# INF 4300
# Linear feature transforms

Anne Solberg (anne@ifi.uio.no)

Today:

- Feature transformation through principal component analysis
- Fisher's linear discriminant function



## Linear feature transforms

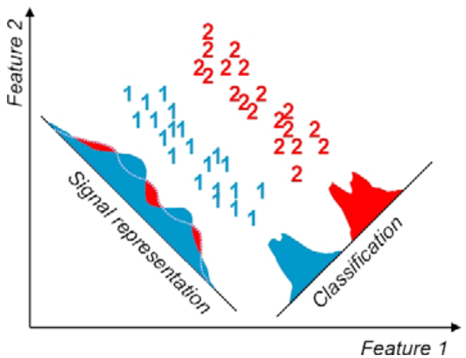
- Feature extraction can be stated as
  - Given a feature space $x_i \in \mathbb{R}_n$ find an optimal mapping $y = f(x) : \mathbb{R}_n \rightarrow \mathbb{R}_m$ with $m < n$.
  - An optimal mapping in classification :the transformed feature vector $y$ yield the same classification rate as $x$.
- The optimal mapping may be a non-linear function
  - Difficult to generate/optimize non-linear transforms
  - Feature extraction is therefore usually limited to linear transforms $y = A^T x$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{11} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}$$



## Signal representation vs classification

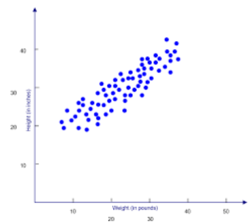
- Principal components analysis (PCA)
  - - signal representation, unsupervised
  - Minimize the mean square representation error (unsupervised)
- Linear discriminant analysis (LDA)
  - -classification, supervised
  - Maximize the distance between the classes (supervised)






## Idea behind (Principal Component Transform)

- Find a projection **y**=$A^T$**x** of the feature vector **x**
- Three interpretations of PCA:
  - Find the projection that maximize the variance along the selected projection
  - Minimize the reconstruction error (squared distance between original and transformed data)
  - Find a transform that gives uncorrelated features



## Definitions: Correlation matrix vs. covariance matrix

- $\Sigma_x$ is the covariance matrix of x

$$\Sigma_X = E\left[(x-\mu)(x-\mu)^T\right]$$

- $R_x$ is the correlation matrix of x

$$R_X = E\left[(x)(x)^T\right]$$

- $R_x=\Sigma_x$ if $\mu_x=0$.

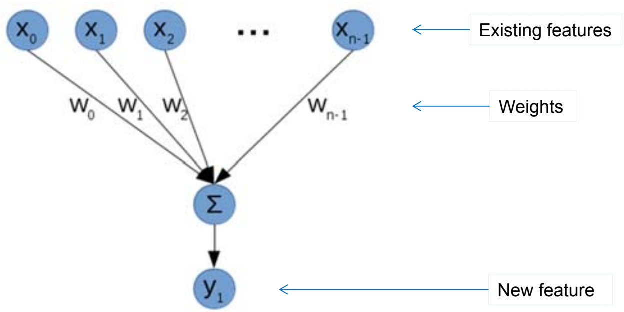


## Principal component or Karhunen-Loeve transform

- Let x be a feature vector.
- Features are often correlated, which might lead to redundancies.
- We now derive a transform which yields **uncorrelated** features.
- We seek a linear transform $y=A^Tx$, and the $y_i$s should be uncorrelated.
- The $y_i$s are uncorrelated if $E[y(i)y(j)^T]=0$, $i\neq j$.
- If we can express the information in x using uncorrelated features, we might need **fewer** coefficients.



## Linear feature transforms I/II






## Linear feature transforms II/II

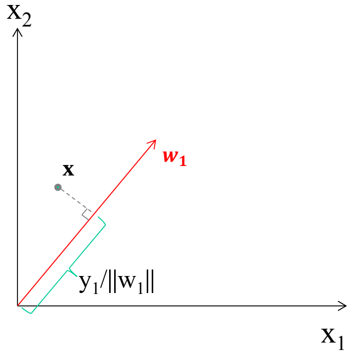
- Multiple output features by applying different weights for each one:

$$y_1=\sum_{i=0}^{n-1} w_{i1}x_i,\quad y_2=\sum_{i=0}^{n-1} w_{i2}x_i,\quad \ldots \quad y_m=\sum_{i=0}^{n-1} w_{im}x_i$$

- In matrix notation $\mathbf{y} = \mathbf{A}^T\mathbf{x}$, $\mathbf{A}=[\mathbf{w}_1\ \mathbf{w}_2 \ldots \mathbf{w}_m]$
- If **y** has fewer elements than **x**, we get a feature reduction

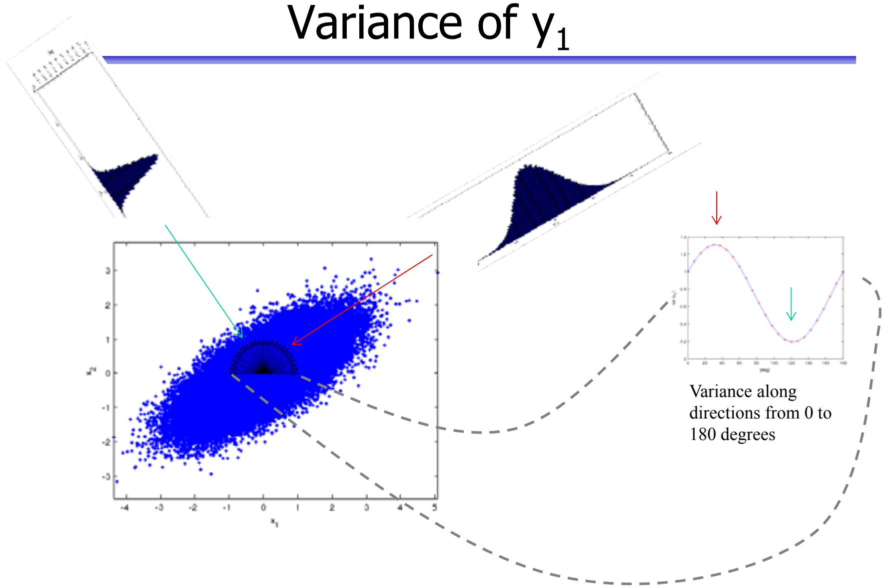
## The weights | Visualization and intuition

$$y_1 = \sum_{i=0}^{n-1} w_{i1} x_i = \boldsymbol{w_1}^T \boldsymbol{x}$$






## Variance of $y_1$






## Variance of $y_1$ cont.

- Assume mean of **x** is subtracted

$$\sigma_{y_1}^2 = \frac{1}{N}\sum_i y_i^2$$

$$= \frac{1}{N}\sum_i (\mathbf{w}^T\mathbf{x}_i)^2 = \frac{1}{N}\sum_i \mathbf{w}^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{w} = \mathbf{w}^T\underbrace{\left(\frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T\right)}_{\text{The sample covariance matrix / scatter matrix; } \mathbf{R}}\mathbf{w}$$

$$= \mathbf{w}^T\mathbf{R}\mathbf{w}$$

Called $\sigma^2_w$ on some slides



## Variance and projection residuals

Single sample

Projection onto **w**, assuming |**w**|=1

$$\begin{aligned}
||\vec{x}_i - \underbrace{(\vec{w}\cdot\vec{x}_i)}_{\text{«}y_i\text{»}}\vec{w}||^2 &= (\vec{x}_i - (\vec{w}\cdot\vec{x}_i)\vec{w})\cdot(\vec{x}_i - (\vec{w}\cdot\vec{x}_i)\vec{w}) \\
&= \vec{x}_i\cdot\vec{x}_i - \vec{x}_i\cdot(\vec{w}\cdot\vec{x}_i)\vec{w} \\
&\quad -(\vec{w}\cdot\vec{x}_i)\vec{w}\cdot\vec{x}_i + (\vec{w}\cdot\vec{x}_i)\vec{w}\cdot(\vec{w}\cdot\vec{x}_i)\vec{w} \\
&= ||\vec{x}_i||^2 - 2(\vec{w}\cdot\vec{x}_i)^2 + (\vec{w}\cdot\vec{x}_i)^2\vec{w}\cdot\vec{w} \\
&= \vec{x}_i\cdot\vec{x}_i - \underbrace{(\vec{w}\cdot\vec{x}_i)^2}_{\text{«}y_i^2\text{»}}
\end{aligned}$$

w·w=1

$$MSE(\vec{w}) = \frac{1}{n}\left(\sum_{i=1}^{n}||\vec{x}_i||^2 - \sum_{i=1}^{n}(\vec{w}\cdot\vec{x}_i)^2\right)$$

Sum all *n* samples (not dimensions)

Note: Max variance ↔ min projection residuals!

$\sigma^2_w$

## Criterion function

- Goal: Find transform minimizing representation error
- We start with a single weight-vector, **w**, giving us a single feature, $y_1$
- Let $J(\mathbf{w}) = \mathbf{w}^T\mathbf{R}\mathbf{w} = \sigma_w^2$
- Now, let's find $\max_{\boldsymbol{w}} J(\boldsymbol{w})$ $s.t. ||\boldsymbol{w}|| = 1$

As we learned on the previous slide, maximizing this is equivalent to minimizing representation error



## Maximizing variance of $y_1$

$$\mathscr{L}(\mathbf{w},\lambda) \equiv \sigma_\mathbf{w}^2 - \lambda(\mathbf{w}^T\mathbf{w}-1)$$

$$\frac{\partial L}{\partial \lambda} = -\mathbf{w}^T\mathbf{w}+1$$

$$\frac{\partial L}{\partial \mathbf{w}} = 2\mathrm{R}\mathbf{w} - 2\lambda\mathbf{w}$$

Lagrangian function for maximizing $\sigma^2_w$ with the constraint $\mathbf{w}^T\mathbf{w}$=1

$\Downarrow$ Equating zero

$$\mathbf{w}^T\mathbf{w} = 1$$

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

Unfamiliar with Lagrangian multipliers? See http://biostat.mc.vanderbilt.edu/wiki/pub/Main/CourseBios362/LagrangeMultipliers-Bishop-PatternRecognitionMachineLearning.pdf

The maximizing **w** is an eigenvector of R!

And $\sigma^2_w$=λ! [Why?]



## $\mathbf{w}_2$, $\mathbf{w}_3$, .. I/III

- Ok, I've got the $\mathbf{w}_1$ giving me the transform (linear weights) that maximizes the variance / minimizes the representation error ..
- .. Now I want another one that again maximizes the variance / minimizes the representation error, but the new feature should be uncorrelated with my previous one ..
- .. Which $\mathbf{w}_2$ would give me this?



## Eigendecomposition of covariance matrices

Real-valued, symmetric, «n-dimensional» covariance matrix

$$\mathbf{R} = \lambda_1\mathbf{a}_1\mathbf{a}_1' + \lambda_2\mathbf{a}_2\mathbf{a}_2' + \ldots + \lambda_n\mathbf{a}_n\mathbf{a}_n'$$

Eigenvalue (let's say largest)

Eigenvector corresponding to $\lambda_1$

Smallest eigenvalue

$\mathbf{a}^T_i\mathbf{a}_j = 0$ for i ≠ j

Remember: $\lambda_i$=variance of $\mathbf{x}^T\mathbf{a}_i$

# $\mathbf{w}_2$, $\mathbf{w}_3$, .. II/III

- What does uncorrelated mean? Zero covariance.
- Covariance of $y_1$ and $y_2$:

$$\frac{1}{N}\sum_{i}^{N} y_1(i)' y_2(i) = \frac{1}{N}\sum_{i}^{N} \mathbf{w}_1'\mathbf{x}(i)\mathbf{x}(i)'\mathbf{w}_2 = \mathbf{w}_1'\mathbf{R}\mathbf{w}_2$$

- We already have that $\mathbf{w}_1 = \mathbf{a}_1$
- From last slide, requiring $\mathbf{w}_1'\mathbf{R}\mathbf{w}_2 = \mathbf{a}_1'\mathbf{R}\mathbf{w}_2 = 0$ means requiring $\mathbf{w}_2'\mathbf{a}_1 = 0$



# $\mathbf{w}_2$, $\mathbf{w}_3$, .. III/III

- We want $\max_{\mathbf{w}} \mathbf{w}'\mathbf{R}\mathbf{w}$, s.t. $|\mathbf{w}|=1$ *and* $\mathbf{w}'\mathbf{a}_1=0$
- We can simply remove $\lambda_1\mathbf{a}_1\mathbf{a}_1'$ from $\mathbf{R}$, creating $\mathbf{R}_{next} = \mathbf{R} - \lambda_1\mathbf{a}_1\mathbf{a}_1'$, and again find $\max_{\mathbf{w}} \mathbf{w}'\mathbf{R}_{next}\mathbf{w}$ s.t. $|\mathbf{w}|=1$
- Studying the decomposition of $\mathbf{R}$ (a few slides back), we see that the solution is the eigenvector corresponding to the second largest eigenvalue
- Similarly, the $\mathbf{w}_3$, $\mathbf{w}_4$ etc. are given by the following eigenvectors sorted according to their eigenvalues



# $\mathbf{w}_2$, $\mathbf{w}_3$, .. III+/III

$\max_{\mathbf{w}} \mathbf{w}'\mathbf{R}\mathbf{w}$, s.t. $|\mathbf{w}|=1$

$$\mathbf{R} = \lambda_1\mathbf{a}_1\mathbf{a}_1' + \lambda_2\mathbf{a}_2\mathbf{a}_2' + \ldots + \lambda_n\mathbf{a}_n\mathbf{a}_n' \quad \rightarrow \quad \mathbf{w}=\mathbf{a}_1$$

$$\mathbf{R} = \cancel{\lambda_1\mathbf{a}_1\mathbf{a}_1'} + \lambda_2\mathbf{a}_2\mathbf{a}_2' + \ldots + \lambda_n\mathbf{a}_n\mathbf{a}_n' \quad \rightarrow \quad \mathbf{w}=\mathbf{a}_2$$

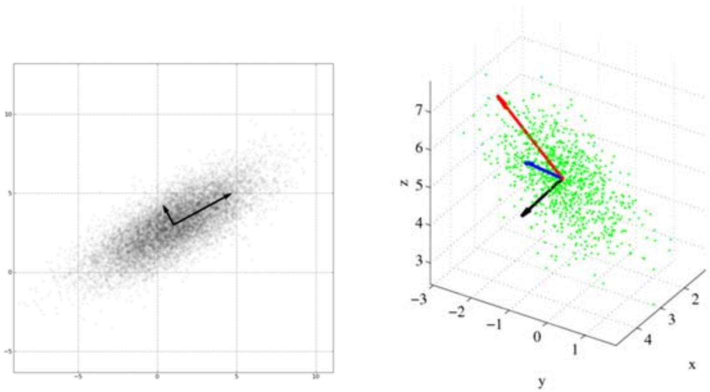
$$\mathbf{R} = \cancel{\lambda_1\mathbf{a}_1\mathbf{a}_1'} + \cancel{\lambda_2\mathbf{a}_2\mathbf{a}_2'} + \ldots + \lambda_n\mathbf{a}_n\mathbf{a}_n' \quad \rightarrow \quad \mathbf{w}=\mathbf{a}_3$$

… etc.

Eigenvectors sorted by their corresponding eigenvalues



# Example of distributions and eigenvectors



(Illustration courtesy of «the Internet»)

# Principal component transform (PCA)

- Place the *m* «principle» eigenvectors (the ones with the largest eigenvalues) along the columns of A
- Then the transform $\mathbf{y} = \mathbf{A}^T\mathbf{x}$ gives you the *m* first principle components
- The *m*-dimensional **y**
  - have uncorrelated elements
  - retains as much variance as possible
  - gives the best (in the mean-square sense) description of the original data (through the «image»/projection/reconstruction **Ay**)

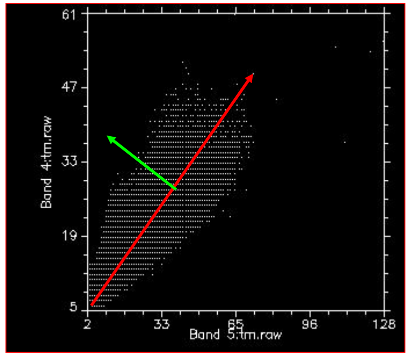
Note: The eigenvectors themselves can often give interesting information

PCA is also known as Karhunen-Loeve transform



# Geometrical interpretation of principal components

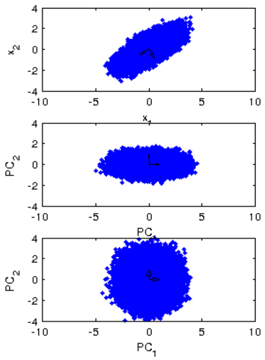
- The eigenvector corresponding to the largest eigenvalue is the direction in n-dimensional space with highest variance.
- The next principal component is orthogonal to the first, and along the direction with the second largest variance.




Note that the direction with the highest variance is NOT related to separability between classes.



# PCA and rotation and «whitening»




If we use all eigenvectors in the transform, $\mathbf{y} = \mathbf{A}^t\mathbf{x}$, we simply rotate our data so that our new features are uncorrelated, i.e., cov(**y**) is a diagonal matrix.

If we as a next step scale each feature by their $\sigma^{-1}$, $\mathbf{y} = \mathbf{D}^{(-1/2)}\mathbf{A}^t\mathbf{x}$, where **D** is a diagonal matrix of eigenvalues (i.e., variances), we get cov(**y**)=**I**. We say that we have «whitened» the data.

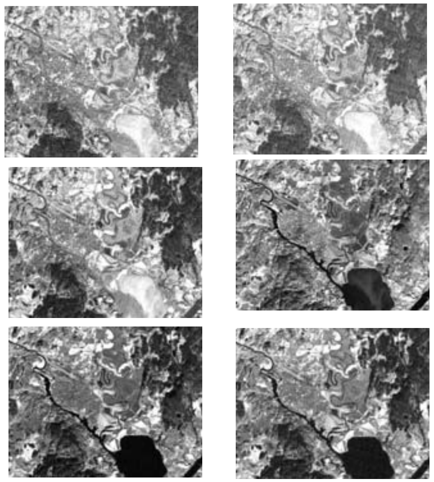
*Note*: Uncorrelated variables need not appear round/spherical:



# PCA and multiband images

- We can compute the principal component transform for an image with *n* bands
- Let **X** be an *N*x*n* matrix having a row for each image sample
- Sample covariance matrix (after mean subtracted): $\boldsymbol{R} = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{X}$
- Place the (sorted) eigenvectors along the columns of **A**
- **Y**=**XA** will then contain the image samples, however most of the variance is in the «bands» with the lowest index (corresponding to the largest eigenvalues), and the new features are uncorrelated

# PCA example – original image

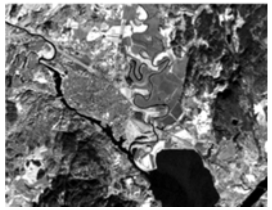


- Satellite image from Kjeller
- 6 spectral bands with different wavelengths

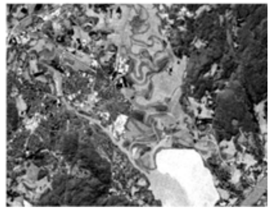
| | | | |
|---|---|---|---|
| 1 | Blue | 0.45-0.52 | Max. penetration of water |
| 2 | Green | 0.52-0.60 | Vegetation and chlorophyll |
| 3 | Red | 0.63-0.69 | Vegetation type |
| 4 | Near-IR | 0.76-0.90 | Biomass |
| 5 | Mid-IR | 1.55-1.75 | Moisture/water content in vegetation/soil |
| 7 | Mid-IR | 2.08-2.35 | Minerals |



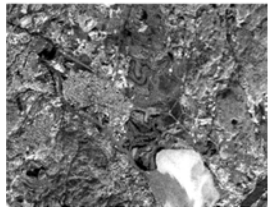
## Example cont: Principal component images



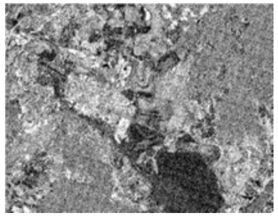
Principal component 1



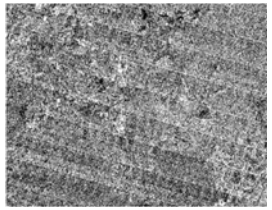
Principal component 2



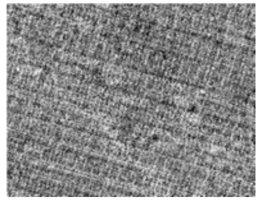
Principal component 3



Principal component 4



Principal component 5



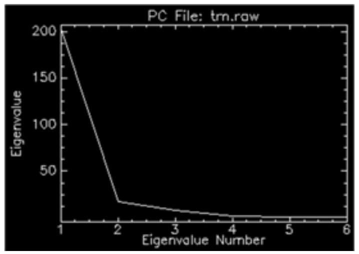
Principal component 6



## Example cont: Inspecting the eigenvalues

The mean-square representation error we get with m of the N PCA-components is given as

$$E\left[\|x-\hat{x}\|^2\right] = \sum_{i=1}^{N-1}\lambda_i - \sum_{i=1}^{m}\lambda_i = \sum_{i=m}^{N-1}\lambda_i$$

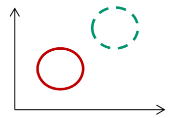



Plotting $\lambda$s will give indications on how many features are needed for representation

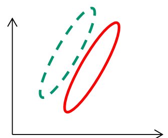


# PCA and classification

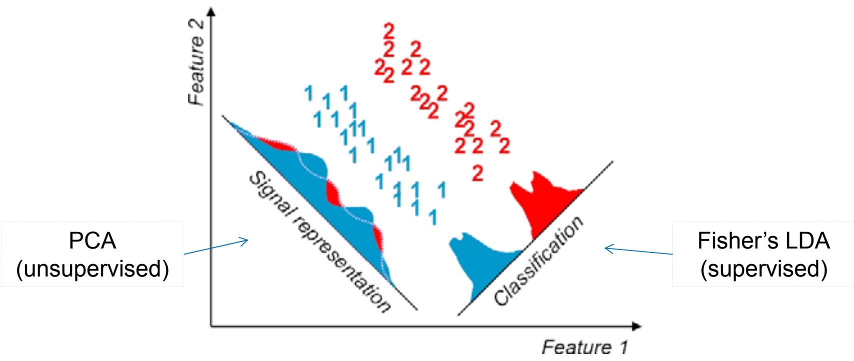
- Reduce overfitting by detecting directions/components without any/very little variance
- Sometimes high variation means useful features for classification:



- .. and sometimes not:



## Intro to Fisher's linear discriminant



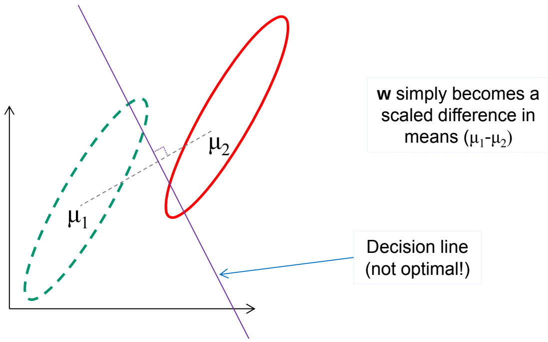



## Criterion function - a first attempt

- To find a good projection vector for classification, we need to define a measure of separation between the projections. This will be the criterion function $J(\mathbf{w})$
- A naive choice would be projected mean difference, $J(w) = |\tilde{\mu}_1 - \tilde{\mu}_2|^2$, s.t. $|\mathbf{w}|=1$.

This criterion does not consider variance in **y**.

Optimal only when $\mathrm{cov}(\mathbf{x}) = \sigma^2\mathbf{I}$ for all classes (then var(**y**) does not change with **w**).






## A criterion function including variance

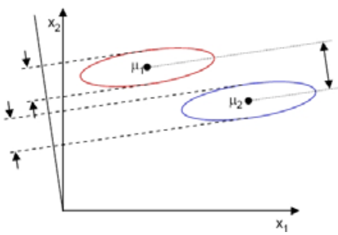
- Fisher's solution: Maximize a function that represents the difference between the means, scaled by a measure of the within-class scatter
- Define classwise scatter (scaled variance)

$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2$$

- $\tilde{s}_1^2 + \tilde{s}_2^2$ is *within class scatter*
- Fisher's criterion is then

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

- We look for a projection where examples from the same class are close to each other, while at the same time projected mean values are as far apart as possible






## Scatter matrices – M classes

- Within-class scatter matrix:

$$S_w = \sum_{i=1}^{M} P(\omega_i) S_i$$

$$S_i = E\left[(x-\mu_i)(x-\mu_i)_T\right]$$

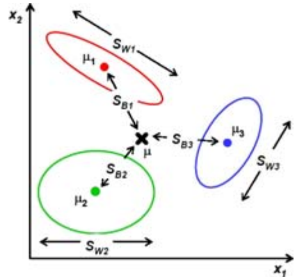
Weighted average of each class' sample covariance matrix

- Between-class scatter matrix:

$$S_b = \sum_{i=1}^{M} P(\omega_i)(\mu_i - \mu)(\mu_i - \mu)^T$$

$$\mu = \sum_{i=1}^{M} P(\omega_i)\mu_i$$

Sample covariance matrix for the means

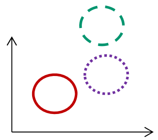



Fisher criterion in terms of within-class and between-class scatter matrices:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}$$

# Multiple classes, $\mathbf{S}_W = \sigma^2 \mathbf{I}$

- If $\mathbf{S}_w=\sigma^2\mathbf{I}$, we can fix $||\mathbf{w}||=1$ and make the denominator in $J(\mathbf{w})$ independent of $\mathbf{w}$ → $J(\mathbf{w})$ guided by the spread of the means ($\mathbf{S}_b$) only:



$$J(\mathbf{w})=\mathbf{w}^T\mathbf{S}_b\mathbf{w}$$

We should know how to maximize this s.t. $|\mathbf{w}|=1$ by now!

- Weight-vector giving maximum separability is given by principal eigenvector of $\mathbf{S}_b$
  - Second best (and orthogonal to first) by next-to-principal
  - … etc. for higher dimensional settings
  - … until a maximum of M-1 dimensions (number of classes minus one) [If classes are «isotropically» Gaussian distributed, all discriminatory information is in this subspace!]



# General $\mathbf{S}_W$ I/II

- We saw that $\mathbf{S}_w=\mathbf{I}$ gave Fisher criterion independent of $\mathbf{S}_w$, and only dependent on $\mathbf{S}_b$

- We can get there by «whitening» the data before applying the Fisher criterion
  - Whitening data by rotation and scaling -> No general loss as distribution overlap does not change

- We must find $\mathbf{y}=\mathbf{A}^T\mathbf{x}$ that yields $\mathbf{S}_{wy} = \mathbf{I}$
  - We have seen that PCA gives uncorrelated data, per-feature scaling can give unit variance per feature:
  - $\mathbf{y} = \mathbf{D}^{-1/2}\mathbf{A}^T\mathbf{x}$, where $\mathbf{A}$ has eigenvectors of $\mathbf{S}_w$ as columns, and $\mathbf{D}$ is a diagonal matrix with corresponding eigenvalues

$$\mathbf{S}_{w_y} = \frac{1}{N}\sum_i (\mathbf{D}^{-1/2}\mathbf{A}^T\mathbf{x}_i)(\mathbf{D}^{-1/2}\mathbf{A}^T\mathbf{x}_i)^T = \mathbf{D}^{-1/2}\mathbf{A}^T\mathbf{S}_w\mathbf{A}\mathbf{D}^{-1/2} = \mathbf{D}^{-1/2}\mathbf{D}\mathbf{D}^{-1/2} = \mathbf{I}$$



# General $\mathbf{S}_W$ II/II

- Let $\mathbf{B} = \mathbf{D}^{-1/2}\mathbf{A}^T$ (the whitening transform)
- $\mathbf{S}_b$ becomes after whitening step:

$$\mathbf{S}_{by} = \mathbf{B}\mathbf{S}_b\mathbf{B}^T$$

- Ignoring the denominator (which is now independent of $\mathbf{w}$), we get
  - $J_y(\mathbf{w}) = \mathbf{w}^T\mathbf{S}_{by}\mathbf{w} = \mathbf{w}^T\mathbf{B}\mathbf{S}_b\mathbf{B}^T\mathbf{w}$, s.t. $|\mathbf{w}|=1$

- The weight-vectors, $\mathbf{w}^*$, maximizing separation are now given by the principal eigenvectors of $\mathbf{B}\mathbf{S}_b\mathbf{B}^T$ (in the whitened space)

Set $J_y(\mathbf{w}^*)=J(\mathbf{w})$ to see this

- In the original space, $\mathbf{w} = \mathbf{B}^T\mathbf{w}^* = \mathbf{A}\mathbf{D}^{-1/2}\mathbf{w}^*$



# Solving Fisher more directly

- Alternatively, you can notice that

$$J(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_b\mathbf{w}}{\mathbf{w}^T\mathbf{S}_w\mathbf{w}}$$

- .. is a «generalized Rayleigh quotient» and look up the solution for its maximum, which is the principal eigenvector of

$$\boxed{\mathbf{S}_W^{-1}\mathbf{S}_b}$$

- The following solutions (orthogonal in $\mathbf{S}_w$, i.e., $\mathbf{w}_i^T\mathbf{S}_w\mathbf{w}_j=0$, for $i\neq j$) are the next principal eigenvectors

Note that the obtained **w**s are identical (up to scaling) to those from the two-step procedure from the previous slides

## Computing Fishers linear discriminant

- For l=M-1:
  - Form a matrix C such that its columns are the M-1 eigenvectors of $S_{xw}^{-1}S_{xb}$
  - Set $\hat{y} = C^T x$
  - This gives us the maximum $J_3$ value.
  - This means that we can reduce the dimension from m to M-1 without loss in class separability power (but only if $J_3$ is a correct measure of class separability.)
  - Alternative view: with a Bayesian model we compute the probabilities $P(\omega_i|x)$ for each class (i=1,...M). Once M-1 probabilities are found, the remaining $P(\omega_M|x)$ is given because the $P(\omega_i|x)$'s sum to one.



## Computation: Case 2: l<M-1

- Form C by selecting the eigenvectors corresponding to the l largest eigenvalues of

$$S_{xw}^{-1}S_{xb}$$

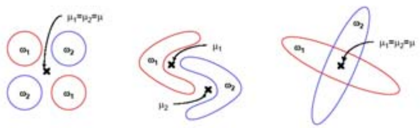
- We now have a loss of discriminating power since

$$J_{3,\hat{y}} < J_{3,x}$$



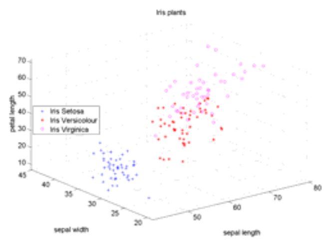
## Limitations of Fisher's discriminant

- Its criterion function is based on all classes having a similarly-shaped Gaussian distribution
  - Any deviance from this could lead to problems / suboptimal or poor solutions
- It produces at most M-*1* (meaningful) feature projections
- One could «overfit» $\mathbf{S}_w$
- It will fail when the discriminatory information is not in the mean but in the variance of the data (failing to meet that stated in the first bulletpoint!)

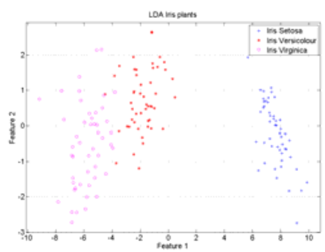





## Fisher's discriminant example




← Original 3D data




Best 2 Fisher's


Best 2 PCA

# Summary

- PCA (unsupervised)
  - Max variance <-> min projection error
  - Eigenvectors of sample cov.mat. / scatter matrix
- Fisher's linear discriminant (supervised)
  - Maximizes spread of means while minimizing intra-class spread
  - $\mathbf{S}_{wy}=\mathbf{I}$ and «whitening of data»
  - Eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$
  - At most nClasses-1 features
  - Limitations



# Literature on pattern recognition

- A review on statistical pattern recognition (still good fifteen years on):
  - A. Jain, R. Duin and J. Mao: Statistical pattern recognition: a review, IEEE Trans. Pattern analysis and Machine Intelligence, vol. 22, no. 1, January 2001, pp. 4--
- Classical PR-books
  - R. Duda, P. Hart and D. Stork, Pattern Classification, 2. ed. Wiley, 2001
  - B. Ripley, Pattern Recognition and Neural Networks, Cambridge Press, 1996.
  - S. Theodoridis and K. Koutroumbas, Pattern Recognition, Academic Press, 2006.