Principal Component Analysis

Lecturer:
Jiří Matas

Authors:
Ondřej Drbohlav, Jiří Matas

Centre for Machine Perception
Czech Technical University, Prague
http://cmp.felk.cvut.cz

1.1.2017

- Alternative name: Karhunene Loeve transform
- Used for: data approximation, identifying sources of variance in the data

Maximum variance formulation (1/3)

Let the data be $\{\mathbf{x}_i \mid i = 1, 2, ..., N\}$, with sample mean $\overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n$.

Let us find the unit vector $\mathbf{u}_1$ to project to such that the variance $J(\mathbf{u}_1)$ of the projected data is *maximized*. The projection $\mathbf{x}_n^{(\mathrm{p})}$ of an $\mathbf{x}_n$ to one-dimensional subspace generated by $\mathbf{u}_1$ is given by

$$\mathbf{x}_n^{(\mathrm{p})} = \mathbf{u}_1 \left(\mathbf{u}_1^{\mathrm{T}} \mathbf{x}_n\right), \quad \mathbf{u}_1^{\mathrm{T}} \mathbf{u}_1 = 1 . \tag{1}$$

The variance $J(\mathbf{u}_1)$ of projected data is

$$J(\mathbf{u}_1) = \frac{1}{N} \sum_{n=1}^{N} \left(\mathbf{u}_1^{\mathrm{T}} \mathbf{x}_n - \mathbf{u}_1^{\mathrm{T}} \overline{\mathbf{x}}\right)^2 = \frac{1}{N} \sum_{n=1}^{N} \mathbf{u}_1^{\mathrm{T}} (\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^{\mathrm{T}} \mathbf{u}_1 = \mathbf{u}_1^{\mathrm{T}} \mathbf{S} \mathbf{u}_1 , \tag{2}$$

where $\mathbf{S}$ is the normalized scatter matrix:

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^{\mathrm{T}} . \tag{3}$$

The Lagrangian of this optimization problem is

$$L(\mathbf{u}_1, \lambda_1) = J(\mathbf{u}_1) + \lambda_1 \underbrace{(1 - \mathbf{u}_1^{\mathrm{T}} \mathbf{u}_1)}_{\text{constraint}} = \mathbf{u}_1^{\mathrm{T}} \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^{\mathrm{T}} \mathbf{u}_1)\,, \tag{4}$$

where λ_1 is the Lagrange multiplier. Taking the derivative w.r.t. the vector $\mathbf{u}_1$ and setting it to zero gives

$$\frac{\partial L(\mathbf{u}_1, \lambda_1)}{\partial \mathbf{u}_1} = \mathbf{S}\mathbf{u}_1 - \lambda_1 \mathbf{u}_1 = 0\,, \tag{5}$$

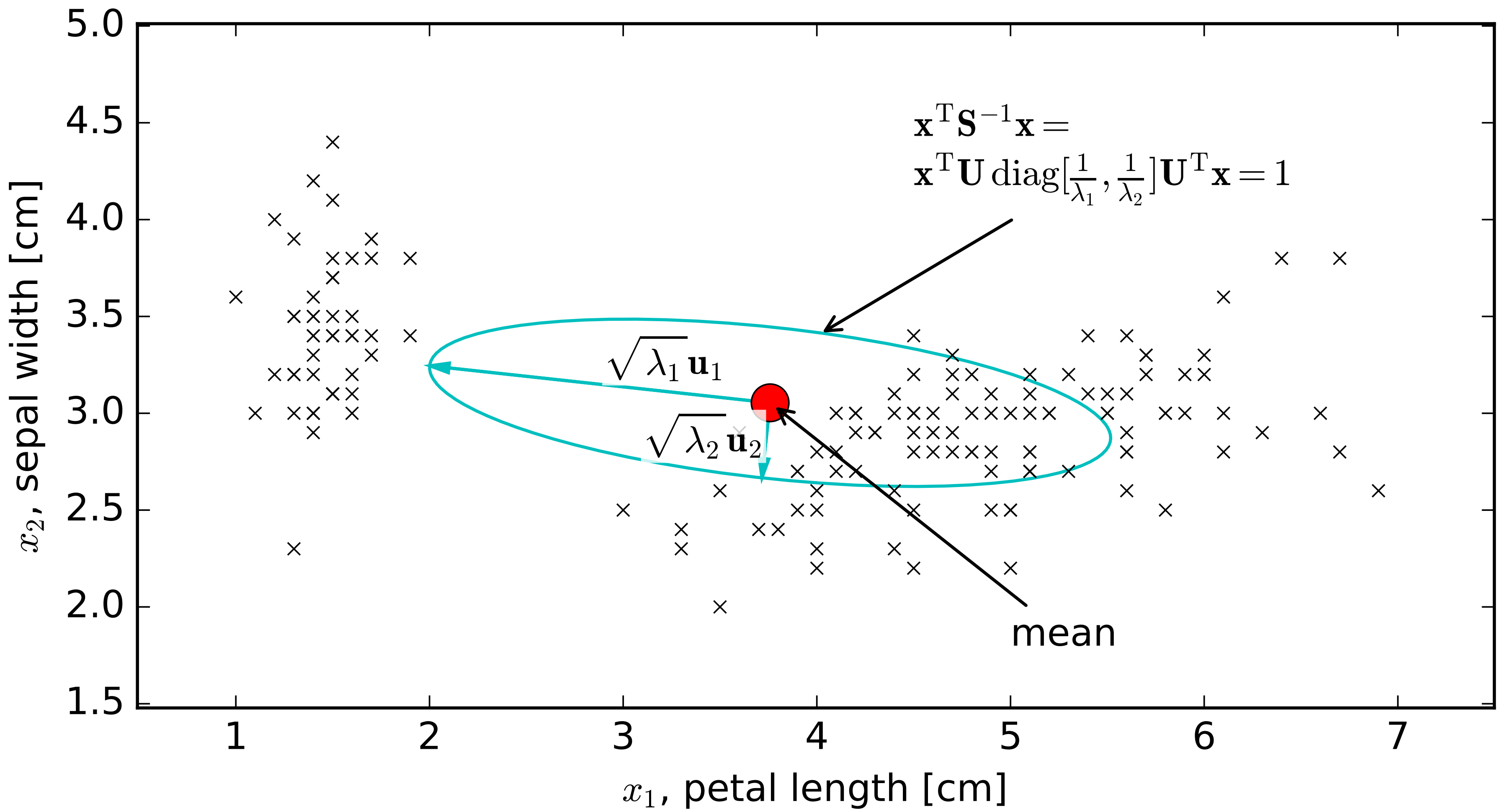
and thus

$$\mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1\,. \tag{6}$$

This is the characteristic equation for the covariance matrix $\mathbf{S}$. Any eigenvalue λ_1 and its corresponding eigenvector $\mathbf{v}_1$ solves this equation, with variance $J(\mathbf{u}_1)$ equal to:

$$J(\mathbf{u}_1) = \mathbf{u}_1^{\mathrm{T}} \mathbf{S} \mathbf{u}_1 = \mathbf{u}_1^{\mathrm{T}} \lambda_1 \mathbf{u}_1 = \lambda_1\,. \tag{7}$$

The maximum is attained if λ_1 is the largest eigenvalue of the matrix $\mathbf{S}$ and $\mathbf{u}_1$ is its corresponding eigenvector.



Data shown as crosses $\times$. Iris dataset: feature vectors are 4-dimensional, here dimensions 2 and 3 used (petal length and sepal width).

Eigenvalues: $\lambda_1 = 3.148$, $\lambda_2 = 0.153$, eigenvectors $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2]$. Variance is maximized when data are projected to direction $\mathbf{u}_1$.

Recall: The variance of a 1-D projection is maximized when data are projected to the direction of the eigenvector of $\mathbf{S}$ corresponding to the largest eigenvalue.

$\mathbf{S}$ is symmetric and positive semidefinite. The eigenvectors corresponding to different eigenvalues are orthogonal.

It follows that the D-dimensional subspace maximizing the variance of the data is the one formed by D eigenvectors of $\mathbf{S}$ corresponding the the D largest eigenvalues.

Note: "Variance" in the above sentence is the sum of variances in individual orthogonal directions. For a 2-D subspace,

$$J(\mathbf{u}_1, \mathbf{u}_2) = \frac{1}{N} \sum_{n=1}^{N} [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x}_n - \overline{\mathbf{x}})]^2 + [\mathbf{u}_2^{\mathrm{T}}(\mathbf{x}_n - \overline{\mathbf{x}})]^2 \,. \tag{8}$$

Equivalence to Minimum error formulation (1/2)

Consider the complete orthogonal basis $\{\mathbf{u}_i\}$ where $i = 1, \ldots, D$. Thus

$$\mathbf{u}_i^{\mathrm{T}} \mathbf{u}_j = \delta_{ij} \tag{9}$$

Each point can be represented as

$$\mathbf{x}_n = \sum_{i=1}^{D} \alpha_{ni} \mathbf{u}_i \,, \tag{10}$$

and

$$\mathbf{x}_n = \sum_{i=1}^{D} (\mathbf{x}_n^{\mathrm{T}} \mathbf{u}_i) \mathbf{u}_i \,. \tag{11}$$

This is just expressing $\mathbf{x}_n$ in a rotated coordinate system given by orthonormal system $\{\mathbf{u}_i\}$. Let us create an approximation to each $\mathbf{x}_n$ by truncating this expansion to only M components, the remaining $D - M$ components approximated by constants b_i. The approximation $\tilde{\mathbf{x}}_n$:

$$\tilde{\mathbf{x}}_n = \sum_{i=1}^{M} (\mathbf{x}_n^{\mathrm{T}} \mathbf{u}_i) \mathbf{u}_i + \sum_{i=M+1}^{D} b_i \mathbf{u}_i \tag{12}$$

$$\tilde{\mathbf{x}}_n = \sum_{i=1}^{M} (\mathbf{x}_n^{\mathrm{T}} \mathbf{u}_i) \mathbf{u}_i + \sum_{i=M+1}^{D} b_i \mathbf{u}_i \tag{12}$$

Clearly,

$$b_i = \overline{\mathbf{x}}^{\mathrm{T}} \mathbf{u}_i, i = M+1, \ldots, D \tag{13}$$

The task is to find the optimal orthonormal basis $\{\mathbf{u}_i\}$ which produces the best approximation measured by

$$J(\{\mathbf{u}_i\}) = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2 \tag{14}$$

The minimum error criterion is the complement of the maximum variance criterion, and thus the solution to the set $\{\mathbf{u}_i\}$ is the same.

Recall that the ML estimate of the Multivariate Normal Distribution is defined by sample mean $\overline{\mathbf{x}}$ and sample covariance matrix $\mathbf{S}$. The model is

$$p(\mathbf{x} \mid \overline{\mathbf{x}}, \mathbf{S}) = \frac{1}{\sqrt{|2\pi\mathbf{S}|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\overline{\mathbf{x}})^{\mathrm{T}}\mathbf{S}^{-1}(\mathbf{x}-\overline{\mathbf{x}})\right\} \tag{15}$$

Denote stacked eigenvectors in descending order of their eigenvalues as $\mathbf{U}$,

$$\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_D\} \tag{16}$$

Therefore (characteristic equation)

$$\mathbf{S}\mathbf{U} = \mathbf{U}\boldsymbol{\Lambda} = \mathbf{U}\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D \end{bmatrix}, \tag{17}$$

and

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}. \tag{18}$$

We approximate the data, as before, by projecting to first M eigenvectors. Thus, given data point $\mathbf{x}$ we have

$$\mathbf{x} - \overline{\mathbf{x}} = (\delta_1, \delta_2, ..., \delta_M, \delta_{M+1}, ..., \delta_D) \tag{19}$$

Note that we only can compute δ_1 .. δ_M, as often we don't or can't store all eigenvectors for computing all δ's. However, we can easily compute

$$\Delta = \delta_{M+1}^2 + \delta_{M+2}^2 + ... + \delta_D^2 = \|\mathbf{x} - \overline{\mathbf{x}}\|^2 - \delta_1^2 - \delta_2^2 - ... - \delta_M^2 \tag{20}$$

and the exponent is then approximated as

$$-\frac{1}{2}(\mathbf{x} - \overline{\mathbf{x}})^{\mathrm{T}}\mathbf{S}^{-1}(\mathbf{x} - \overline{\mathbf{x}}) \simeq -\frac{1}{2}\left(\frac{\delta_1^2}{\lambda_1} + \frac{\delta_2^2}{\lambda_2} + \frac{\delta_3^2}{\lambda_3} + ...\frac{\delta_M^2}{\lambda_M} + \frac{\Delta}{\lambda}\right) \tag{21}$$

Common choice: $\lambda = \lambda_{M+1}$

Dimensionality of data can be high, and even higher than number of samples.

Consider dimensionality $D = 1\mathrm{M}$ (one million) and number of samples $N = 100$. All analysis still applies, but it would be wasteful to compute eigenvectors for the 1Mx1M matrix, as its rank will anyway be at most N (thus 100). Let us define $\mathbf{X}$ to be a matrix formed by stacking all the data vectors (after having subtracted the mean from them): $\mathbf{X} = [\mathbf{x}_1 - \overline{\mathbf{x}}, \mathbf{x}_2 - \overline{\mathbf{x}}, ..., \mathbf{x}_N - \overline{\mathbf{x}}]$.

Thus,

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^{\mathrm{T}} = \frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{T}}\,. \tag{22}$$

The characteristic equation is then

$$\frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{u} = \lambda\mathbf{u}\,. \tag{23}$$

Left-multiplying both sides by $\mathbf{X}^{\mathrm{T}}$ gives

$$\frac{1}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}\overbrace{(\mathbf{X}^{\mathrm{T}}\mathbf{u})}^{\mathbf{w}} = \lambda\overbrace{(\mathbf{X}^{\mathrm{T}}\mathbf{u})}^{\mathbf{w}}\,. \tag{24}$$

Thus, $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, which is only 100×100, has exactly the same set of eigenvalues:

$$\frac{1}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} = \lambda\mathbf{w}\,. \tag{25}$$

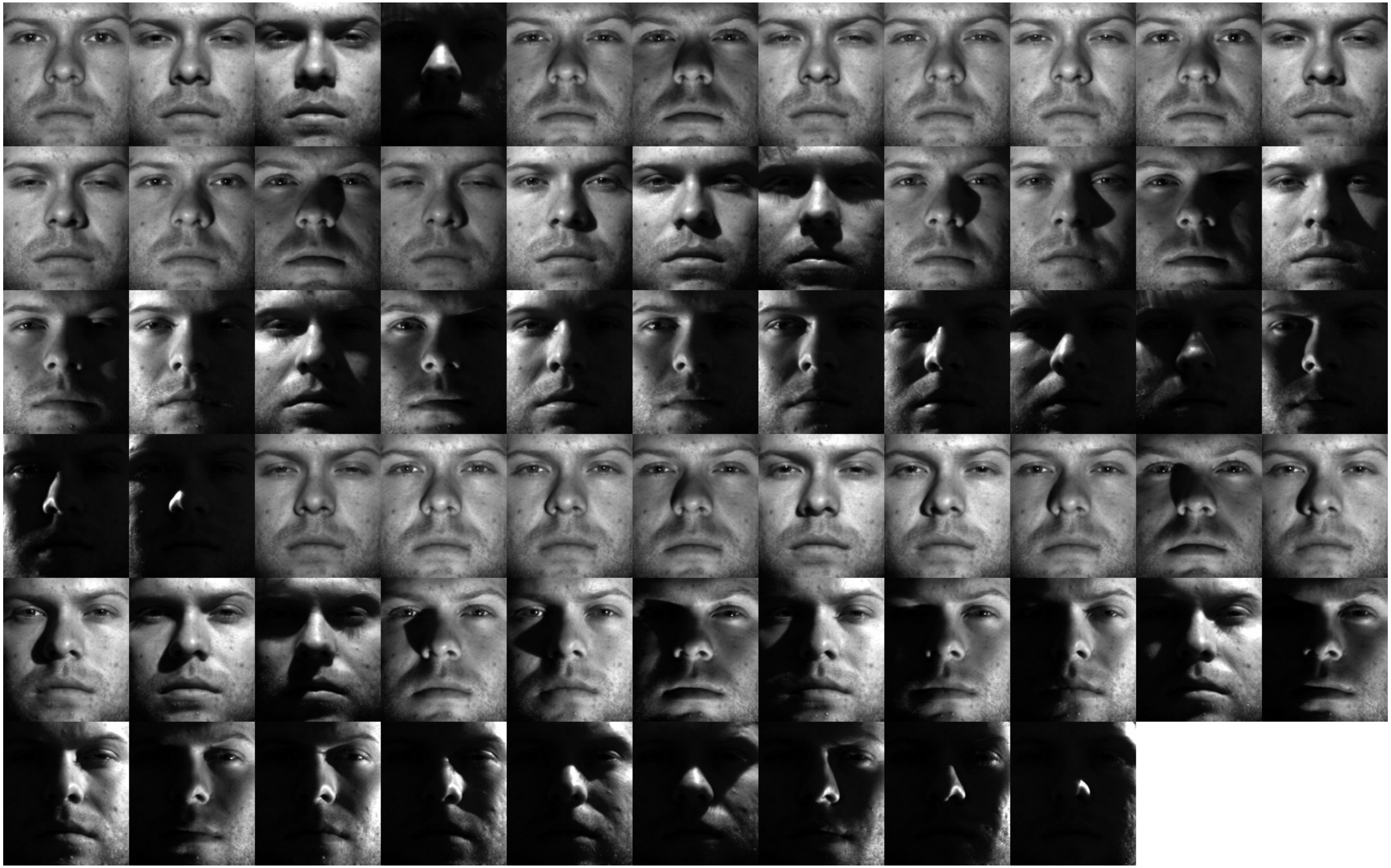
Left-multiplying now by $\mathbf{X}$, we get

$$\frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{w}) = \lambda(\mathbf{X}\mathbf{w})\,. \tag{26}$$

Conclusion: If $D \gg N$, form the matrix $\mathbf{T} = \frac{1}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}$ and compute its eigenvalues λ's and eigenvectors $\mathbf{w}$. Compute the eigenvectors of $\mathbf{S} = \frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{T}}$ as

$$\mathbf{v} = \frac{\mathbf{X}\mathbf{w}}{\|\mathbf{X}\mathbf{w}\|}\,. \tag{27}$$

images of 38 subjects, each under 64 different illumination conditions:



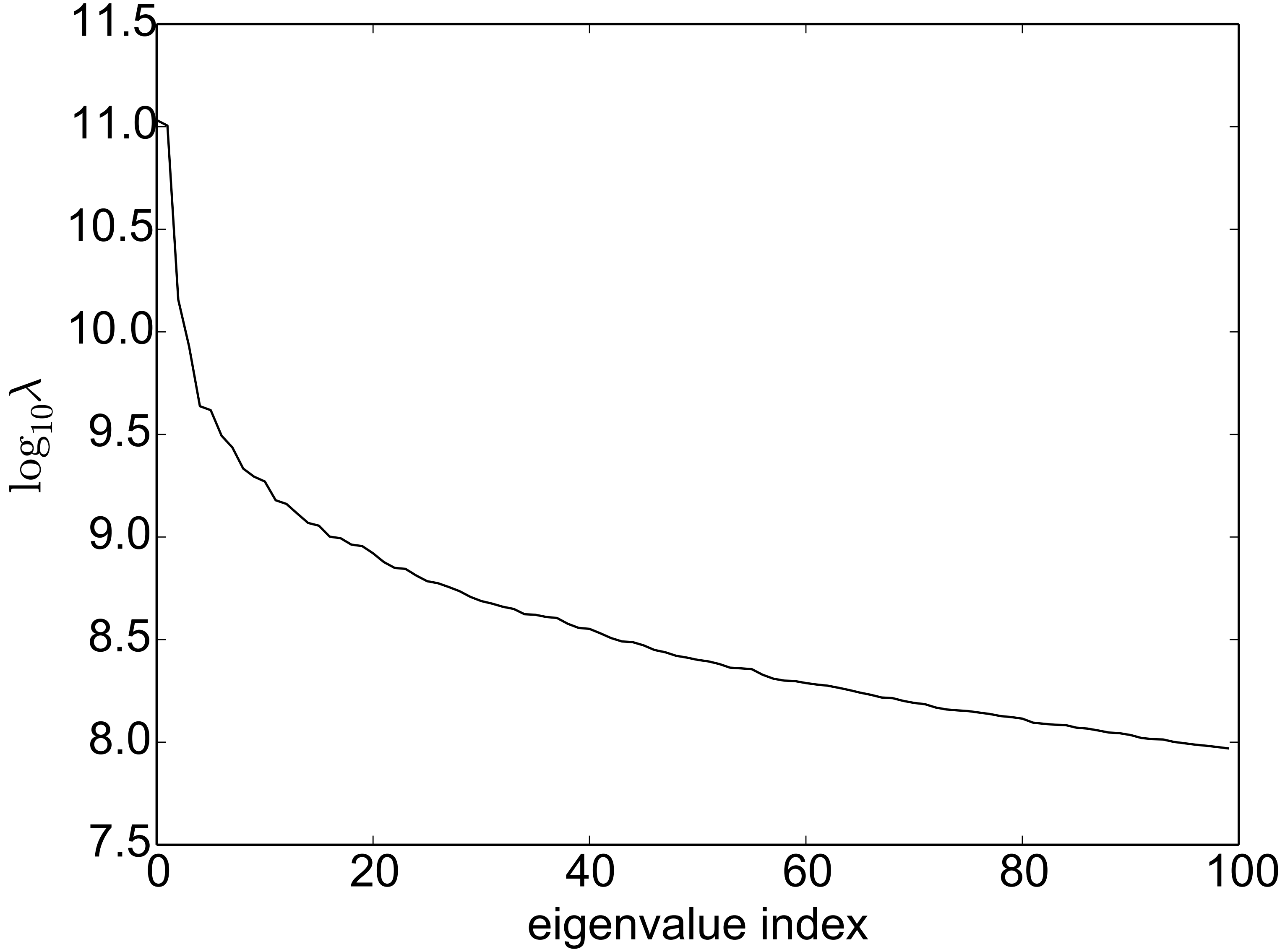
Subject 1, 64 illumination conditions

Example 2 - Yale database (2/5)

images of 38 subjects, each under 64 different illumination conditions:

38 subjects

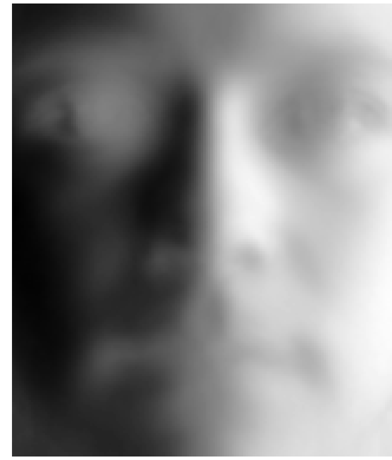
images of 38 subjects, each under 64 different illumination conditions. Thus, there is $38 \times 64 = 2432$ images in total. Each of them is a feature vector with $192 \times 168 = 32256$ dimensions (pixels). PCA gives the following eigenvalues:



Example 2 - Yale database (4/5)

mean

1st ev



2nd ev

3rd ev

first 72 eigenvectors

Example 2 - Yale database (5/5)

Reconstruction of original vector using eigenvectors

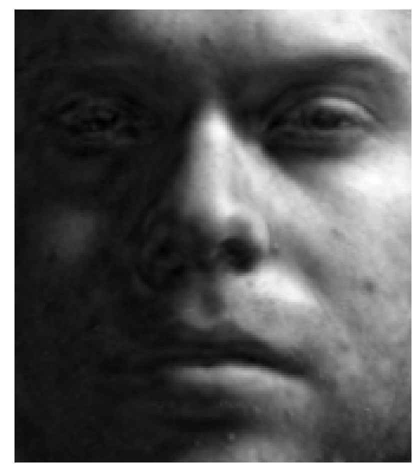
original

mean and 3 evs

mean and 10 evs

mean and 50 evs

mean and 100 evs



mean and 300 evs