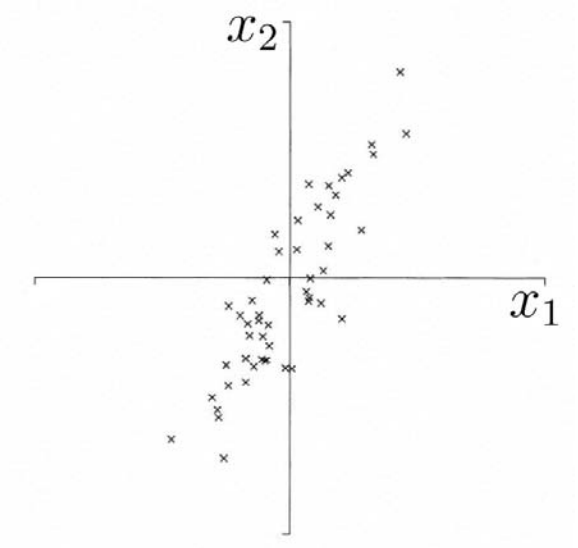
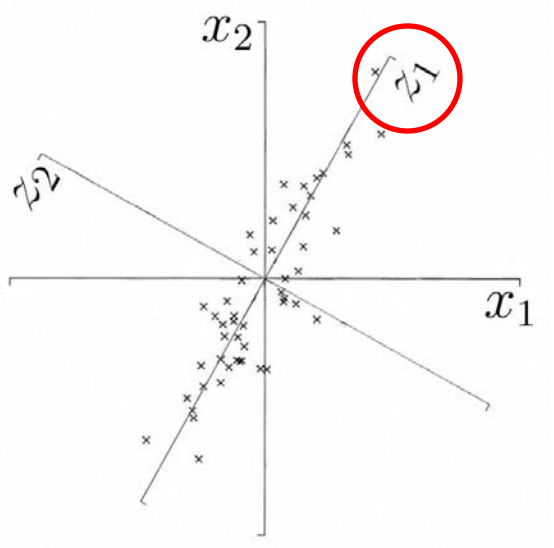
Principal Component Analysis

Why Principal Component Analysis?

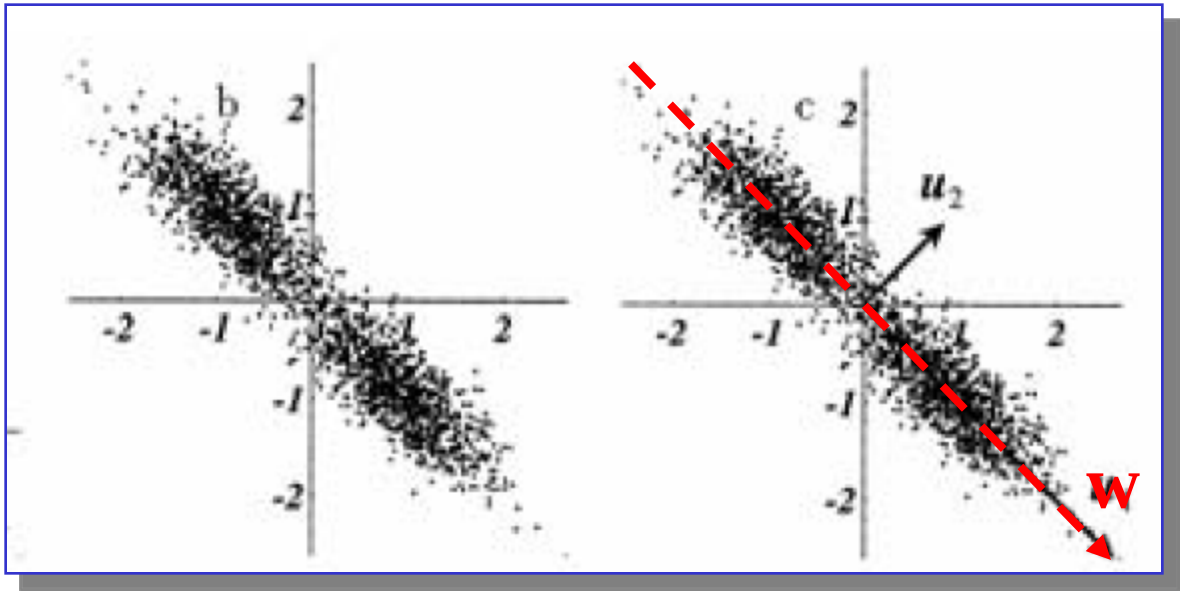
- Motivation
  - Find bases which has high variance in data
  - to remove components containing low/no information.
  - Encode data with small number of bases with low MSE

- Applications:
  - feature extracti
  - visualization
  - compression





•In the Principal Component Analysis (PCA) the goal is to find direction w, where the variance of the data is largest.



What is subspace? (1/2)

- **Find a basis in a low dimensional sub-space:**
  - **Approximate vectors by projecting them in a low dimensional sub-space:**

(1) Original space representation:

$$\begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_N \end{bmatrix}$$

$$x = a_1 v_1 + a_2 v_2 + \dots + a_N v_N$$

where $v_1, v_2, \dots, v_n$ is a base in the original N-dimensional space

(2) Lower-dimensional sub-space representation:

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_K \end{bmatrix}$$

$$\hat{x} = b_1 u_1 + b_2 u_2 + \dots + b_K u_K$$

where $u_1, u_2, \dots, u_K$ is a base in the K-dimensional sub-space (K<N)

- ***Note:*** **if K=N, then** $\hat{x} = x$

What is subspace? (2/2)

- **Example (K=N):**

$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{(standard basis)}$$

$$x_v = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = 3v_1 + 3v_2 + 3v_3$$

$$u_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, u_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, u_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{(some other basis)}$$

$$x_u = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = 0u_1 + 0u_2 + 3u_3$$

thus, $x_v = x_u$

Centered data points x in *n*-dimensional space:

$$\vec{\mathrm{x}} = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix}, \qquad \vec{\mu} = \mathrm{E}\{\vec{\mathrm{x}}\} = \begin{pmatrix} 0 \\ \dots \\ 0 \end{pmatrix}$$

μ is mean value of the vector x.

Covariance matrix C for the centered data:

$$\mathrm{C} = \mathrm{E}\{\vec{\mathrm{x}}\vec{\mathrm{x}}'\} = \mathrm{E}\left\{ \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix} \cdot \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix} \right\},$$

$$c_{i,j} = \mathrm{E}\{x_i x_j\}.$$

Here E{*f*(x)} is expectation value of *f*(x).

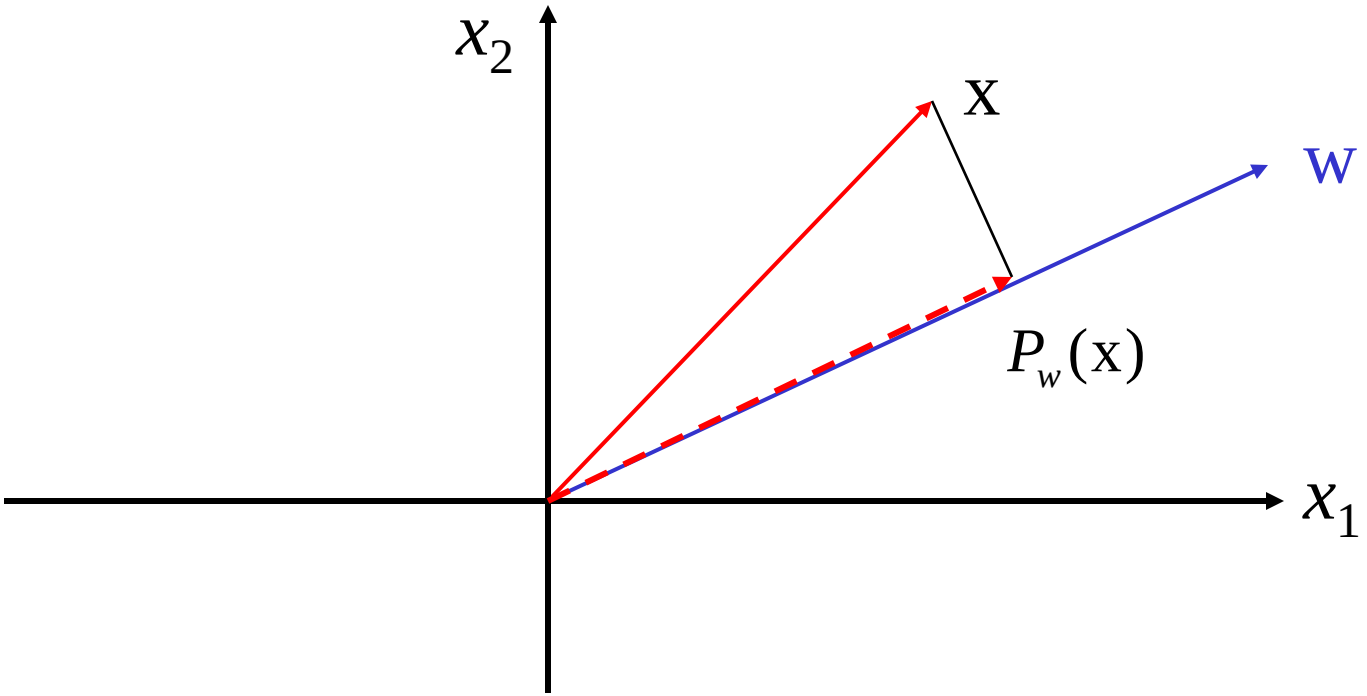
Projection of the data point **x** on to direction **w**:

$$P_w(\mathrm{x}) =< \mathrm{x}, \mathrm{w} >= \mathrm{x}' \cdot \mathrm{w}.$$

The variance of the projection on to the direction w:

$$\sigma_{\mathrm{w}}^2 = \mathrm{E}\left\{\left\|P_w(\mathrm{x})\right\|^2\right\} = \mathrm{E}\left\{\left(\mathrm{x}'\mathrm{w}\right)'\mathrm{x}'\mathrm{w}\right\} = \mathrm{E}\left\{\mathrm{w}'\mathrm{x}\mathrm{x}'\mathrm{w}\right\} =$$

$$= \mathrm{w}'\mathrm{E}\left\{\mathrm{x}\mathrm{x}'\right\}\mathrm{w} = \mathrm{w}'\mathrm{C}\mathrm{w}.$$



So, $\sigma_{\mathrm{w}}^2 = \mathrm{w}'\mathrm{C}\mathrm{w}.$

The vector w should be normalized: $\|\mathrm{w}\|^2 = 1.$

Hence, finding the normalized direction of maximal variances reduces to the following computation.

Maximizing variance: The normalized direction **w** that maximizes the variance can be found by solving the following problem:

$$\max_{w} \{\mathrm{w}'\mathrm{C}\mathrm{w}\},$$

$$\text{subject to:} \quad \|\mathrm{w}\|^2 = \mathrm{w}'\mathrm{w} = 1.$$

The constrained optimization problem is reduced to unconstrained one using method of Lagrange multipliers:

$$\max_{\mathrm{w}}\left\{\mathrm{w}'\mathrm{C}\mathrm{w}-\lambda(\mathrm{w}'\mathrm{w}-1)\right\}$$

$$\frac{\partial}{\partial \mathrm{w}}\left(\mathrm{w}'\mathrm{C}\mathrm{w}-\lambda\mathrm{w}'\mathrm{w}\right)=\mathrm{C}\mathrm{w}-\lambda\mathrm{w}$$

Condition for maximum of the function:

$$\mathrm{C}\mathrm{w}-\lambda\mathrm{w}=0.$$

We have to solve the following equation:

$$\boxed{\mathrm{C}\mathrm{w}=\lambda\mathrm{w}}$$

and find eigenvalues λ_i and eigenvectors w_i of the covariance matrix C.

$$\mathrm{C}\mathbf{w} = \lambda \mathbf{w}$$

Covariance matrix C is symmetric, so the equation has *n* distinct solutions:

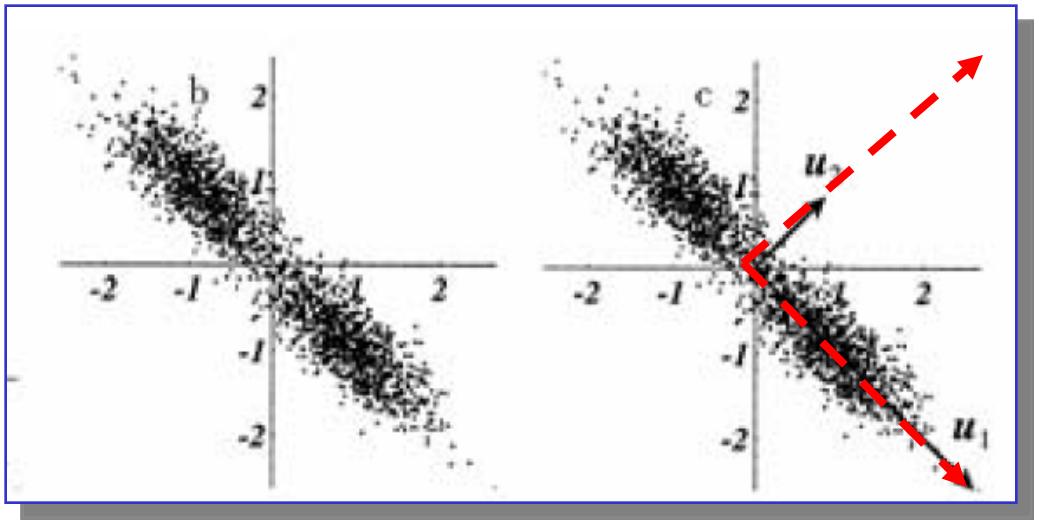
- *n* eigenvectors ($\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$) that form orthonormal basis in *n* dimensional space:

$$\mathbf{w}_i^{'}\mathbf{w}_j = \begin{cases} 1, & i = j; \\ 0 & i \neq j. \end{cases}$$

- *n* positive eigenvalues that are the data variances along the corresponding eigenvectors:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0.$$

- Direction of the maximum variance is given by the eigenvector w_1 corresponding to the largest eigenvalue λ_1 and the variance of the projection on to the direction is equal to the largest eigenvalue λ_1.
- The direction w_1 is called the first principal axes.
- The direction w_2 is called the second principal axes, and so on.



•Another names for PCA:

1) Karhunen-Loewe Transformation (KLT);

2) Hotelling Transformation.

•Properties of PCA

1) Data decorrelation;

2) Dimensionality reduction.

2) PCA dimensionality reduction:

•The objective of PCA is to perform dimensionality reduction while preserving as much of the data in high-dimensonal space as possible:

* for visualization,
* for compression,
* to cancel data containing low/no information.

.

2) PCA dimensionality reduction: Main idea

• Find the *m* first eigenvectors corresponding to the *m* largest eigenvalues.

• Project the data points into the subspace spanned on to the first *m* eigenvectors:

$$P_{\mathrm{w}_1,\ldots,\mathrm{w}_m}(\mathrm{x}) = \sum_{k=1}^{m} (\mathrm{x}'\mathrm{w}_k)\mathrm{w}_k$$

- Input data: $\mathrm{x} = \sum_{k=1}^{n} (\mathrm{x}'\mathrm{w}_k)\mathrm{w}_k$
- Data projected into the subspace spanned on to the first *m* eigenvectors: $P_{\mathrm{w}_1,\ldots,\mathrm{w}_m}(\mathrm{x}) = \sum_{k=1}^{m} (\mathrm{x}'\mathrm{w}_k)\mathrm{w}_k$
- Error caused by the dimensionality reduction:

$$\delta\mathrm{x} = \mathrm{x} - P_{\mathrm{w}_1,\ldots,\mathrm{w}_m}(\mathrm{x}) = \sum_{k=m+1}^{n} (\mathrm{x}'\mathrm{w}_k)\mathrm{w}_k$$

- Variance of the error:

$$\sigma_m^2 = \mathrm{E}\left\{\left\|\mathrm{x} - P_{\mathrm{w}_1,\ldots,\mathrm{w}_m}(\mathrm{x})\right\|^2\right\} = \sum_{k=m+1}^{n} \lambda_k .$$

• Varianc of the error is equal to the sum of eigenvalues for dropped-out dimensions:

$$\sigma_m^2 = \mathrm{E}\left\{\left\|\mathrm{x} - P_{\mathrm{w}_1,\ldots,\mathrm{w}_m}(\mathrm{x})\right\|^2\right\} = \sum_{k=m+1}^{n} \lambda_k .$$

• The PCA used to a some sense optimal representation of data in a low-dimensional subspace of the original high-dimensional pattern space. PCA provides the mimimal mean squared error among all other linear transformations.

• This subspace is spanned by the first *m* eigenvectors of covariance matrix C corresponding to *m* largest eigenvalues.

PCA on Faces.

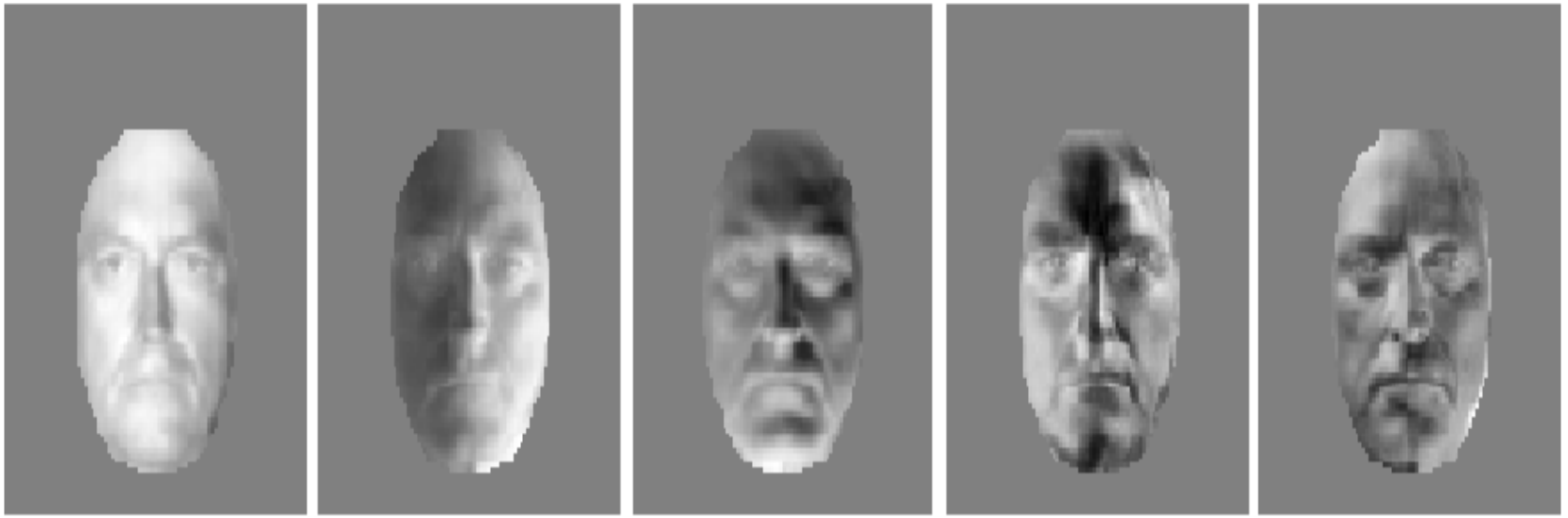


Figure 4: The eigenvectors calculated from the sparse set for the human face. Note that the images were only lit from the right so the eigenvectors are not perfectly symmetric. Observe also that the first three eigenvectors appear to be images of the face illuminated from three orthogonal lighting conditions in agreement with the orthogonal lighting conjecture.

Optimal Reconstruction

$$\min_U \|A - UU^\top A\|_F^2 \qquad subject\ to \qquad U^\top U = I$$

q=1

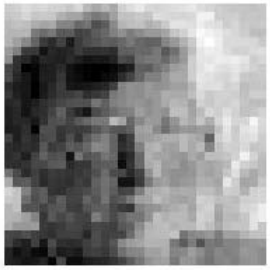
q=2

q=4

q=8

q=16

q=32



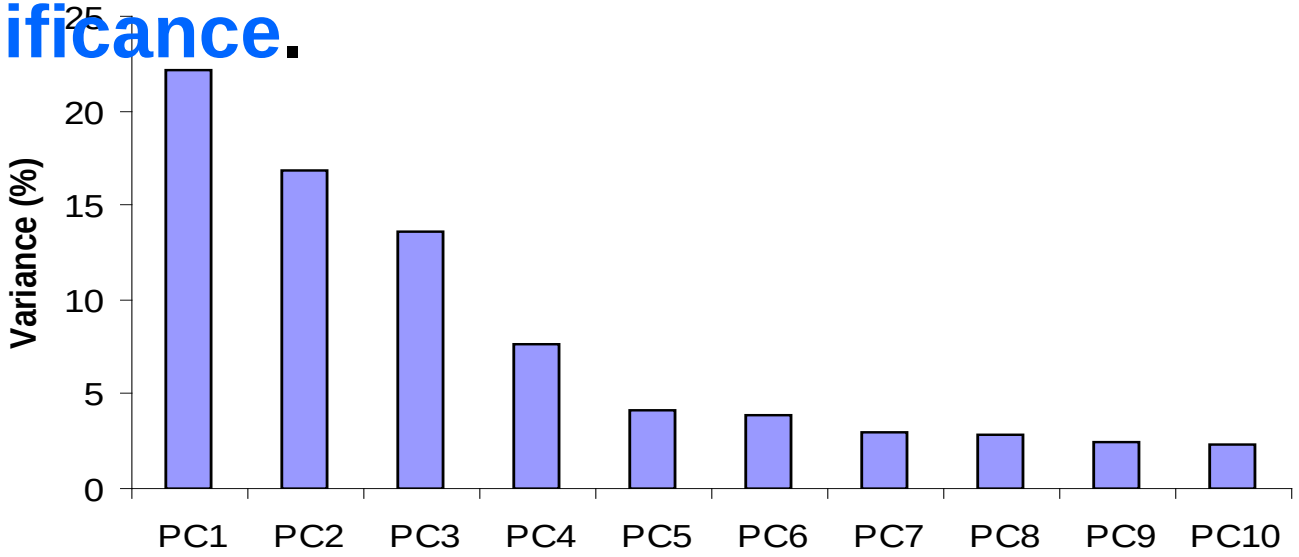
q=64

q=100…

Original Image

Dimensionality Reduction (1/2)

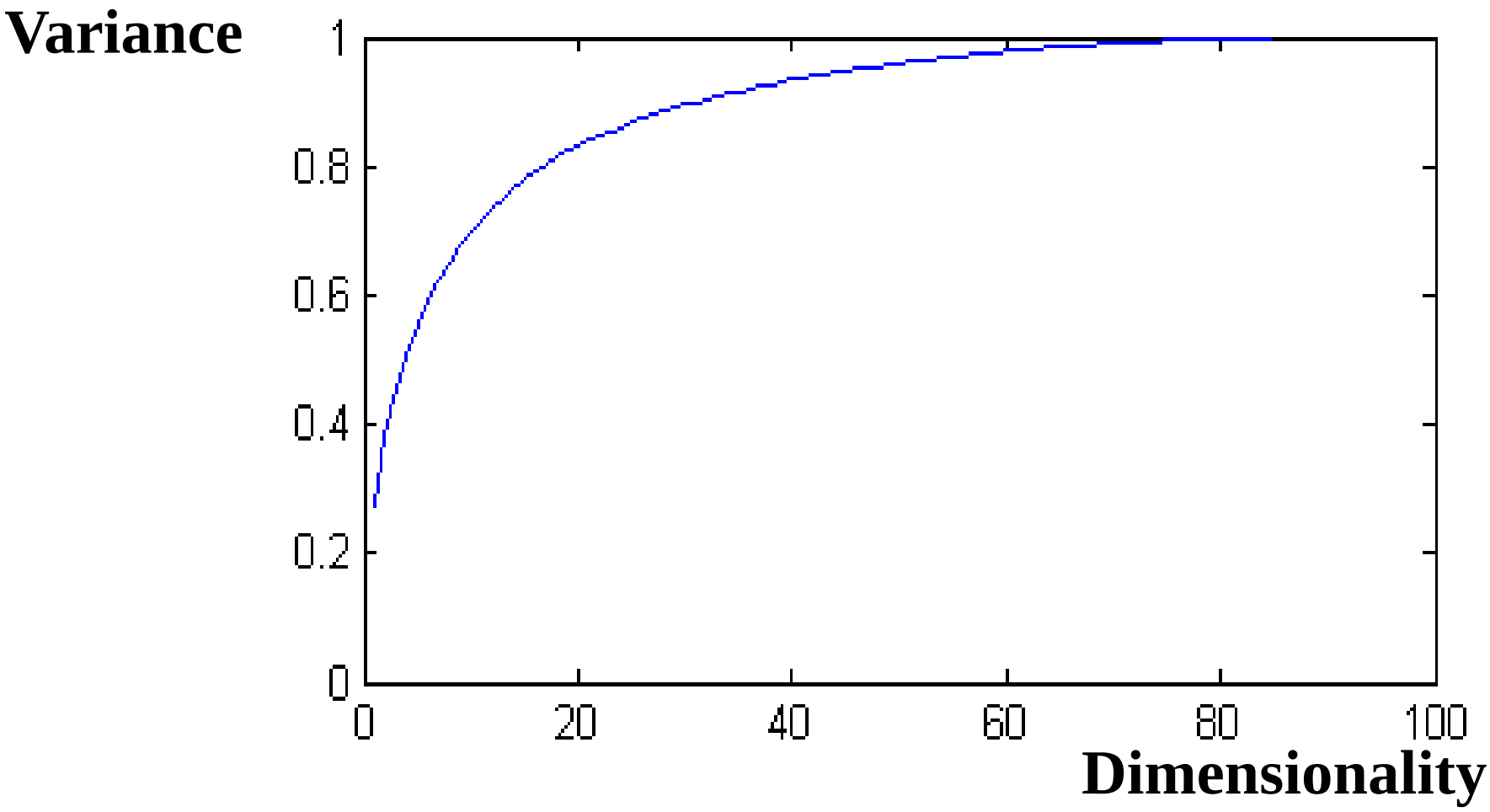
Can *ignore* the components of less significance.



You do lose some information, but if the eigenvalues are small, you don't lose much

- **n dimensions in original data**
- **calculate n eigenvectors and eigenvalues**
- **choose only the first p eigenvectors, based on their eigenvalues**

Dimensionality Reduction (2/2)



How to work in practice?

- We have training set X of size l (l n-dimensional vectors):

$$\mathrm{X} = \begin{pmatrix} x_{1,1} & \dots & x_{1,n} \\ & \dots & \\ x_{l,1} & \dots & x_{l,n} \end{pmatrix}$$

- Evaluate covariance matrix using the training set X:

$$\mathrm{C} = \frac{1}{l}\mathrm{X}'\mathrm{X}, \qquad \text{where} \quad c_{j,k} = \frac{1}{l}\sum_{i=1}^{l} x_{i,j}x_{i,k}.$$

- Find eigenvalues and eigenvectors for the covariance matrix.

Principal Component Analysis (PCA)

takes an initial subset of the principal axes of the training data and project the data (both training and test) into the space spanned by this set of eigenvectors.

•The data is projected onto subspace spanned by *m* first eigenvectors of covariance matrix. The new coordinates are known as principal coordinates with eigenvectors referred as principal axes.

Algorithm:

Input: Dataset $X=\{x_1, x_2, \ldots, x_l\} \subseteq \Re^n$,

---

Process:

$$\mu = \frac{1}{l}\sum_{i=1}^{l} x_i$$

$$C = \frac{1}{l}\sum_{i=1}^{l} (x_i - \mu)(x_i - \mu)'$$

$$[W, \Lambda] = \text{eig}(lC)$$

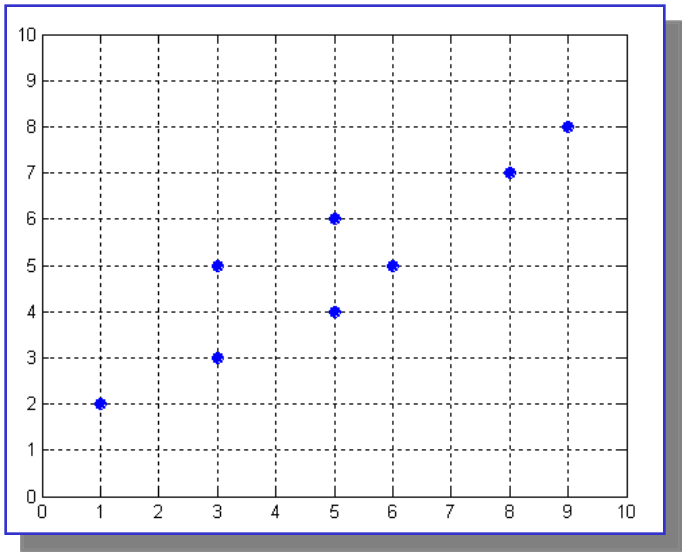
$$\tilde{x}_i = W \cdot x_i, \quad i = 1,2,...,l.$$

Output: transformed data $\tilde{S} = \{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_l\}$

$$\tilde{x} = \{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_k\}$$

Example: 8 vectors in 2-D space

X=[1,2; 3,3; 3,5; 5,4; 5,6; 6,5; 8,7; 9,8];



•Mean values:

$$\bar{x} = \frac{1}{8}(1+3+3+5+5+6+8+9) = \frac{40}{8} = 5$$

• Centered data: [-4,-3; -2,-2; -2,0; 0,-1; 0,1; 1,0; 3,2; 4,3]

• Covariance matrix C:

$$c_{1,1} = \frac{1}{8}((-4)(-4)+(-2)(-2)+(-2)(-2)+0\cdot 0+0\cdot 0+1\cdot 1+3\cdot 3+4\cdot 4) = \frac{50}{8} = 6.25$$

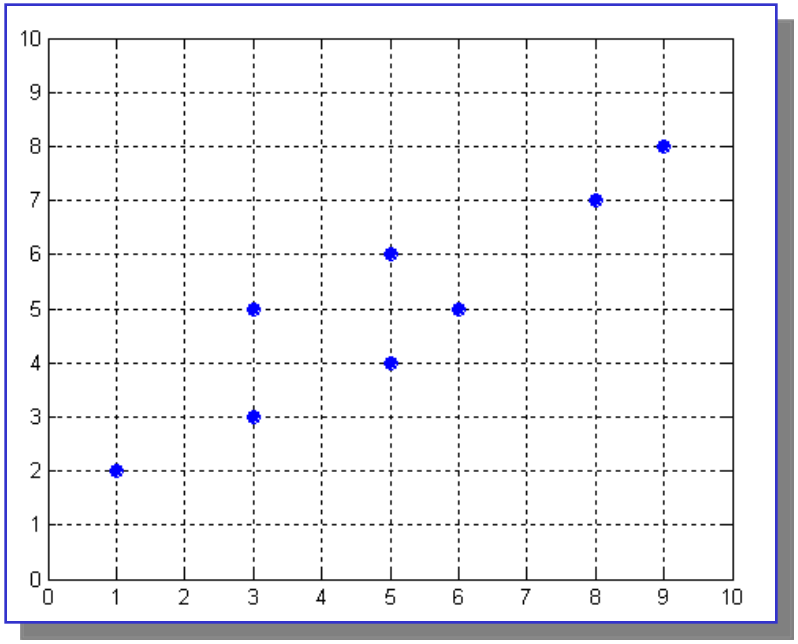
$$c_{1,2} = c_{2,1} = \frac{1}{8}((-4)(-3)+(-2)(-2)+(-2)\cdot 0+0\cdot(-1)+0\cdot 1+1\cdot 0+3\cdot 2+4\cdot 3) = \frac{34}{8} = 4.25$$

$$c_{2,2} = \frac{1}{8}((-3)(-3)+(-2)(-2)+0\cdot 0+(-1)(-1)+1\cdot 1+0\cdot 0+2\cdot 2+3\cdot 3) = \frac{28}{8} = 3.5$$

$$C = \begin{pmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{pmatrix}$$

Example: 8 vectors in 2-D space

X=[1,2; 3,3; 3,5; 5,4; 5,6; 6,5; 8,7; 9,8];



Covariance matrix C:

$$C = \begin{pmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{pmatrix}$$

$$Cw = \lambda w$$

Find eigenvalues and eigenvectors of the covariance matrix C:

$$\begin{pmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{pmatrix} \cdot \begin{pmatrix} w_{x_1} \\ w_{x_2} \end{pmatrix} = \lambda \begin{pmatrix} w_{x_1} \\ w_{x_2} \end{pmatrix}$$

1) Find eigenvalues

$$C \cdot w = \lambda w$$

$$\left(C - \lambda \mathrm{I}_n\right) w = 0$$

Solve the characteristic equation:

$$\det\left(C - \lambda \mathrm{I}_n\right) = 0$$

$$\det\begin{pmatrix} 6.25 - \lambda & 4.25 \\ 4.25 & 3.5 - \lambda \end{pmatrix} = 0$$

$$(6.25 - \lambda)(3.5 - \lambda) - 4.25 \cdot 4.25 = 0$$

Eigenvalues: $\lambda_1 = 9.34$

$\lambda_2 = 0.41$

2) Find eigenvectors for the eigenvalues:

a) $$\begin{pmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{pmatrix} \cdot \begin{pmatrix} w_{11} \\ w_{12} \end{pmatrix} = \begin{pmatrix} 9.34 \cdot w_{11} \\ 9.34 \cdot w_{12} \end{pmatrix} \Rightarrow w_{11} = 1.376 \cdot w_{11}$$

b) $$\begin{pmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{pmatrix} \cdot \begin{pmatrix} w_{21} \\ w_{22} \end{pmatrix} = \begin{pmatrix} 0.41 \cdot w_{21} \\ 0.41 \cdot w_{22} \end{pmatrix} \Rightarrow w_{21} = -1.376 \cdot w_{22}$$

Normalized the eigenvectors

$$w_{11}^2 + w_{12}^2 = 1 \Rightarrow (1.376)^2 \cdot w_{12}^2 + w_{12}^2 = 1 \Rightarrow \begin{pmatrix} w_{11} \\ w_{12} \end{pmatrix} = \begin{pmatrix} 0.81 \\ 0.59 \end{pmatrix}$$

$$w_{21}^2 + w_{22}^2 = 1 \Rightarrow (1.376)^2 \cdot w_{22}^2 + w_{22}^2 = 1 \Rightarrow \begin{pmatrix} w_{21} \\ w_{22} \end{pmatrix} = \begin{pmatrix} -0.59 \\ 0.81 \end{pmatrix}$$

Eigenvectors: $\begin{pmatrix} w_{11} \\ w_{12} \end{pmatrix} = \begin{pmatrix} 0.81 \\ 0.59 \end{pmatrix}$

$\begin{pmatrix} w_{21} \\ w_{22} \end{pmatrix} = \begin{pmatrix} -0.59 \\ 0.81 \end{pmatrix}$

Check orthogonality: $w_1' w_2 = \begin{pmatrix} 0.81 & 0.59 \end{pmatrix} \begin{pmatrix} -0.59 \\ 0.81 \end{pmatrix} = 0$

Check normalization: $w_1' w_1 = \begin{pmatrix} 0.81 & 0.59 \end{pmatrix} \begin{pmatrix} 0.81 \\ 0.59 \end{pmatrix} = 1$

$w_2' w_2 = \begin{pmatrix} -0.59 & 0.81 \end{pmatrix} \begin{pmatrix} -0.59 \\ 0.81 \end{pmatrix} = 1$

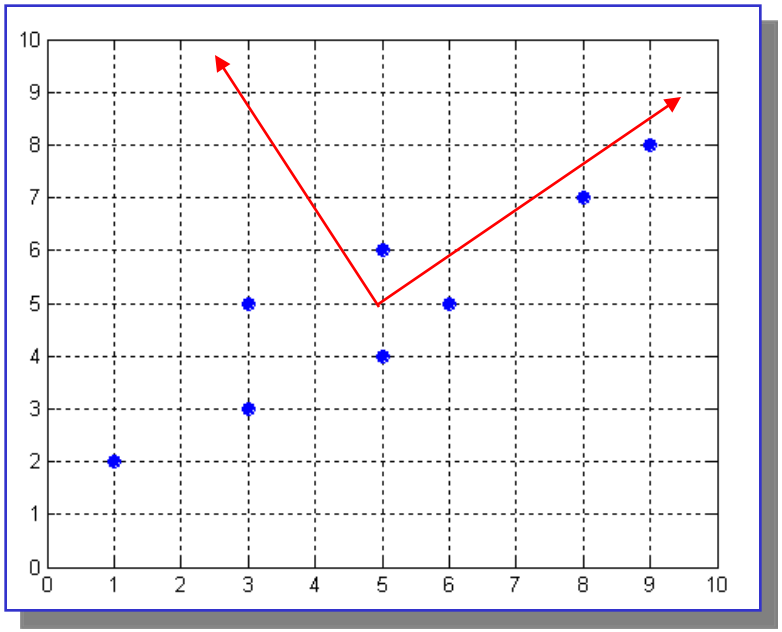
Orthonormal basis: $W = \begin{pmatrix} w_1' \\ \dots \\ w_k' \end{pmatrix}$

Transfromation (projection) into new basis: $\tilde{x} = Wx$

$$\begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} = \begin{pmatrix} w_1' \\ w_2' \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Example:

$$\begin{pmatrix} 0.81 & 0.59 \\ -0.59 & 0.81 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 5 \\ -0.6 \end{pmatrix}$$

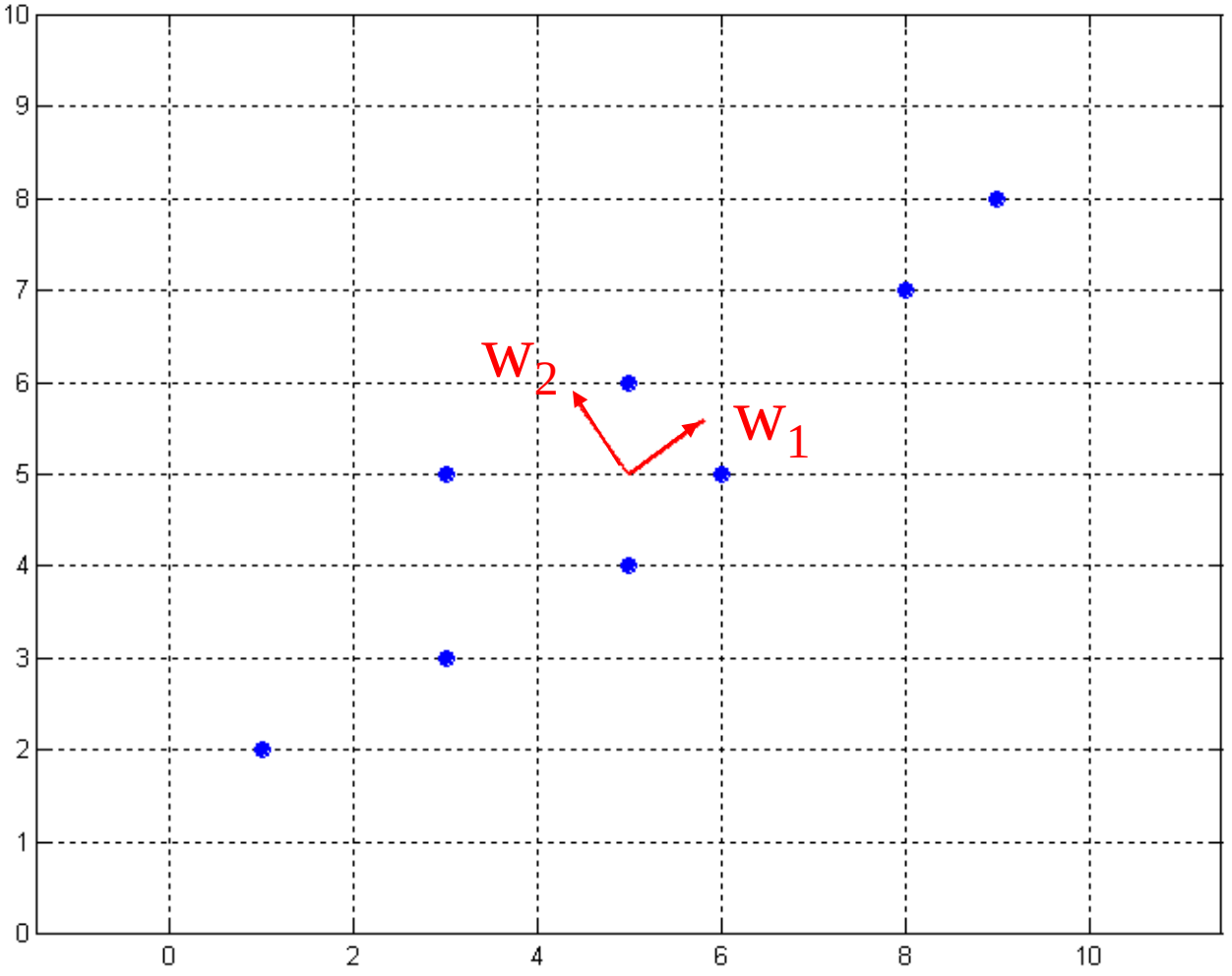


$$w_1 = \begin{pmatrix} 0.8086 \\ 0.5883 \end{pmatrix}$$

$$w_2 = \begin{pmatrix} -0.5883 \\ 0.8086 \end{pmatrix}$$

$$\lambda_1 = 9.34$$

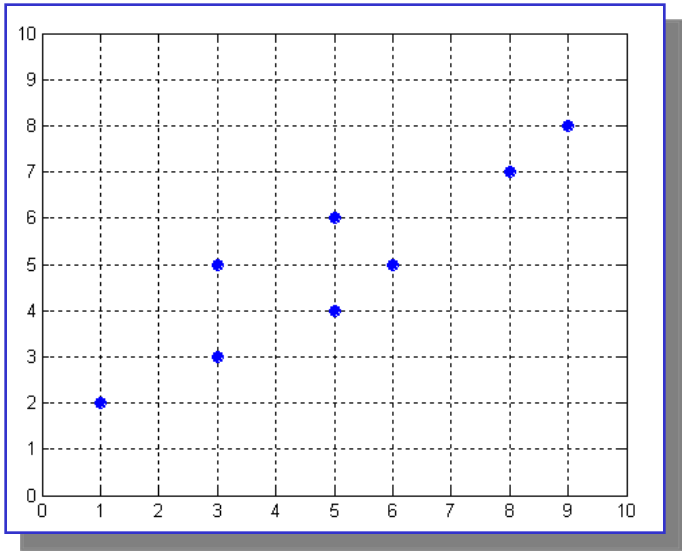
$$\lambda_2 = 0.41$$



The principal axes for the test data.

The example with Matlab

```
X=[1,2; 3,3; 3,5; 5,4; 5,6; 6,5; 8,7; 9,8];
X1=X(:,1);
X2=X(:,2);
X1=X1-mean(X1); % Centered data
X2=X2-mean(X2); % Centered data
C=cov(X1,X2);  eval C;
```



$$C = \begin{pmatrix} 7.1429 & 4.8571 \\ 4.8571 & 4.0 \end{pmatrix}$$

% Divided by (l-1), not l!

```
[W,Lambda]=eig(C);
```

$$\text{Lambda} = \begin{pmatrix} 0.4664 & 0 \\ 0 & 10.6764 \end{pmatrix} \qquad W = \begin{pmatrix} 0.5883 & -0.8086 \\ -0.8086 & -0.5883 \end{pmatrix}$$

Eigenvectors do not depend on scaling of covariance matrix, eigenvalues do.

Derivation of PCs

Assume that

$$E[\mathbf{x}] = \mathbf{0} \qquad a = \mathbf{x}^T\mathbf{q} = \mathbf{q}^T\mathbf{x} \qquad \|\mathbf{q}\| = (\mathbf{q}^T\mathbf{q})^{1/2} = 1$$

$$\Rightarrow \sigma^2 = E[a^2] - E[a]^2 = E[a^2]$$

$$= E[(\mathbf{q}^T\mathbf{x})(\mathbf{x}^T\mathbf{q})] = \mathbf{q}^T E[\mathbf{x}\mathbf{x}^T]\mathbf{q} = \mathbf{q}^T\mathbf{R}\mathbf{q}$$

Find q's maximizing this!!

Principal component q can be obtained by Eigenvector decomposition such as SVD!

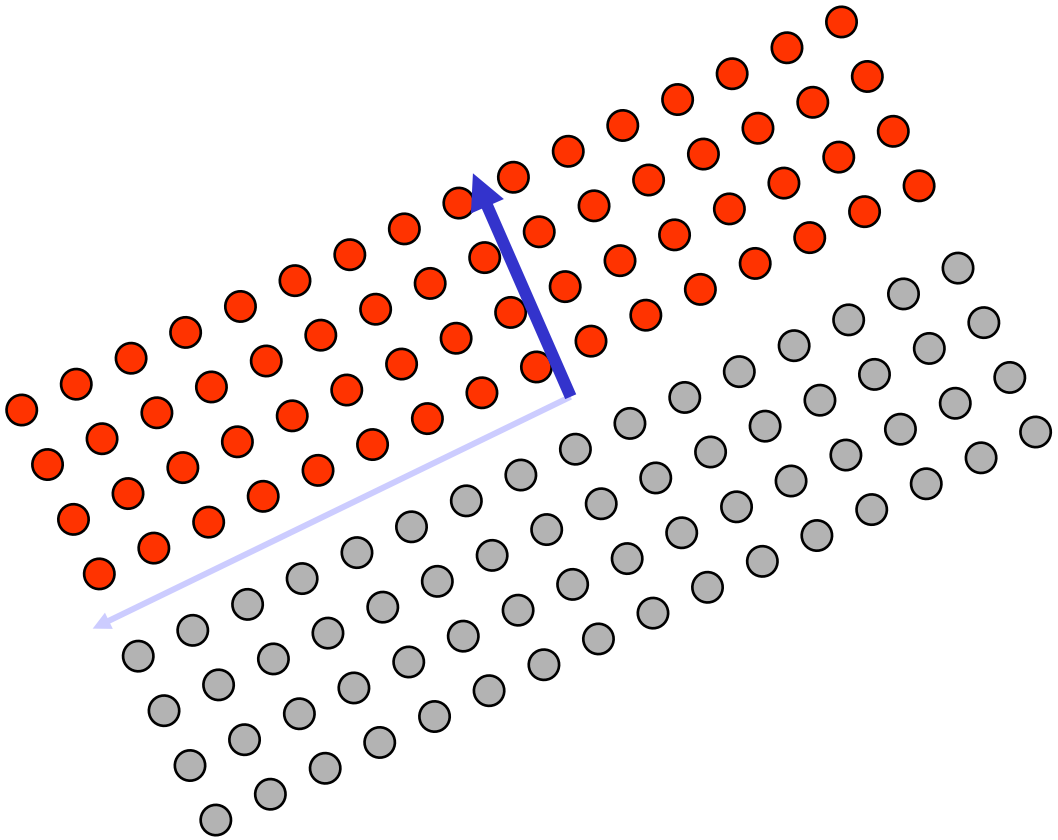
$$\mathbf{R} = \mathbf{Q}\Lambda\mathbf{Q}^T, \quad \mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_j, \ldots, \mathbf{q}_m], \quad \Lambda = diag[\lambda_1, \lambda_2, \ldots, \lambda_j, \ldots, \lambda_m]$$

$$\Leftrightarrow \mathbf{R}\mathbf{q}_j = \lambda_j\mathbf{q}_j \qquad j = 1, 2, \ldots, m$$

$$\Rightarrow \mathbf{R}\mathbf{q} = \lambda\mathbf{q}$$

Limitations of PCA

Are the maximal variance dimensions the relevant dimensions for preservation?



Linear Discriminant Analysis (1/6)

- **What is the goal of LDA?**
  - **Perform dimensionality reduction “while preserving as much of the class discriminatory information as possible”.**
  - **Seeks to find directions along which the classes are best separated.**
  - **Takes into consideration the scatter *within-classes* but also the scatter *between-classes*.**
  - **For example of face recognition, more capable of distinguishing image variation due to identity from variation due to other sources such as illumination and expression.**

Linear Discriminant Analysis (2/6)

Within-class scatter matrix $S_w = \sum_{i=1}^{c} \sum_{j=1}^{n_i} (Y_j - M_i)(Y_j - M_i)^T$

Between-class scatter matrix $S_b = \sum_{i=1}^{c} (M_i - M)(M_i - M)^T$

projection matrix

$$\mathbf{y} = U^T \mathbf{x}$$

- **LDA computes a transformation that maximizes the between-class scatter while minimizing the within-class scatter:**

$$\max \frac{|\tilde{S}_b|}{|\tilde{S}_w|} = \max \frac{|U^T S_b U|}{|U^T S_w U|}$$

products of eigenvalues !

$$S_w^{-1} S_b = U \Lambda U^T$$

$\tilde{S}_b, \tilde{S}_w$ **: scatter matrices of the projected data** $\mathbf{y}$

Linear Discriminant Analysis (3/6)

- **Does S_w^{-1} always exist?**
  - **If S_w is non-singular, we can obtain a conventional eigenvalue problem by writing:**

$$S_w^{-1} S_b = U \Lambda U^T$$

  - **In practice, S_w is often singular since the data are image vectors with large dimensionality while the size of the data set is much smaller ($M << N$)**

  - **c.f. Since S_b has at most rank C-1, the max number of eigenvectors with non-zero eigenvalues is C-1 (i.e., max dimensionality of sub-space is C-1)**

Linear Discriminant Analysis (4/6)

- **Does S_w^{-1} always exist? .**
  - **To alleviate this problem, we can use PCA first:**

1) **PCA is first applied to the data set to reduce its dimensionality.**

$$\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} \text{--->} PCA \text{--->} \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_K \end{bmatrix}$$

2) **LDA is then applied to find the most discriminative directions:**

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_K \end{bmatrix} \text{--->} LDA \text{--->} \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_{C-1} \end{bmatrix}$$