# CS-565 Computer Vision

Nazar Khan

PUCIT

Lecture 30

# PRINCIPAL COMPONENT ANALYSIS

# Principal Component Analysis

- Widely used technique for dimensionality reduction and object recognition.
- Projects a set of signals onto a lower dimensional orthogonal space.
- Abbreviated as PCA.
- Also known as the Karhunen-Loeve transform.

# PCA

- Consider a set of signals $X=[x_1,\ldots, x_N]$ where each $x_i = \in \mathbb{R}^D$.
- **Goal**: Project each $x_i$ onto a space with dimensionality $M<D$ while maximising the variance of the projected data.

# PCA

- To begin, let us set M=1, i.e, projection onto a 1-dimensional space.
- We can define the direction of this space by a vector $u_1 \in \mathbb{R}^D$.
- For convenience, let $u_1^T u_1 = 1$
  - We are only interested in the direction defined by $u_1$ and not the magnitude of $u_1$.

# PCA

- Each data point $x_i$ is projected onto a scalar value $u_1^T x_i$.

Mean of the projected data is given by $u_1^T \bar{x}$ where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$

is the mean of the data points.

Variance of the projected data is given by $\frac{1}{N}\sum_{i=1}^{N}\left(u_1^T x_i - u_1^T \bar{x}\right)^2 = u_1^T S u_1$

where $S = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)\left(x_i - \bar{x}\right)^T$ is the data covariance matrix.

# PCA

- Our goal was to maximise the variance of the projected points.
- That is, we want to maximise $u_1^T S u_1$ with respect to $u_1$.
- To prevent $||u_1|| \rightarrow \infty$, we must constrain the norm of $u_1$.
  - This constraint comes from the normalization condition $u_1^T u_1 = 1$.

# PCA

- To maximise $f(u_1)=u_1^T S u_1$ with the contstraint $u_1^T u_1=1$, we use the **method of Lagrange multipliers**.
- Let $g(u_1)=1-u_1^T u_1$ denote the **constraint function**.
- Our constrained maximisation $f(u_1)$ is equivalent to the unconstrained maximisation of $f(u_1)+\lambda_1 g(u_1)$.

# PCA

- Set $d/du_1\ f(u_1)+\lambda_1 g(u_1)$ equal to zero to find optimal $u_1$.

  $$d/du_1\ (u_1^T S u_1)+\lambda_1(1- u_1^T u_1)=0$$

  $$Su_1=\lambda_1 u_1$$

  which says that the optimal u1 must be an eigenvector of S.
- By left-multiplying by $u_1^T$ we see that $u_1^T S u_1=\lambda_1$. That is $f(u_1)=\lambda_1$.
- So, $u_1$ must be the eigenvector corresponding to the largest eigenvalue of S.
  - This eigenvector is also known as the **first principal component**.

# PCA

- For M>1, note that eigenvectors of S are orthogonal to each other.
- So the eigenvector $u_2$ corresponding to the second largest eigenvalue $\lambda_2$ of S gives the direction of maximum variance orthogonal to $u_1$.
- Similarly, the eigenvector $u_i$ corresponding to the $i^{th}$ largest eigenvalue $\lambda_i$ of S gives the direction of maximum variance orthogonal to the subspace $[u_1, u_2, \dots , u_{i-1}]$.

# PCA

**Summary**

- Compute data covariance matrix

$$S = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})(x_i - \overline{x})^T$$

- Pick the M eigenvectors of S corresponding to the M largest eigenvalues.

# Principal Theorem of Eigenspace Representations

- Consider N images that are represented as vectors $f_1,\ldots,f_N \in \mathbb{R}^D$.
  - Usually one has less images than pixels, i.e. N<<D (e.g. D = 65536, N = 1000).
- Then the D×D covariance matrix S is symmetric, and
  - has at most N nonvanishing eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N > 0$.
  - with corresponding orthonormal eigenvectors $u_1,\ldots,u_N$.
- Every image $f_i$ can be represented using these m eigenvectors of S:

$$f_i = \bar{f} + \sum_{j=1}^{N} a_{ij} u_j \quad (i = 1,\ldots,N)$$

$$\text{where } a_{ij} = \left(f_i - \bar{f}\right)^T u_j$$

Projection of difference from mean onto eigenvector $u_j$.

# Dimensionality Reduction using PCA

- Since eigenvalues represent the variance along the direction of the corresponding eigenvector, eigenvalues close to 0 and their eigenvectors can be ignored.
  – They do not represent directions of significant variation.
- Usually, only k<<N significant eigenvalues exist where N=number of non-zero eigen-values.
  – For example k=5 and N=1000.
- So, each data point $f_i$ can be represented even more compactly

$$\boxed{\begin{array}{l} f_i = \bar{f} + \sum_{j=1}^{k} a_{ij} u_j \quad (i = 1, \ldots, N) \\ \text{where } a_{ij} = \left(f_i - \bar{f}\right)^T u_j \end{array}}$$

# Computational Aspects

- Usually, covariance matrix $S \in R^{D \times D}$ is very large.
  - Images of size 256 × 256 pixels yield D = 65536.
  - Thus, S has size 65536 × 65536.
  - Since the matrix S is not sparse, one would not even want to store it, let alone compute its eigen decomposition.
  - A direct computation of all eigenvalues and eigenvectors of S would be far too time consuming.
- However, there is a trick.

# Computational Aspects

- Define $D=[x_1-\bar{x},\dots,x_N-\bar{x}]$.
- Then $S=DD^T/N$ is the DxD covariance matrix. (**Verify**)
- Since N<<D, let us consider the much smaller matrix $T=D^TD/N$.
  - The m eigenvalues of T are also eigenvalues of S.
  - Moreover, T contains all nonvanishing eigenvalues of S:
    - The remaining D−N eigenvalues of S are zero.
  - If $w_i$ is an eigenvector of T, then $v_i := Dw_i$ is an eigenvector of S.
    - norm($v_i$) might not be 1, so it must be renormalised.
- Advantage: instead of working with a 65536 × 65536 matrix, work with a 1000 x 1000 matrix.

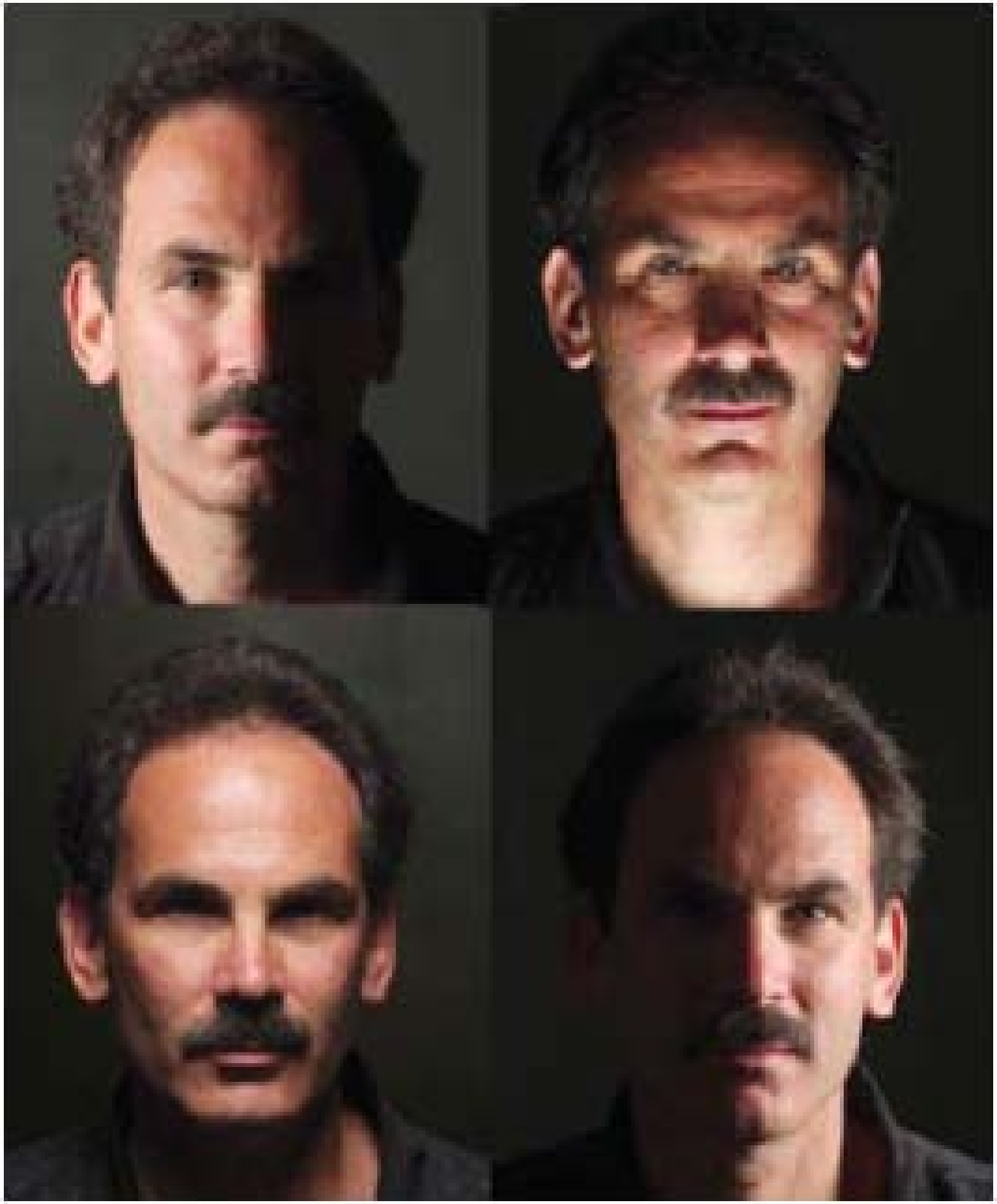
# Computational Aspects

- One can also ignore the eigen-decomposition completely and compute the M largest eigenvalues and their corresponding eigenvectors via the iterative **Power Method**.

# Training and Recognition via PCA

- Image sets of different objects can yield their corresponding subspaces.
  - $X_{planes} \rightarrow U_{planes}$ via PCA
  - $X_{bikes} \rightarrow U_{bikes}$ via PCA
- A new object can be projected onto both subspaces.
- The subspace with the smallest error gives the most similar object in the data base.

12 images of a 3-D object being viewed from different directions. Author: S. Kiefer (2006).



The same face illuminated from four different directions. Author: D. Kriegman.