_n - \frac{1}{n}1_n 1_n^T\right)$
- Forming the covariance matrix [centering matrix is symmetric and idempotent]
  $C = \bar{X}\bar{X}^T = X\left(I_n - \frac{1}{n}1_n 1_n^T\right)X^T$
- Eigen-decomposing (diagonalizing) the covariance matrix [spectral theorem] $C = U\Lambda U^T$
- Choosing eigenvectors of the top-m eigenvalues W = [$u_1$, …, $u_m$], corresponding $\lambda_1 \geq \cdots \geq \lambda_m$

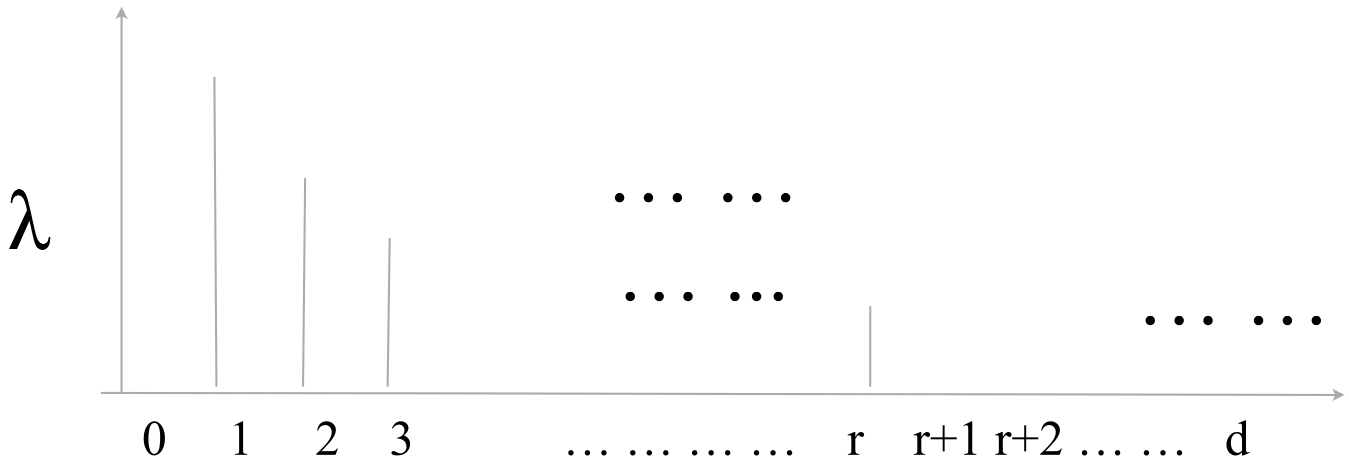
# The PCA algorithm — new data

- The model includes m and W: O((m+1)d) vs. O(nd)
- given a new data point **x**, its representation is $\tilde{x} = W^T(x - \mu)$, and its reconstruction is $W\tilde{x} + \mu$
- why centering is important:
  - matrix R = XX$^T$ is the correlation matrix and 2R is the covariance matrix for data[X -X], this is the data set formed by putting the negates of each individual data



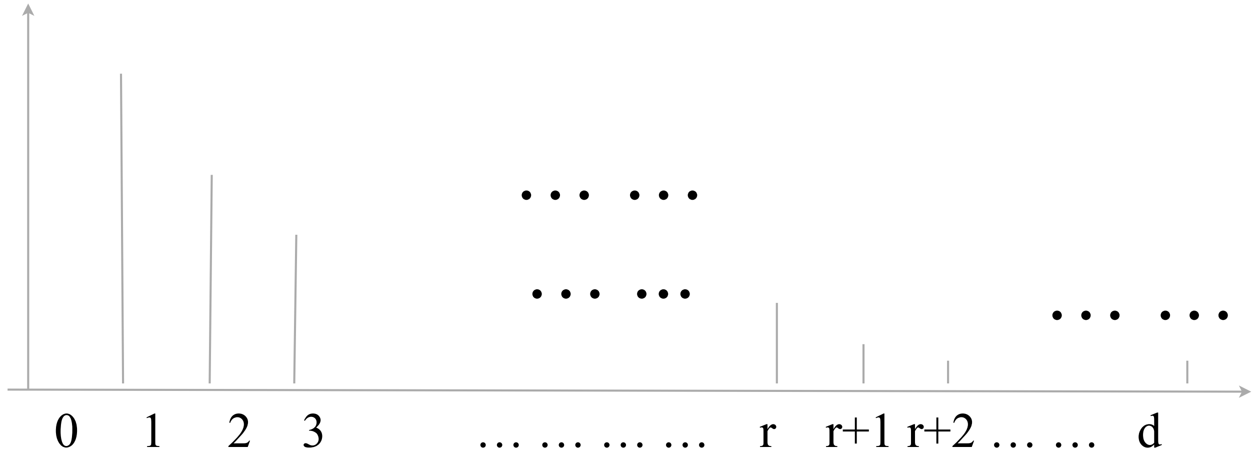
# how to choose dimension

- for data matrix X with rank r < d, all principal values $\lambda_{r+1},\ldots,\lambda_d$, are zero, assuming descending order of principal values



# how to choose the cutoff dimension

- assume $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_d > 0$

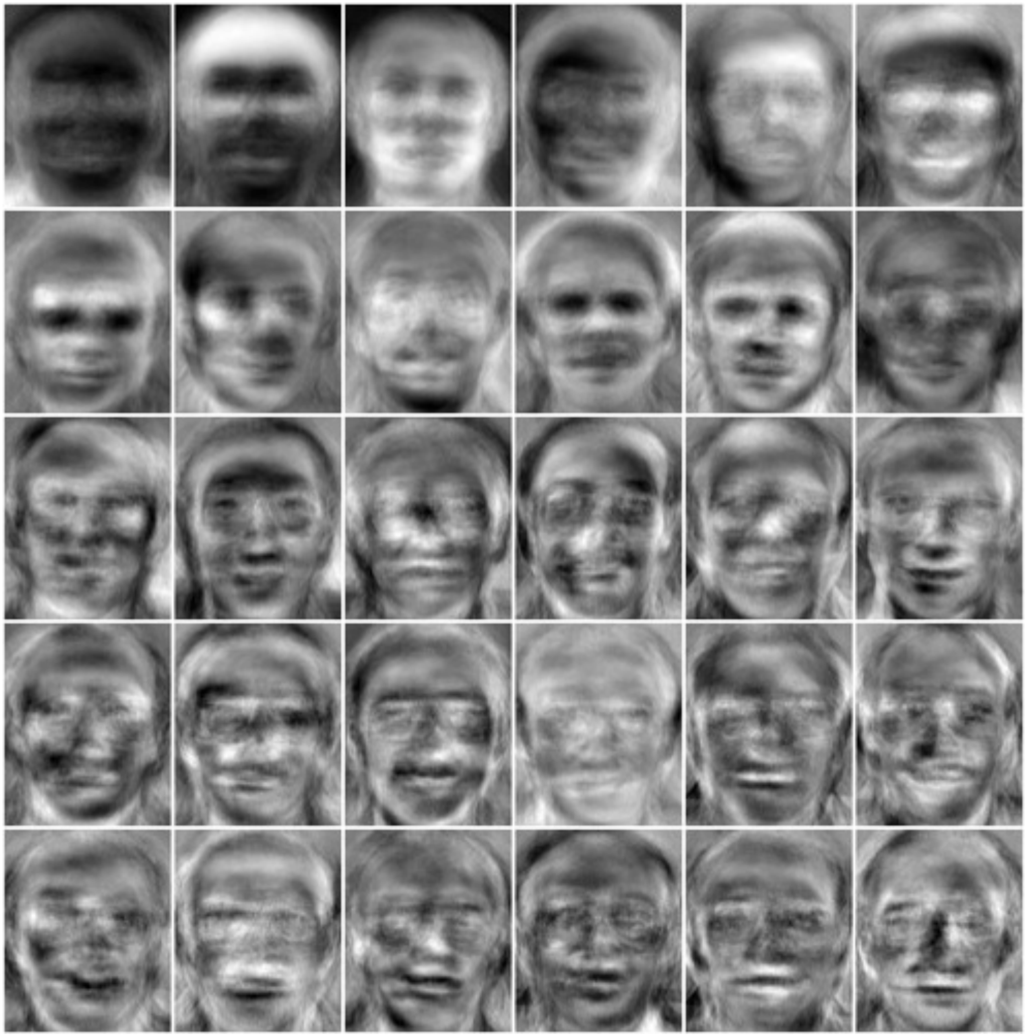


- heuristic 1: pick threshold $\varepsilon > 0$, and choose r such that $\lambda_r \geq \varepsilon \geq \lambda_{r+1} \ldots \geq \lambda_d > 0$ any principal value smaller than the threshold may be caused by noise
- heuristic 2: pick fraction $0 < \eta < 1$, and choose r such that keep significant fraction of energy

$$\frac{\sum_{i=1}^{r-1} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \leq \eta \leq \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

# application of PCA: face recognition

- face recognition algorithm developed by M. Turk and A. Pentland in 1987
- represent a high dim face image with first few *eigenfaces*
- the coefficients of a face on *eigenfaces* are used to match against all faces in a database



# The covariance and Gramm matrices

- The covariance matrix is of dimension d x d
  - when d is large, eigenvalue decomposition of covariance matrix can be difficult
- The Gram (inner product) matrix: $G = X^T X$ is of dimension n x n
- correlation (covariance, outer product) matrix: $C = XX^T$
  - G and C are both positive definite matrices
  - G and C share the same **non-zero** eigenvalues
  - G and C’s eigenvectors are related by X

# PCA with Gram matrix

- PCA is based on the eigenvalue decomposition of the covariance matrix.
- Covariance matrix and the (centered) Gram matrix share eigenvalues and eigenvectors are related.
- PCA with the centered Gram matrix
  - Centering data matrix $\bar{X} = X\left(I_n - \frac{1}{n}1_n 1_n^T\right)$
  - Form the centered Gram matrix (double centering) $\bar{G} = \bar{X}^T\bar{X} = \left(I - \frac{1}{n}11^T\right)X^T X\left(I - \frac{1}{n}11^T\right)$
  - Eigenvalue decompose $\bar{G} = U\Lambda U^T$
- Choosing eigenvectors of the top-m eigenvalues W = X[$u_1$, …, $u_m$], corresponding $\lambda_1 \geq \cdots \geq \lambda_m$