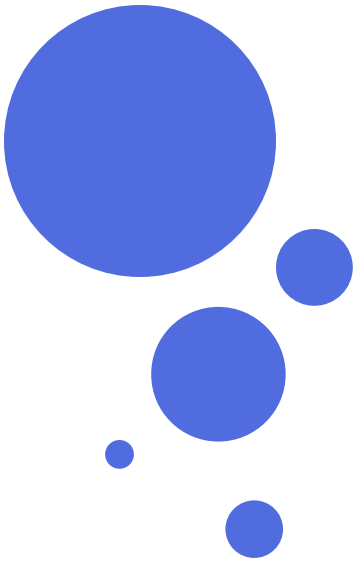
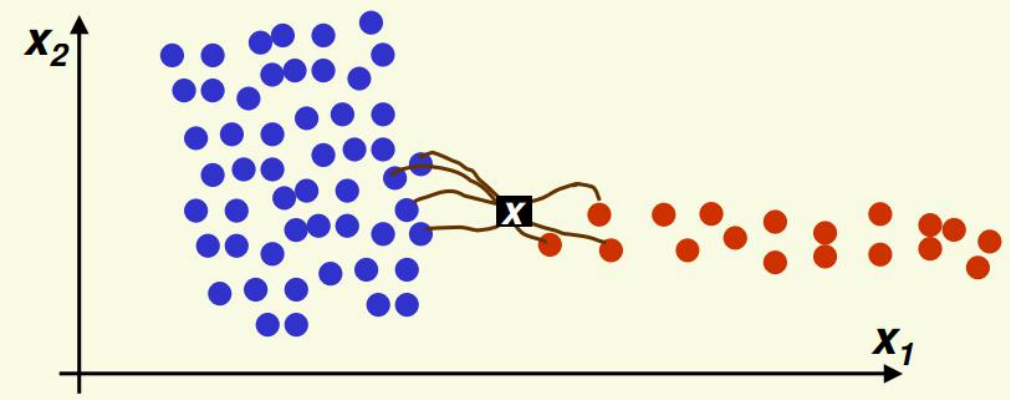
Rensselaer

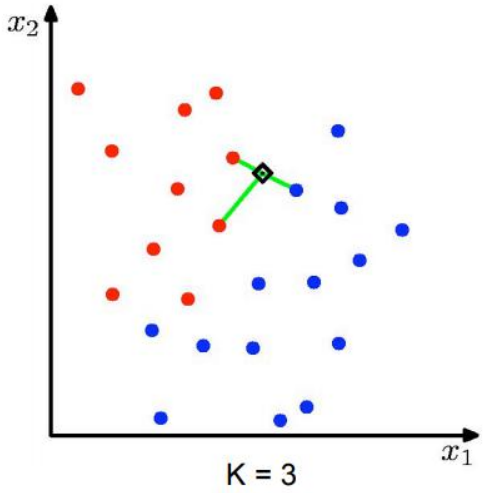


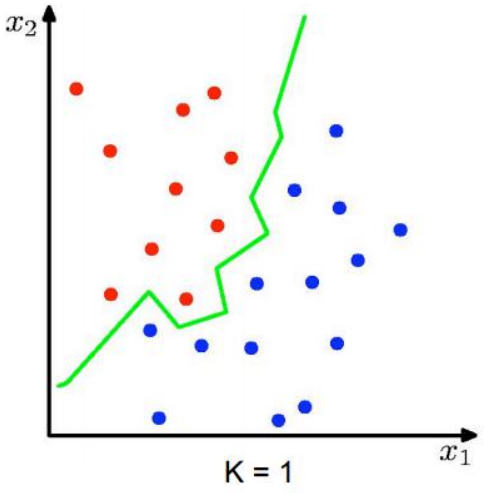
Lecture 8: Dimensionality – PCA and Fisher Linear Discrimination

Dr. Chengjiang Long
Computer Vision Researcher at Kitware Inc.
Adjunct Professor at RPI.
Email: **longc3@rpi.edu**

Recap Previous Lecture







Outline

- Principal component analysis (PCA)
- Application examples with PCA
- Fisher Linear Discriminant
- Multiple Discriminant Analysis

Outline

- **Principal component analysis (PCA)**
- Application examples with PCA
- Fisher Linear Discriminant
- Multiple Discriminant Analysis

PCA: Overview

- Principal component analysis (PCA) is a way to reduce data dimensionality
- PCA projects high dimensional data to a lower dimension
- PCA projects the data in the least square sense– it captures big (principal) variability in the data and ignores small variability

PCA: An Intuitive Approach

- Let us say we have $\mathbf{x}_i$, i=1…N data points in p dimensions (p is large). If we want to represent the data set by a single point $\mathbf{x}_0$, then

$$\mathbf{x}_0 = \mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$$

Sample mean

Can we justify this choice mathematically?

$$J_0(\mathbf{x}_0) = \sum_{i=1}^{N}\left\|\mathbf{x}_i - \mathbf{x}_0\right\|^2$$

It turns out that if you minimize J_0, you get the above solution with sample mean

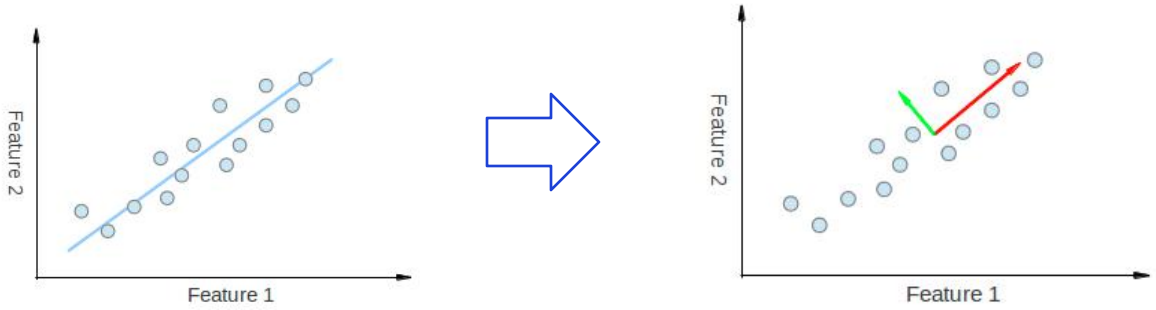
PCA: An Intuitive Approach…

- Representing the data set $\mathbf{x}_i$, i=1…N by its mean is quite uninformative. So let's try to represent the data by a straight line of the form:

$$\mathbf{x} = \mathbf{m} + a\mathbf{e}$$

- This is equation of a straight line that says that it passes through **m, e** is a unit vector along the straight line. And the signed distance of a point **x** from **m** is a
- The training points projected on this straight line would be

$$\mathbf{x}_i = \mathbf{m} + a_i\mathbf{e}, \quad i = 1 \ldots N$$



PCA: An Intuitive Approach…

- Let's now determine a_i's

$$J_1(a_1, a_2, \ldots, a_N, \mathbf{e}) = \sum_{i=1}^{N} \left\| \mathbf{m} + a_i \mathbf{e} - \mathbf{x}_i \right\|^2$$

$$= \sum_{i=1}^{N} a_i^2 \, \| \mathbf{e} \|^2 - 2 \sum_{i=1}^{N} a_i \mathbf{e}^T (\mathbf{x}_i - \mathbf{m}) + \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{m} \|^2$$

$$= \sum_{i=1}^{N} a_i^2 - 2 \sum_{i=1}^{N} a_i \mathbf{e}^T (\mathbf{x}_i - \mathbf{m}) + \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{m} \|^2$$

Partially differentiating with respect to a_i we get: $a_i = \mathbf{e}^T (\mathbf{x}_i - \mathbf{m})$

Plugging in this expression for a_i in J_1 we get:

$$J_1(\mathbf{e}) = -\sum_{i=1}^{N} \mathbf{e}^T (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \mathbf{e} + \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{m} \|^2 = -\mathbf{e}^T S \mathbf{e} + \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{m} \|^2$$

where $S = \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$ is called the **scatter matrix**

PCA: An Intuitive Approach…

$$J_1(\mathbf{e}) = -\sum_{i=1}^{N} \mathbf{e}^T(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T\mathbf{e} + \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{m} \|^2 = -\mathbf{e}^T S\mathbf{e} + \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{m} \|^2$$

So minimizing J_1 is equivalent to maximizing: $\mathbf{e}^T S\mathbf{e}$

Subject to the constraint that **e** is a unit vector: $\mathbf{e}^T\mathbf{e} = 1$

Use Lagrange multiplier method to form the objective function:

$$\mathbf{e}^T S\mathbf{e} - \lambda(\mathbf{e}^T\mathbf{e} - 1)$$

Differentiate to obtain the equation: $2S\mathbf{e} - 2\lambda\mathbf{e} = \mathbf{0}$ *or* $\mathbf{Se} = \lambda\mathbf{e}$

Solution is that **e** is the eigenvector of S corresponding to the largest eigenvalue.

PCA: An Intuitive Approach…

- The preceding analysis can be extended in the following way. Instead of projecting the data points on to a straight line, we may now want to project them on a d-dimensional plane of the form:

$$\mathbf{x} = \mathbf{m} + a_1\mathbf{e}_1 + \cdots + a_d\mathbf{e}_d$$

d is much smaller than the original dimension *p*

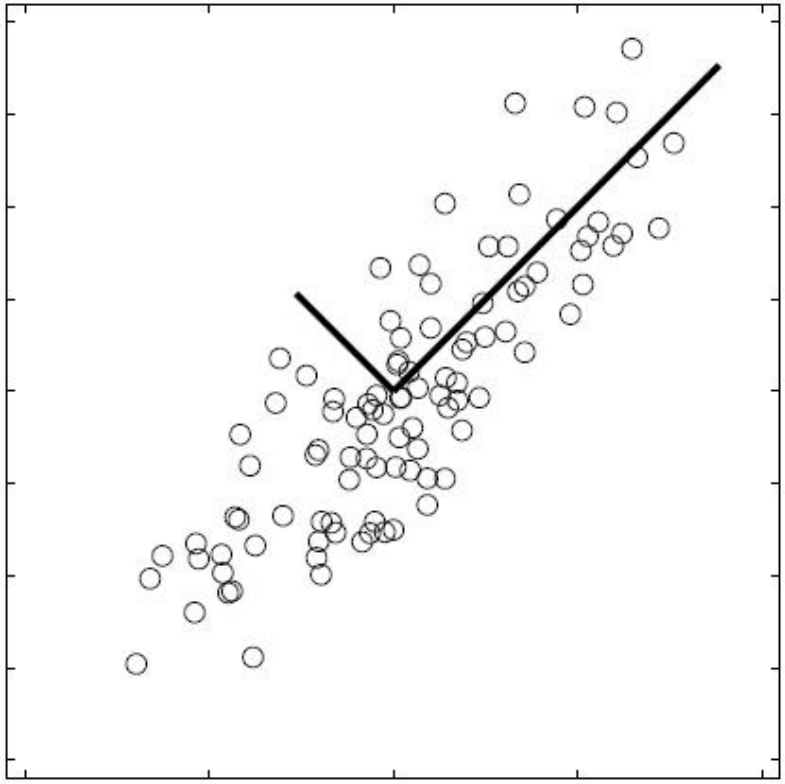
- In this case one can form the objective function:

$$J_d = \sum_{i=1}^{N} \| (\mathbf{m} + \sum_{k=1}^{d} a_{ik}\mathbf{e}_k) - \mathbf{x}_i \|^2$$

It can also be shown that the vectors $\mathbf{e}_1$, $\mathbf{e}_2$, …, $\mathbf{e}_d$ are *d* eigenvectors corresponding to *d* largest eigen values of the scatter matrix

$$S = \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$

PCA: Visualization



Data points are represented in a rotated orthogonal coordinate system: the origin is the mean of the data points and the axes are provided by the eigenvectors.

Computation of PCA

- In practice we compute PCA via SVD (singular value decomposition)
- Form the centered data matrix:

$$X_{p,N} = \left[(\mathbf{x}_1 - \mathbf{m}) \ldots\ldots (\mathbf{x}_N - \mathbf{m})\right]$$

- Compute its SVD:

$$X_{p,N} = U_{p,p} D_{p,p} (V_{N,p})^T$$

U and *V* are orthogonal matrices, *D* is a diagonal matrix

Computation of PCA…

- Note that the scatter matrix can be written as:

$$S = XX^T = UD^2U^T$$

- So the eigenvectors of *S* are the columns of *U* and the eigenvalues are the diagonal elements of D^2
- Take only a few significant eigenvalue eigenvector pairs *d* << *p*; The new reduced dimension representation becomes:

$$\tilde{\mathbf{x}}_i = \mathbf{m} + U_{p,d}(U_{p,d})^T(\mathbf{x}_i - \mathbf{m})$$

Usually we used the features with reduced dimensions to fit the classification models.

Computation of PCA…

- Sometimes we are given only a few high dimensional data points, *i.e.*, $p \geqslant N$
- In such cases compute the SVD of X^T:

$$X^T = V_{N,N} D_{N,N} (U_{p,N})^T$$

So we get:

$$X = U_{p,N} D_{N,N} (V_{N,N})^T$$

Then, proceed as before, choose only $d < N$ significant eigenvalues for data representation:

$$\tilde{\mathbf{x}}_i = \mathbf{m} + U_{p,d} (U_{p,d})^T (\mathbf{x}_i - \mathbf{m})$$

PCA: A Gaussian Viewpoint

$$\mathbf{x} \sim \frac{1}{\sqrt{(2\pi)^p \mid \Sigma \mid}} \exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})) = \prod_{i=1}^{p} \frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{(\mathbf{u}_i^T(\mathbf{x}-\boldsymbol{\mu}))^2}{2\sigma_i^2}),$$

- where the covariance matrix Σ is estimated from the scatter matrix as (1/*N*)*S*. **u**'s and **σ**'s are respectively eigenvectors and eigenvalues of *S*.
- If *p* is large, then we need a even larger number of data points to estimate the covariance matrix. So, when a limited number of training data points is available the estimation of the covariance matrix goes quite wrong. This is known as curse of dimensionality in this context.
- To combat curse of dimensionality, we discard smaller eigenvalues and be content with:

$$\mathbf{x} \sim \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{(\mathbf{u}_i^T(\mathbf{x}-\boldsymbol{\mu}))^2}{2\sigma_i^2}), \text{ where } d < \min(p, N)$$

Outline

- Principal component analysis (PCA)
- **Application examples with PCA**
- Fisher Linear Discriminant
- Multiple Discriminant Analysis

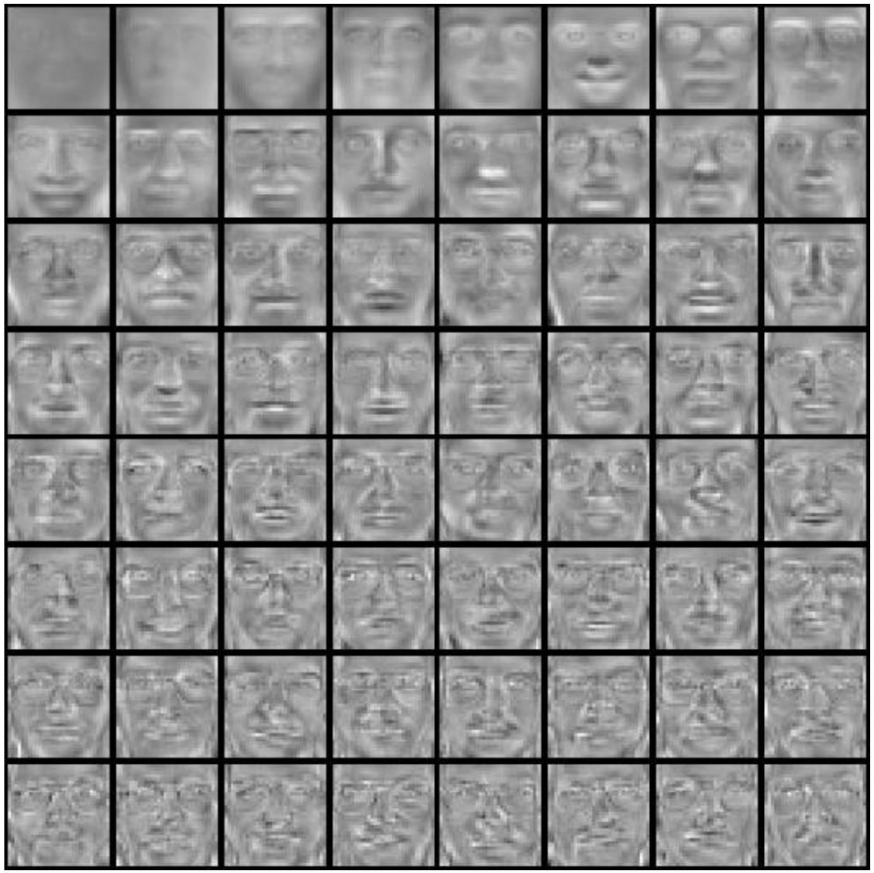
Eigenface

- When viewed as vectors of pixels, face images are extremely high dimensional
  - 100x100 image $= 10{,}000$ dimensions
  - Slow and lots of storage
- But very few $10{,}000$ dimensional vectors are valid face images
- We want to effectively model the subspace of face images

Eigenface

Mean: μ

Top eigenvectors: $u_1,\ldots u_k$



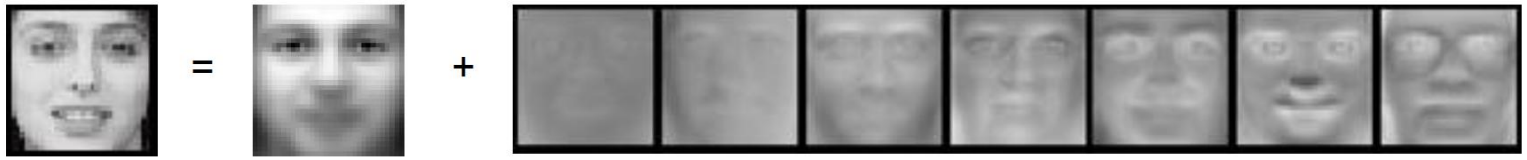
slide by Derek Hoiem

Eigenface: Representation

- Face **x** in "face space" coordinates:

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x}-\mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x}-\mu)]$$
$$= \quad w_1, \ldots, w_k$$

- Reconstruction:



$$\hat{x} = \mu + w_1 u_1 + w_2 u_2 + w_3 u_3 + w_4 u_4 + \ldots$$

slide by Derek Hoiem

Eigenface: Reconstruction

After computing eigenfaces using 400 face images from ORL face database

slide by Derek Hoiem

Image compression

Original Image

- Divide the original $372 \text{x} 492$ image into patches:
  Each patch is an instance that contains $12 \text{x} 12$ pixels on a grid
- View each as a $144-$D vector

Image compression: 144D -> 60D

Image compression: 144D -> 16D

Image compression: 16 most important eigenvectors

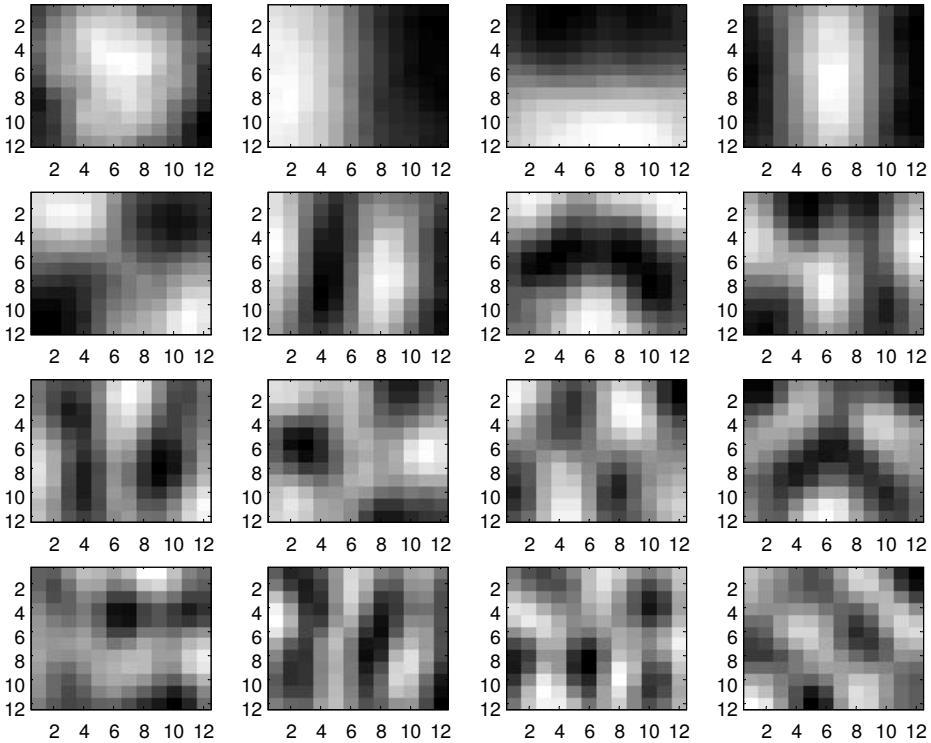


Image compression: 144D -> 6D

Image compression: 6 most important eigenvectors

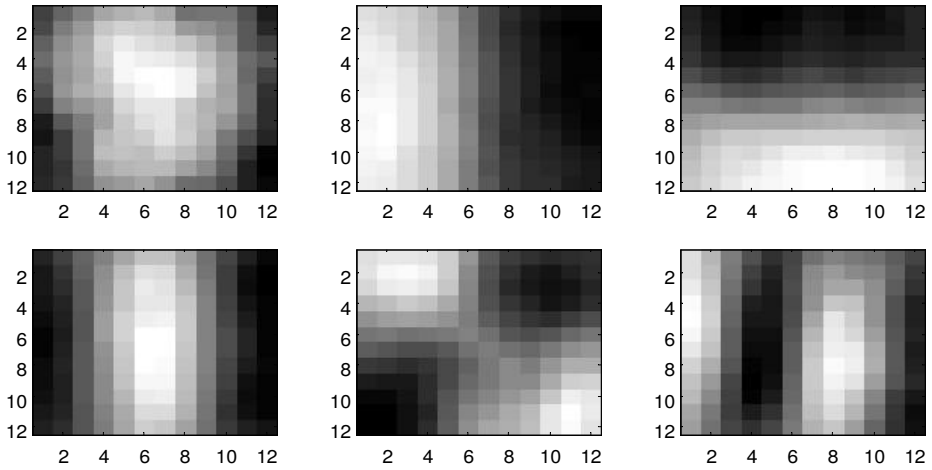


Image compression: 144D -> 3D

Image compression: 3 most important eigenvectors

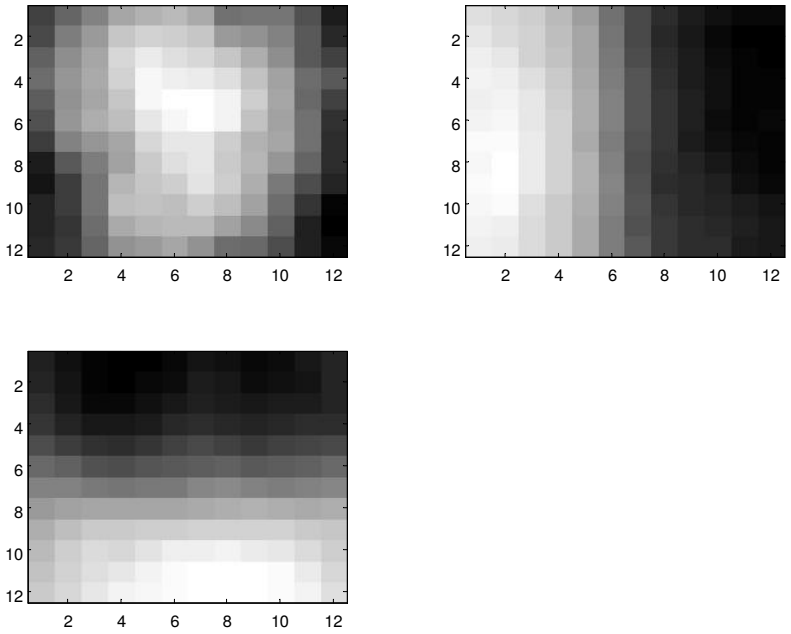
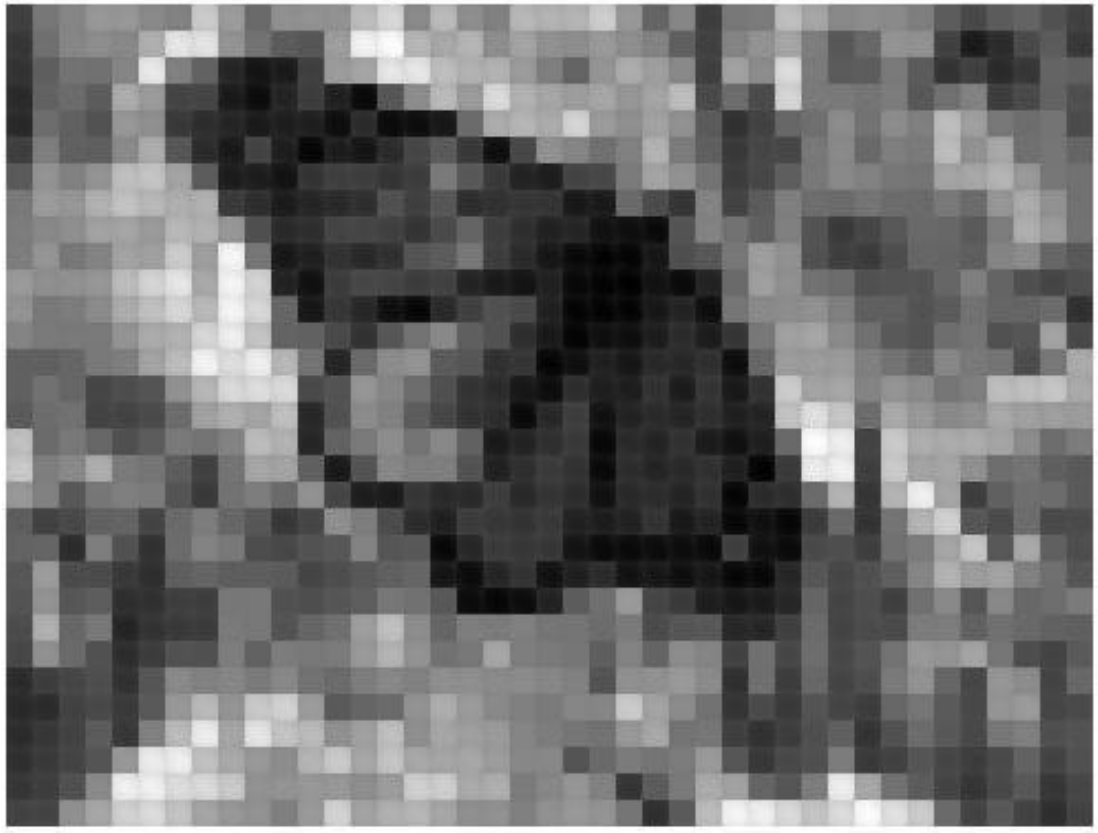


Image compression: 144D -> 1D



Outline

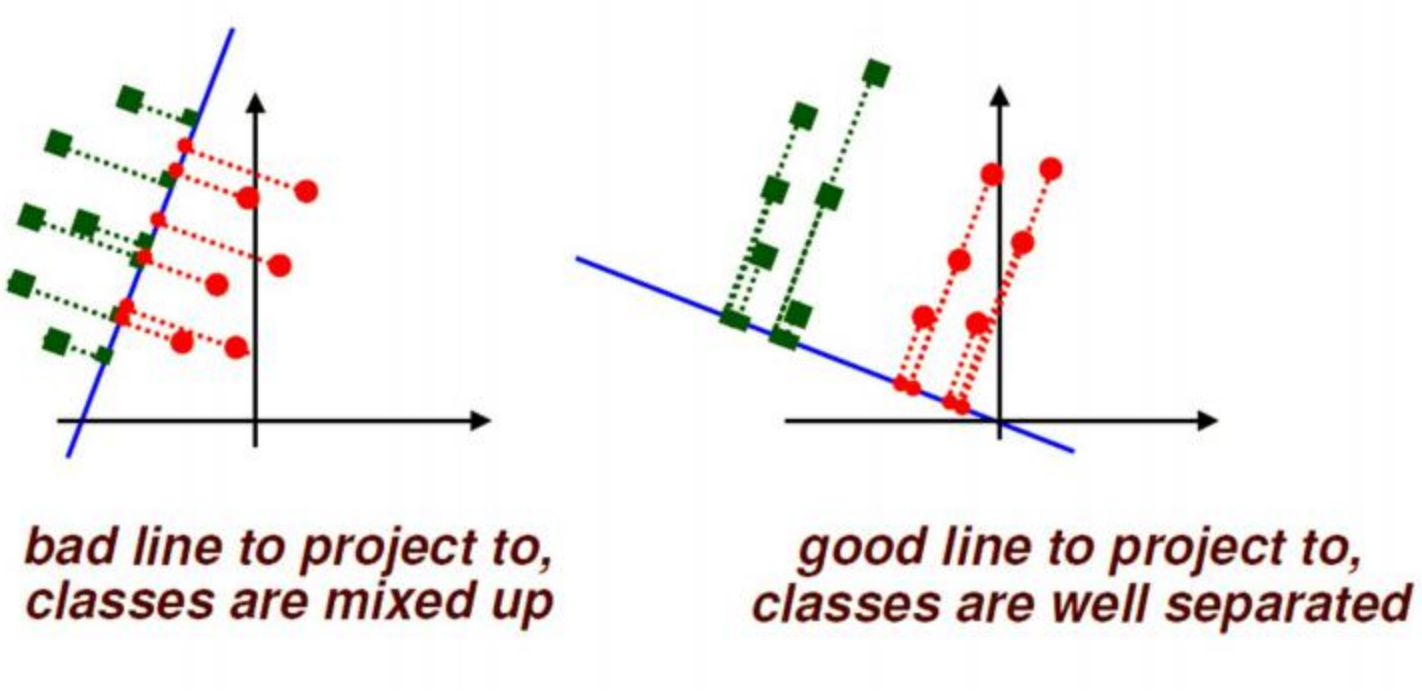
- Principal component analysis (PCA)
- Application examples with PCA
- **Fisher Linear Discriminant**
- Multiple Discriminant Analysis

Data Representation vs. Data Classification

- Fisher Linear Discriminant: project to a line which preserves direction useful for data classification

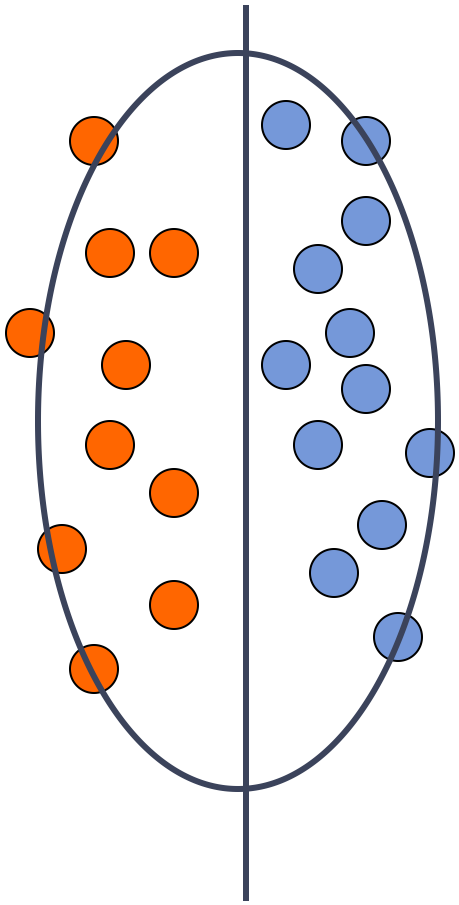
Fisher Linear Discriminant

- Main idea: find projection to a line such that samples from different classes are well separated



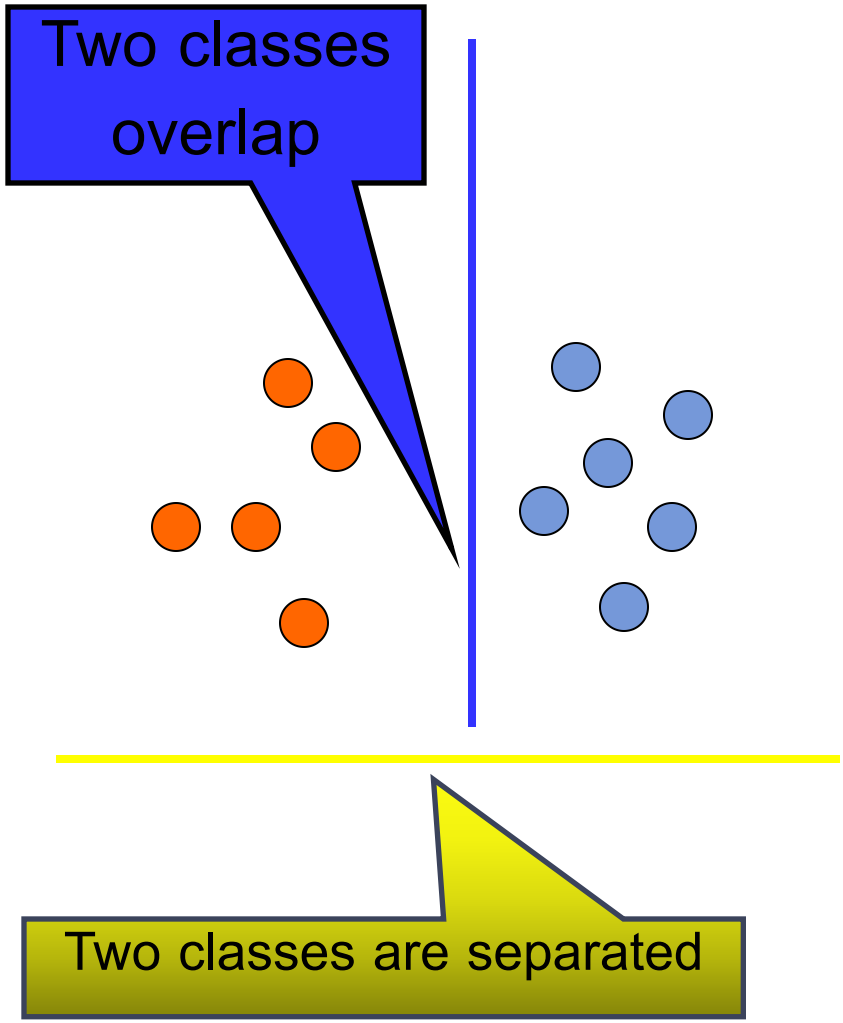
Is PCA a good criterion for classification?

- Data variation determines the projection direction
- What's missing?
  - Class information



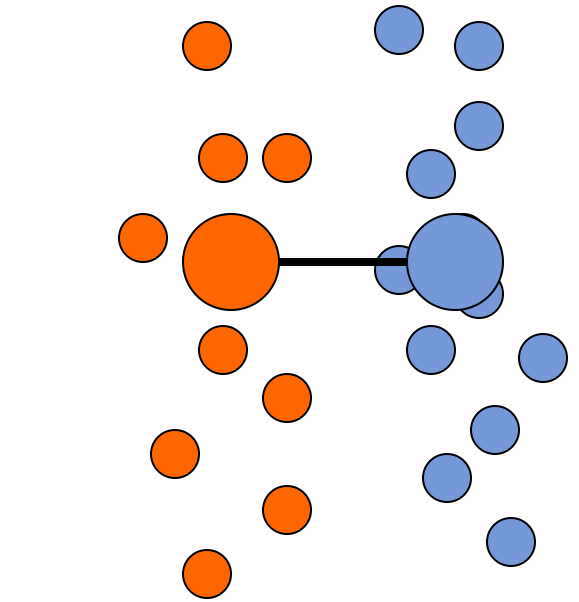
What is a good projection?

- Similarly, what is a good criterion?
  - Separating different classes



What class information may be useful?

- Between−class distance
  - Distance between the centroids of different classes

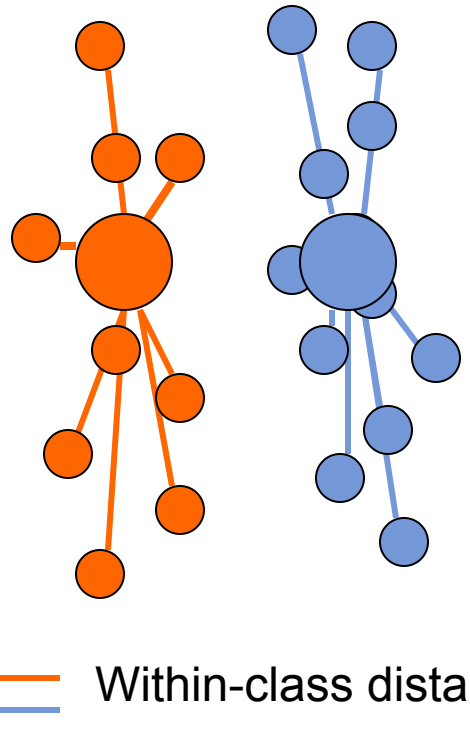


Between-class distance

What class information may be useful?

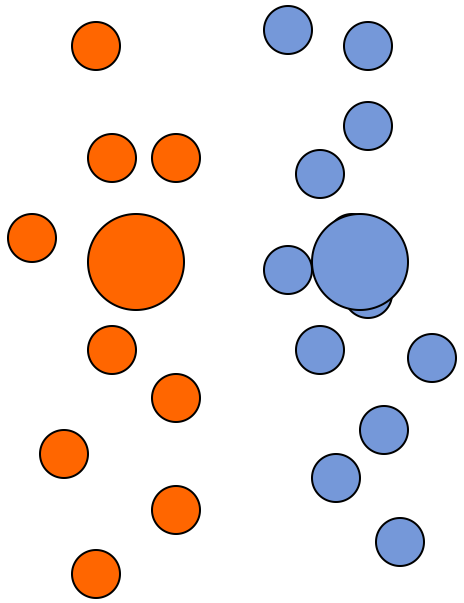
- Between-class distance
  - Distance between the centroids of different classes

- Within−class distance
  - Accumulated distance of an instance to the centroid of its class



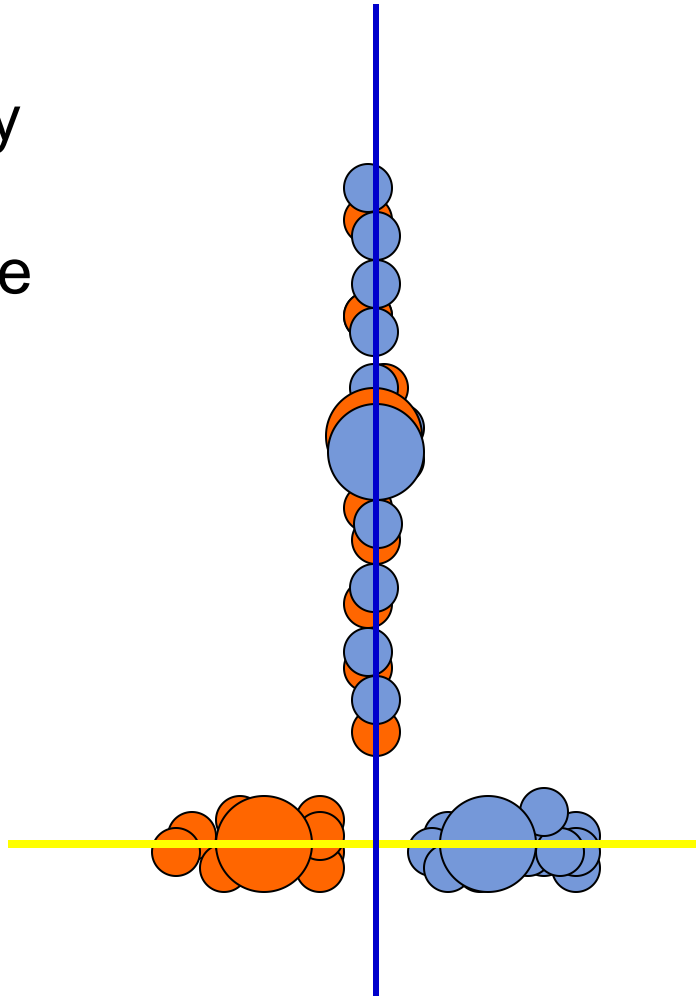
Linear discriminant analysis

- Linear discriminant analysis (LDA) finds most discriminant projection by maximizing between−class distance and minimizing within−class distance



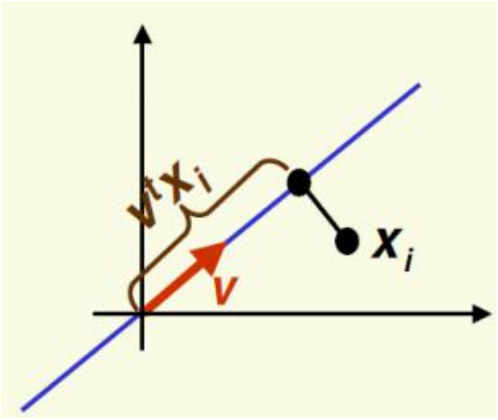
Linear discriminant analysis

- Linear discriminant analysis (LDA) finds most discriminant projection by maximizing between−class distance and minimizing within−class distance



Fisher linear discriminant

- Suppose we have 2 classes and d−dimensional samples $x_1,\ldots,x_n$ where:
  - $n1$ samples come from the first class
  - $n2$ samples come from the second class
- Consider projection on a line
- Let the line direction be given by unit vector v
- The scalar $v^t x_i$ is the distance of the projection of x_i from the origin
- Thus, $v^t x_i$ is the projection of x_i into a one dimensional subspace



Fisher linear discriminant

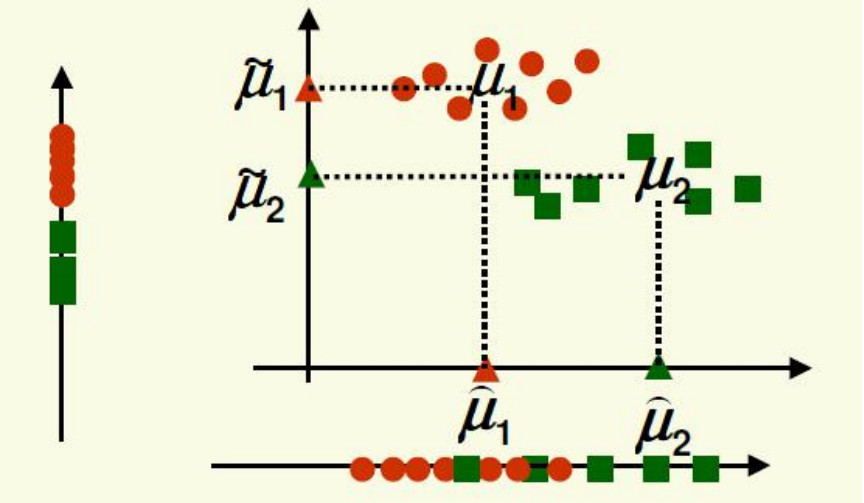
- The projection of sample x_i onto a line in direction v is given by $\mathbf{v}^t\mathbf{x}_i$
- How to measure separation between projections of different classes?
- Let $\tilde{\mu}_1$ and $\tilde{\mu}_2$ be the means of projections of classes 1 and 2
- Let μ_1 and μ_2 be the means of classes 1 and 2
- $|\tilde{\mu}_1 - \tilde{\mu}_2|$ seems like a good measure

$$\tilde{\mu}_1 = \frac{1}{n_1}\sum_{x_i \in C1}^{n_1} \mathbf{v}^t \mathbf{x}_i = \mathbf{v}^t\left(\frac{1}{n_1}\sum_{x_i \in C1}^{n_1} \mathbf{x}_i\right) = \mathbf{v}^t\mu_1$$

similarly, $\tilde{\mu}_2 = \mathbf{v}^t\mu_2$

Fisher linear discriminant

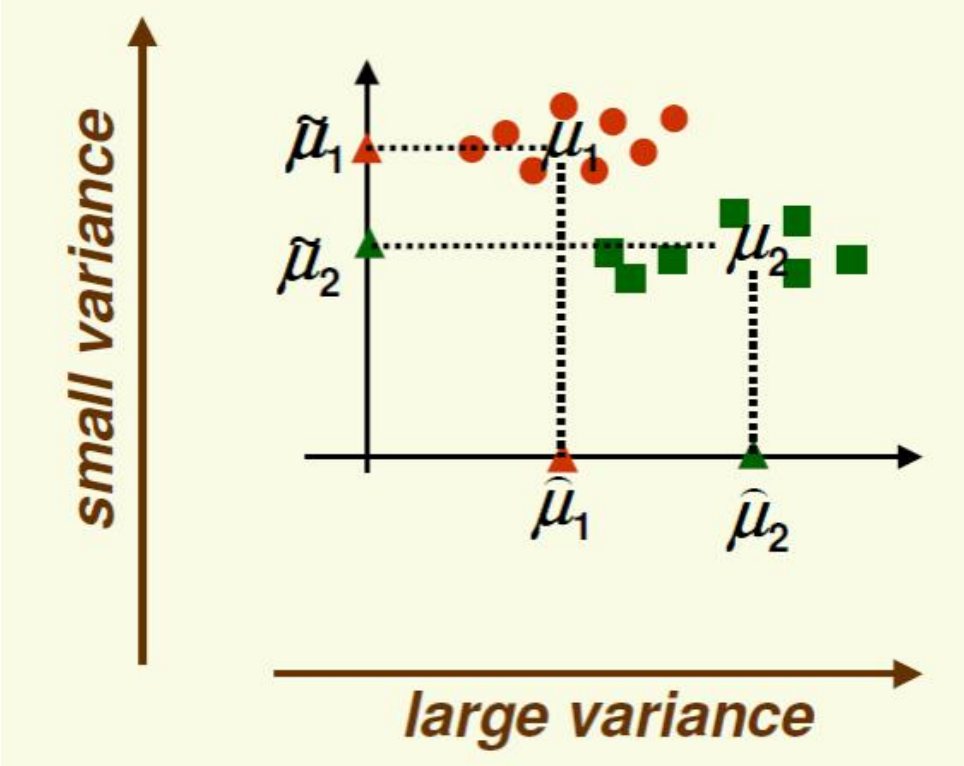
- How good is $|\tilde{\mu}_1 - \tilde{\mu}_2|$ as a measure of separation?
  - The larger it is, the better the expected separation



- The vertical axis is a better line than the horizontal axis to project to for class separability
- However $|\tilde{\mu}_1 - \tilde{\mu}_2| < |\hat{\mu}_1 - \hat{\mu}_2|$

Fisher linear discriminant

- The problem with $|\tilde{\mu}_1 - \tilde{\mu}_2|$ is that it does not consider the variance of the classes

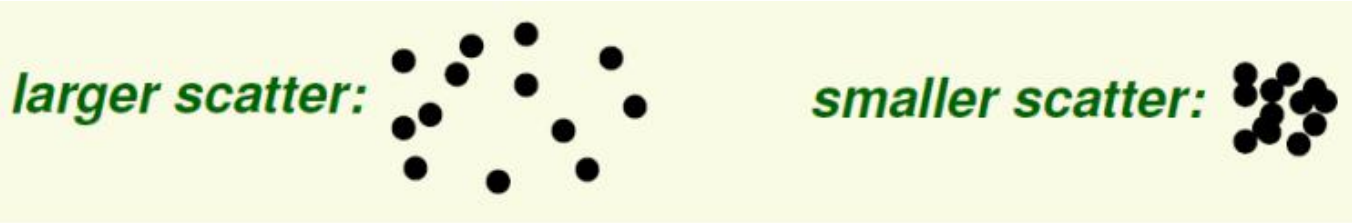


Fisher linear discriminant

- We need to normalize $|\tilde{\mu}_1 - \tilde{\mu}_2|$ by a factor which is proportional to variance
- For samples $z_1, \ldots, z_n$, the sample mean is: $\mu_z = \frac{1}{n}\sum_{i=1}^{n} z_i$
- Define scatter as:

$$s = \sum_{i=1}^{n} (z_i - \mu_z)^2$$

- Thus scatter is just sample variance multiplied by n
  - Scatter measures the same thing as variance, the spread of data around the mean
  - Scatter is just on different scale than variance



Fisher linear discriminant

- Fisher Solution: normalize by scatter $|\tilde{\mu}_1 - \tilde{\mu}_2|$
- Let $y_i = v^t x^i$, be the projected samples
- The scatter for projected samples of class 1 is

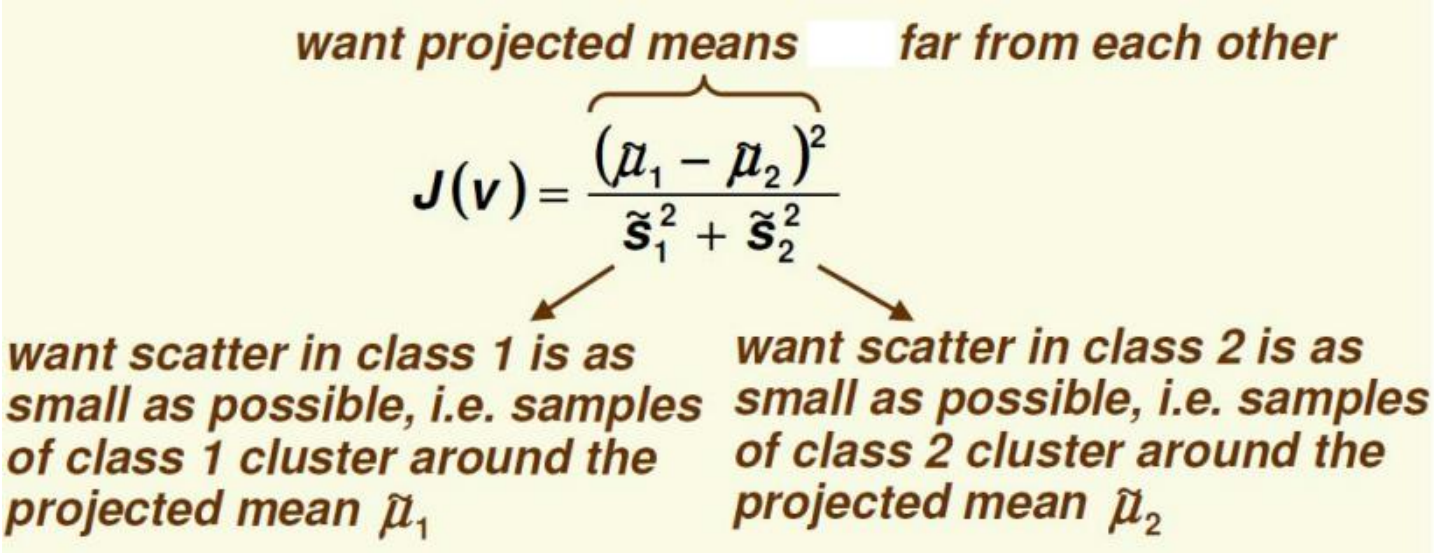
$$\tilde{s}_1^2 = \sum_{y_i \in \text{Class } 1} (y_i - \tilde{\mu}_1)^2$$

- The scatter for projected samples of class 2 is

$$\tilde{s}_2^2 = \sum_{y_i \in \text{Class } 2} (y_i - \tilde{\mu}_2)^2$$

Fisher Linear Discriminant

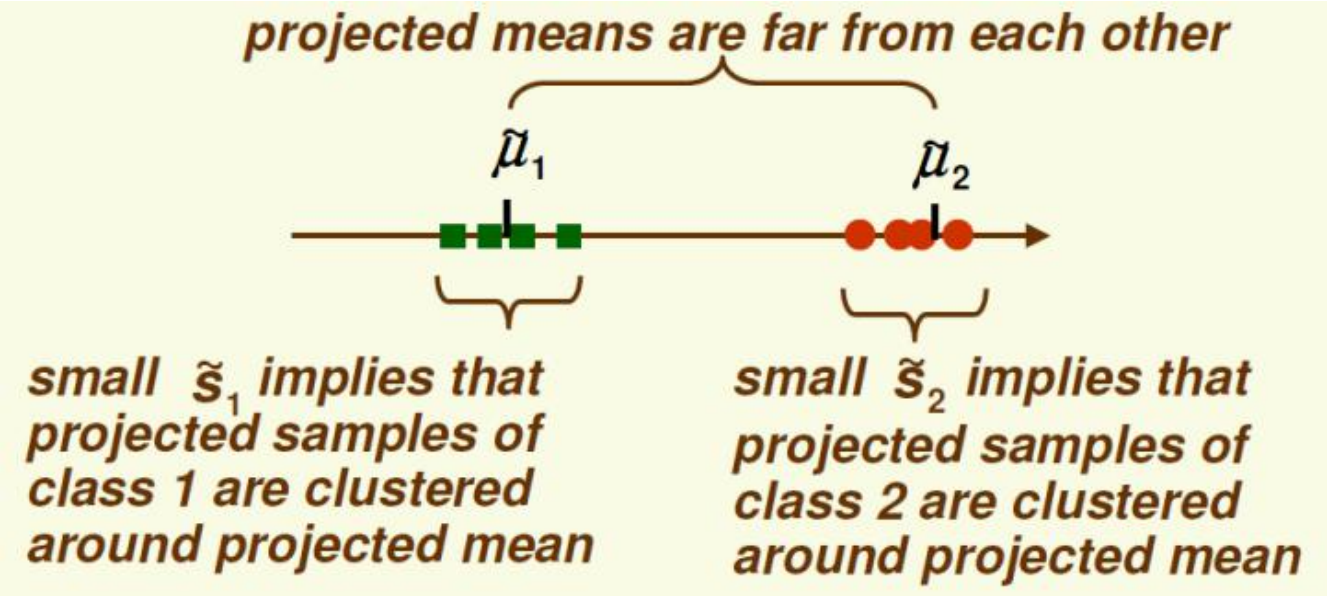
- We need to normalize by both scatter of class 1 and scatter of class 2
- The Fisher linear discriminant is the projection on a line in the direction v which maximizes



Fisher Linear Discriminant

$$J(v) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- If we find v which makes J(v) large, we are guaranteed that the classes are well separated



Fisher Linear Discriminant - Derivation

$$J(v) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- All we need to do now is express J(v) as a function of v and maximize it
  - Straightforward but need linear algebra and calculus
- Define the class scatter matrices S1 and S2.
- These measure the scatter of original samples xi (before projection)

$$S_1 = \sum_{x_i \in Class\ 1} (x_i - \mu_1)(x_i - \mu_1)^t$$

$$S_2 = \sum_{x_i \in Class\ 2} (x_i - \mu_2)(x_i - \mu_2)^t$$

Fisher Linear Discriminant

- Define **within class** scatter matrix

$$S_W = S_1 + S_2$$

$$\tilde{s}_1^2 = \sum_{y_i \in Class\ 1} (y_i - \tilde{\mu}_1)^2$$

- $y_i = v^t x_i$ and $\tilde{\mu}_1 = v^t \mu_1$

$$\begin{aligned}
\tilde{s}_1^2 &= \sum_{y_i \in Class\ 1} \left(v^t x_i - v^t \mu_1\right)^2 \\
&= \sum_{y_i \in Class\ 1} \left(v^t (x_i - \mu_1)\right)^t \left(v^t (x_i - \mu_1)\right) \\
&= \sum_{y_i \in Class\ 1} \left((x_i - \mu_1)^t v\right)^t \left((x_i - \mu_1)^t v\right) \\
&= \sum_{y_i \in Class\ 1} v^t (x_i - \mu_1)(x_i - \mu_1)^t v = v^t S_1 v
\end{aligned}$$

Fisher Linear Discriminant

- Similarly $\tilde{s}_2^2 = \boldsymbol{v}^t \boldsymbol{S}_2 \boldsymbol{v}$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \boldsymbol{v}^t \boldsymbol{S}_1 \boldsymbol{v} + \boldsymbol{v}^t \boldsymbol{S}_2 \boldsymbol{v} = \boldsymbol{v}^t \boldsymbol{S}_W \boldsymbol{v}$$

- Define **between class** scatter matrix

$$\boldsymbol{S}_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t$$

- S_B measures separation of the means of the two classes before projection
- The separation of the projected means canbe written as

$$\begin{aligned}(\tilde{\mu}_1 - \tilde{\mu}_2)^2 &= (\boldsymbol{v}^t \mu_1 - \boldsymbol{v}^t \mu_2)^2 \\ &= \boldsymbol{v}^t (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t \boldsymbol{v} \\ &= \boldsymbol{v}^t \boldsymbol{S}_B \boldsymbol{v}\end{aligned}$$

Fisher Linear Discriminant

- Thus our objective function can be written:

$$J(\mathbf{v}) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{\mathbf{v}^t \mathbf{S}_B \mathbf{v}}{\mathbf{v}^t \mathbf{S}_W \mathbf{v}}$$

- Maximize J(v) by taking the derivative w.r.t. v and setting it to 0

$$\frac{d}{d\mathbf{v}} J(\mathbf{v}) = \frac{\left(\frac{d}{d\mathbf{v}} \mathbf{v}^t \mathbf{S}_B \mathbf{v}\right) \mathbf{v}^t \mathbf{S}_W \mathbf{v} - \left(\frac{d}{d\mathbf{v}} \mathbf{v}^t \mathbf{S}_W \mathbf{v}\right) \mathbf{v}^t \mathbf{S}_B \mathbf{v}}{\left(\mathbf{v}^t \mathbf{S}_W \mathbf{v}\right)^2}$$

$$= \frac{(2\mathbf{S}_B \mathbf{v}) \mathbf{v}^t \mathbf{S}_W \mathbf{v} - (2\mathbf{S}_W \mathbf{v}) \mathbf{v}^t \mathbf{S}_B \mathbf{v}}{\left(\mathbf{v}^t \mathbf{S}_W \mathbf{v}\right)^2} = 0$$

Fisher Linear Discriminant

Need to solve $\boldsymbol{v}^t \boldsymbol{S}_W \boldsymbol{v}(\boldsymbol{S}_B \boldsymbol{v}) - \boldsymbol{v}^t \boldsymbol{S}_B \boldsymbol{v}(\boldsymbol{S}_W \boldsymbol{v}) = 0$

$$\Rightarrow \frac{\boldsymbol{v}^t \boldsymbol{S}_W \boldsymbol{v}(\boldsymbol{S}_B \boldsymbol{v})}{\boldsymbol{v}^t \boldsymbol{S}_W \boldsymbol{v}} - \frac{\boldsymbol{v}^t \boldsymbol{S}_B \boldsymbol{v}(\boldsymbol{S}_W \boldsymbol{v})}{\boldsymbol{v}^t \boldsymbol{S}_W \boldsymbol{v}} = 0$$

$$\Rightarrow \boldsymbol{S}_B \boldsymbol{v} - \frac{\boldsymbol{v}^t \boldsymbol{S}_B \boldsymbol{v}(\boldsymbol{S}_W \boldsymbol{v})}{\boldsymbol{v}^t \boldsymbol{S}_W \boldsymbol{v}} = 0$$

$$\frac{\boldsymbol{v}^t \boldsymbol{S}_B \boldsymbol{v}}{\boldsymbol{v}^t \boldsymbol{S}_W \boldsymbol{v}} = \lambda$$

$$\Rightarrow \underbrace{\boldsymbol{S}_B \boldsymbol{v} = \lambda \boldsymbol{S}_W \boldsymbol{v}}$$

generalized eigenvalue problem

Fisher Linear Discriminant

$$S_B v = \lambda S_W v$$

- If S_W has full rank (the inverse exists), we can convert this to a standard eigenvalue problem

$$S_W^{-1} S_B v = \lambda v$$

- But $S_B x$ for any vector x, points in the same direction as $\mu_1 - \mu_2$

$$S_B x = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t x = (\mu_1 - \mu_2)\underbrace{(\mu_1 - \mu_2)^t x}_{\alpha} = \alpha(\mu_1 - \mu_2)$$

- Based on this, we can solve the eigenvalue problem directly

$$\boxed{v = S_W^{-1}(\mu_1 - \mu_2)}$$

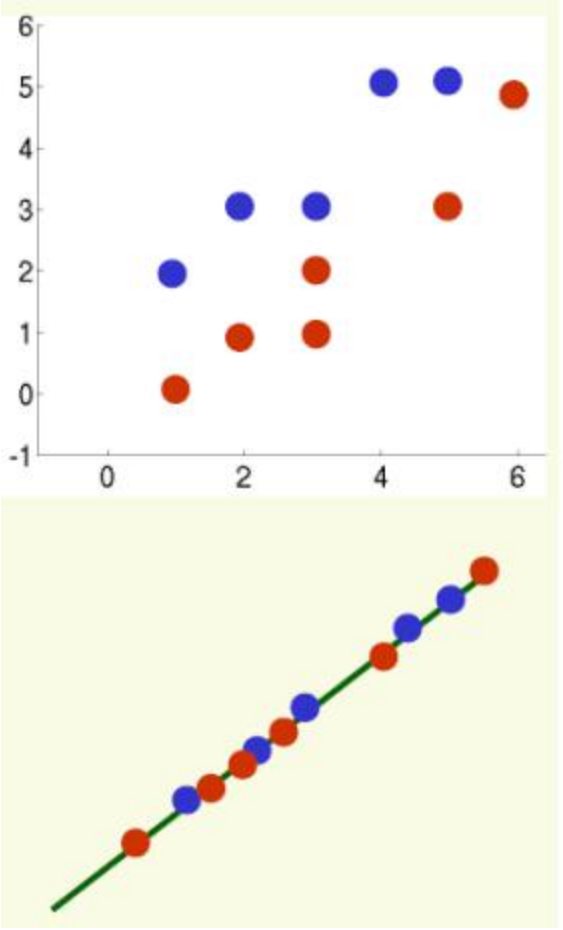
$$S_W^{-1} S_B \underbrace{[S_W^{-1}(\mu_1 - \mu_2)]}_{v} = S_W^{-1}[\alpha(\mu_1 - \mu_2)] = \underbrace{\alpha}_{\lambda}\underbrace{[S_W^{-1}(\mu_1 - \mu_2)]}_{v}$$

Example (1)

- Data
  - – Class 1 has 5 samples
  - $c1=[(1,2),(2,3),(3,3),(4,5),(5,5)]$
  - – Class 2 has 6 samples
  - $c2=[(1,0),(2,1),(3,1),(3,2),(5,3),(6,5)]$
- Arrange data in 2 separate matrices

$$c_1 = \begin{bmatrix} 1 & 2 \\ \vdots & \vdots \\ 5 & 5 \end{bmatrix} \qquad c_2 = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 6 & 5 \end{bmatrix}$$

- Notice that PCA performs very poorly on this data because the direction of largest variance is not helpful for classification



Example (2)

- First compute the mean for each class

$$\mu_1 = mean(c_1) = [3 \quad 3.6]^t \qquad \mu_2 = mean(c_2) = [3.3 \quad 2]^t$$

- Compute scatter matrices S1 and S2 for each class

$$S_1 = 4 * cov(c_1) = \begin{bmatrix} 10 & 8.0 \\ 8.0 & 7.2 \end{bmatrix} \qquad S_2 = 5 * cov(c_2) = \begin{bmatrix} 17.3 & 16 \\ 16 & 16 \end{bmatrix}$$

- Within class scatter:

$$S_W = S_1 + S_2 = \begin{bmatrix} 27.3 & 24 \\ 24 & 23.2 \end{bmatrix}$$

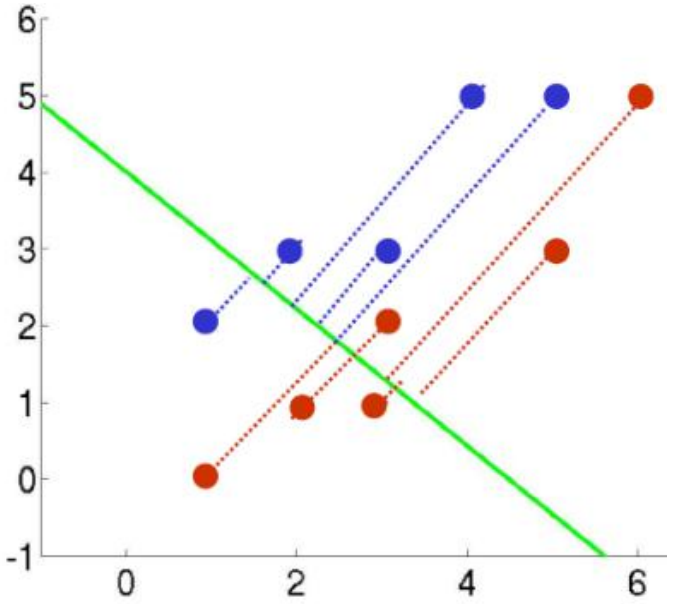
  – it has full rank, don't have to solve for eigenvalues

- The inverse of Sw is: $S_W^{-1} = inv(S_W) = \begin{bmatrix} 0.39 & -0.41 \\ -0.41 & 0.47 \end{bmatrix}$

- Within class scatter: $v = S_W^{-1}(\mu_1 - \mu_2) = \begin{bmatrix} -0.79 \\ 0.89 \end{bmatrix}$

Example (3)

- As long as the line has the right direction, its exact position does not matter
- The last step is to compute the actual 1D vector y
  - Separately for each class



$$Y_1 = v^t c_1^t = \begin{bmatrix} -0.65 & 0.73 \end{bmatrix} \begin{bmatrix} 1 & \cdots & 5 \\ 2 & \cdots & 5 \end{bmatrix} = \begin{bmatrix} 0.81 & \cdots & 0.4 \end{bmatrix}$$

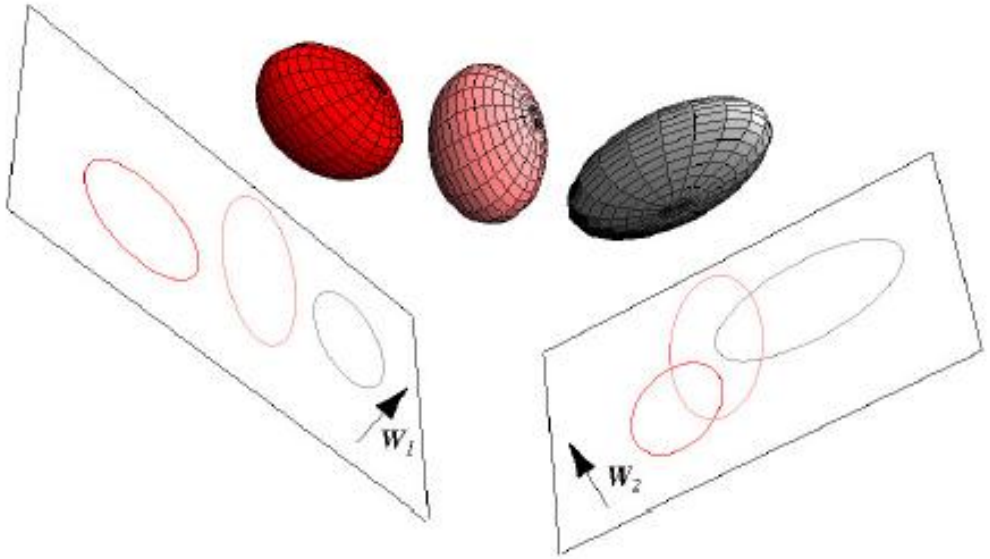
$$Y_2 = v^t c_2^t = \begin{bmatrix} -0.65 & 0.73 \end{bmatrix} \begin{bmatrix} 1 & \cdots & 6 \\ 0 & \cdots & 5 \end{bmatrix} = \begin{bmatrix} -0.65 & \cdots & -0.25 \end{bmatrix}$$

Outline

- Principal component analysis (PCA)
- Application examples with PCA
- Fisher Linear Discriminant
- **Multiple Discriminant Analysis**

Multiple Discriminant Analysis

- Can generalize FLD to multiple classes
  - In case of c classes, we can reduce dimensionality to 1, 2, 3,…, $c-1$ dimensions
  - Project sample xi to a linear subspace $y_i = V^t x_i$
  - V is called projection matrix



Multiple Discriminant Analysis

- Within class scatter matrix:

$$S_W = \sum_{i=1}^{c} S_i = \sum_{i=1}^{c} \sum_{x_k \in \text{class } i} (x_k - \mu_i)(x_k - \mu_i)^t$$

- Between class scatter matrix

$$S_B = \sum_{i=1}^{c} n_i (\mu_i - \mu)(\mu_i - \mu)^t$$

maximum rank is c -1

mean of all data

mean of class i

- Objective function

$$J(V) = \frac{det\left(V^t S_B V\right)}{det\left(V^t S_W V\right)}$$

Multiple Discriminant Analysis

$$J(V) = \frac{det\left(V^t S_B V\right)}{det\left(V^t S_W V\right)}$$

- Solve generalized eigenvalue problem

$$S_B V = \lambda S_W V$$

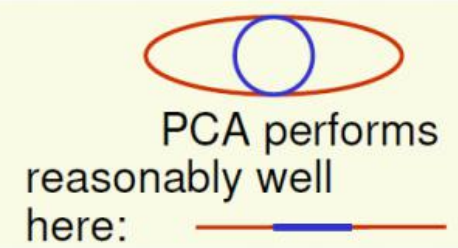
- There are at most $c-1$ distinct eigenvalues
  - with $v_1 \ldots v_{c-1}$ corresponding eigenvectors
- The optimal projection matrix V to a subspace of dimension k is given by the eigenvectors corresponding to the largest k eigenvalues
- Thus, we can project to a subspace of dimension at most $c-1$

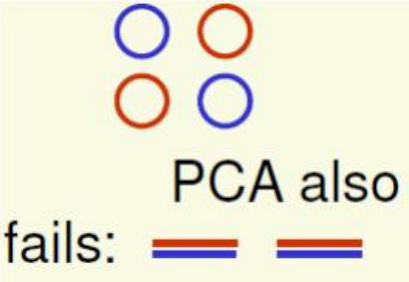
FDA and MDA Drawbacks

- Reduces dimension only to $k = c-1$
  – Unlike PCA where dimension can be chosen to be smaller or larger than $c-1$
- For complex data, projection to even the best line may result in non–separable projected samples

FDA and MDA Drawbacks

- FDA/MDA will fail:
  - If $J(v)$ is always 0: when $\mu_1 = \mu_2$





- If $J(v)$ is always small: classes have large overlap when projected to any line (PCA will also fail)

Q & A