INF 5300
Linear feature transforms

Anne Solberg (anne@ifi.uio.no)

Today:

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

Feature transforms

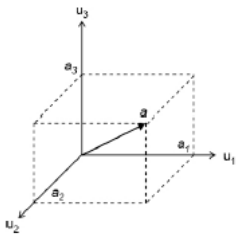
- We now consider computing new features as linear combinations of the existing features.
- From the original feature vector x, we compute a new vector y of transformed features

  $y=A^T x$

  y is M-dimensional, x is *N*-dimensional, A is a $M \times N$ matrix.
- y is normally defined in such a way that it has lower dimension than x.

Vector spaces

- A set of vectors $u_1, u_2, \ldots u_n$ is said to form a basis for a vector space if any arbitrary vector x can be represented by a linear combination $x = a_1 u_1 + a_2 u_2 + \ldots a_n u_n$
  - The coefficients $a_1, a_2, \ldots a_n$ are called the components of vector x with respect to the basis u_i
  - In order to form a basis, it is necessary and sufficient that the u_i vectors be linearly independent
  - A basis u_i is said to be orthogonal if $u_i^T u_j = \begin{cases} \neq 0 & i = j \\ = 0 & i \neq j \end{cases}$
  - A basis u_i is said to be orthonormal if $u_i^T u_j = \begin{cases} = 1 & i = j \\ = 0 & i \neq j \end{cases}$



Linear transformation

- A linear transformation is a mapping from a vector space X^N onto a vector space Y^M, and is represented by a matrix
  - Given a vector $x \in X^N$, the corresponding vector y on Y^M is

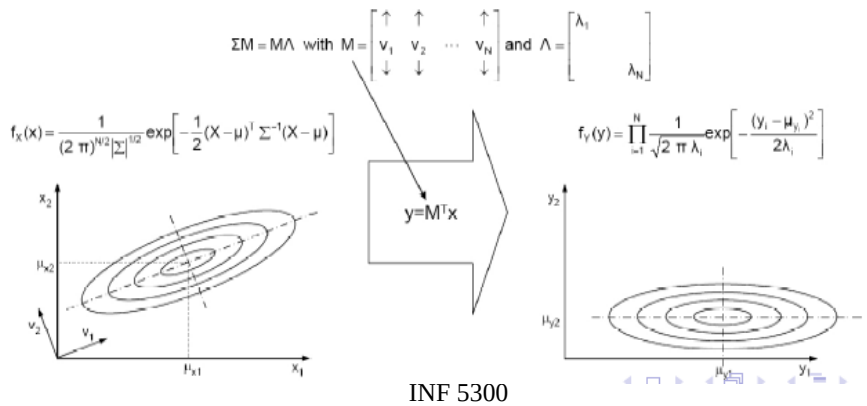
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{11} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m1} & \ldots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}$$

Eigenvalues and eigenvectors

- Given a matrix $A_{N\times N}$, we say that v is an eigenvector if there exists a scalar λ (the eigenvalue) such that $Av = \lambda v \Leftrightarrow v$ is an eigenvector with corresponding eigenvalue λ
- $Av = \lambda v \Rightarrow (A - \lambda I)v = 0 \Rightarrow$
  $|(A - \lambda I)| = 0 \Rightarrow \underbrace{\lambda^N + a_1\lambda^{N-1} + \ldots a_{N-1}\lambda + a_0 = 0}_{\text{Characteristic equation}}$
- Zeroes of the characteristic equation are the eigenvalues of A
- A is non-singular $\Leftrightarrow$ all eigenvalues are non-zero
- A is real and symmetric $\Leftrightarrow$ all eigenvalues are real, and eigenvectors are orthogonal

Interpretation of eigenvectors and eigenvalues

- The eigenvectors of the covariance matrix Σ correspond to the *principal axes* of equiprobability ellipses!
- The linear transformation defined by the eigenvectors of Σ leads to vectors that are uncorrelated *regardless* of the form of the distribution
  - If the distribution happens to be Gaussian, then the transformed vectors will be statistically independent



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 $y = A^T x$

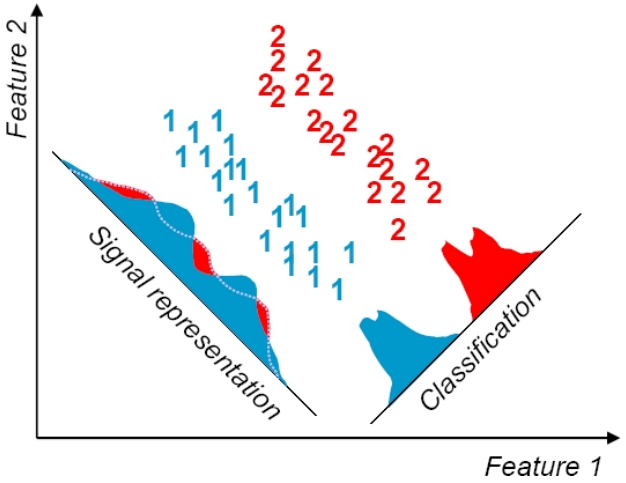
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{11} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}$$

Signal representation vs classification

- The search for the feature extraction mapping $y = f(x)$ is guided by an objective function we want to maximize.
- In general we have two categories of objectives in feature extraction:
  - Signal representation: Accurately approximate the samples in a lower-dimensional space by minimizing the mean square error between the original feature vector and the low-dimensional projection.
  - Classification: Keep (or enhance) class-discriminatory information in a lower-dimensional space.

Signal representation vs classification

- Principal components analysis (PCA)
  - - This representation is optimal in terms of signal representation
  - Upsupervised
  - Minimize the mean square representation error
  - This will correspond to first choosing the direction with maximum variance.
- Fisher's Linear discriminant
  - This transform is optimal in terms of supervised classification.
  - It maximizes the distance between the classes



Correlation matrix vs. covariance matrix

- Σ_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 want a transform which yields uncorrelated features.**
- We seek a linear transform $y=A^Tx$, and the y_is should be uncorrelated.
- The basis vectors will then be linear independent.
- The y_is are uncorrelated if $E[y(i)y(j)]=0$, $i \neq j$.
- If we can express the information in x using uncorrelated features, we might need **fewer** coefficients.

Principal component transform

- The correlation of Y is described by the correlation matrix
  $R_Y=E[yy^T]=E[A^Txx^TA]=A^TR_xA$ R_x is the correlation matrix of X
  R_x is symmetric, thus all eigenvectors are orthogonal.
- We seek uncorrelated components of Y, thus R_y should be diagonal.

From linear algebra:

- R_y will be diagonal if A is formed by the orthogonal eigenvectors a_i, i=0,…,N-1 of R_x: $R_y=A^TR_xA=\Lambda$, where Λ is diagonal with the eigenvalues of R_x, λ_i, on the diagonal.
- We find A by solving the equation $A^TR_xA=\Lambda$ (using Singular Value Decomposition (SVD)).
- A is formed by computing the eigenvectors of R_x. Each eigenvector will be a column of A.

Mean square error approximation

- x can be expressed exactly as a combination of all N basis vectors:

$$x = \sum_{i=0}^{N-1} y(i)a_i, \text{ where } y(i) = a_i^T x$$

- An approximation to x is found by using only m of the basis vectors:

$$\hat{x} = \sum_{i=0}^{m-1} y(i)a_i$$

a projection into the m-dimensional subspace spanned by m eigenvectors

- The PC-transform is based on minimizing the mean square error associated with this approximation.
- The mean square error associated with this approximation is

$$E\left[\|x-\hat{x}\|^2\right] = E\left[\left\|\sum_{i=m}^{N-1} y(i)a_i\right\|^2\right] = E\left[\sum_i \sum_j \left(y(i)a_i^T\right)\left(y(j)a_j\right)\right] =$$
$$\sum_{i=m}^{N-1} E\left[y^2(i)\right] = \sum_{i=m}^{N-1} a_i^T E\left[xx^T\right]a_i$$

- Furthermore, we can find that

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

- The mean square error is thus

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

- The error is minimized if we select the eigenvectors corresponding to the ***m* largest eigenvales of the correlation matrix R_x.**
- The transformed vector y is called the principal components of x. The transform is called the principal component transform or Karhunen-Loeve-transform.

Principal component of the covariance matrix

- Alternatively, we can find the principal components of the covariance matrix Σ_x.
- If we have software for computing principal components of R_x, we can compute principal components from Σ_x by first setting $z=x-\mu_x$ and compute PC(z).

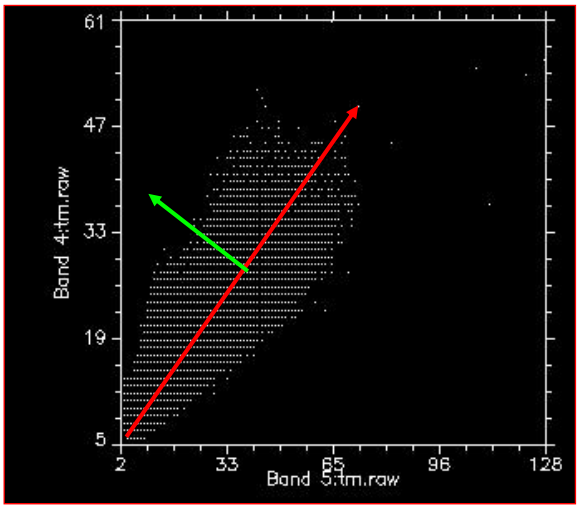
- The principal component transform is not scale invariant, because the eigenvectors are not invariant. Often, normalization to data with zero mean and unit variance is done prior to applying the PC-transform.

Principal components and total variance

- Assume that E[x]=0.
- Let y=PC(x).
- From R_y we know that the variance of component y_j is λ_j.
- The eigenvalues λ_j of the correlation matrix R_x is thus equal to the variance of the transformed features.
- By selecting the *m* eigenvectors with the largest eigenvalues, we select the *m* dimensions with the largest variance.
- The first principal component will be along the direction of the input space which has largest variance.

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.

Principal component images

- For an image with n bands, we can compute the principal component transform of the entire image X.
- Y=PC(X) will then be a new image with n bands, but most of the variance is in the bands with the lowest index (corresponding to the largest eigenvalues).

Principal component images

- For an image with n bands, we can compute the principal component transform of the entire image X.
- Y=PC(X) will then be a new image with n bands, but most of the variance is in the bands with the lowest index (corresponding to the largest eigenvalues).
- Here we use all the pixels in the image to compute the nxn-correlation matrix R_x

PC and compression

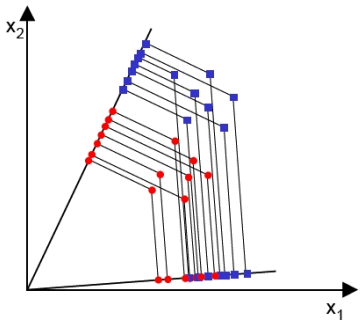
- PC-transform is optimal transform with respect to preserving the energy in the original image.
- For compression purposes, PC-transform is theoretically optimal with respect to maximizing the entropy (from information theory). Entropy is related to randomness and thus to variance.
- The basis vectors are the eigenvectors and vary from image to image. For transmission, both the transform coefficients and the eigenvectors must be transmitted.
- PC-transform can be reasonably well approximated by the Cosinus-transform or Sinus-transform. These use constant basis vectors and are better suited for transmission, since only the coefficients must be transmitted (or stored).

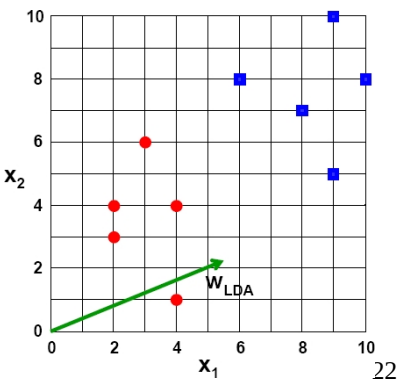
PC vs. Fisher's linear discriminant transform

- The principal component transform has no information about the classes in the data.
- The PC-projection might not be helpful to improve class separability.
- From an input vector x with dimension m, PC-transform gives us a projection y with dimensions 1,…,m (depending on how many eigenvalues we include).
- A projection with Fishers linear discriminant gives us y with dimensions 1,…,K-1, where K is the number of classes.
- Fishers linear discriminant finds the projection that maximizes the ratio of between-class to within-class scatter.

Fisher's Linear Discriminant

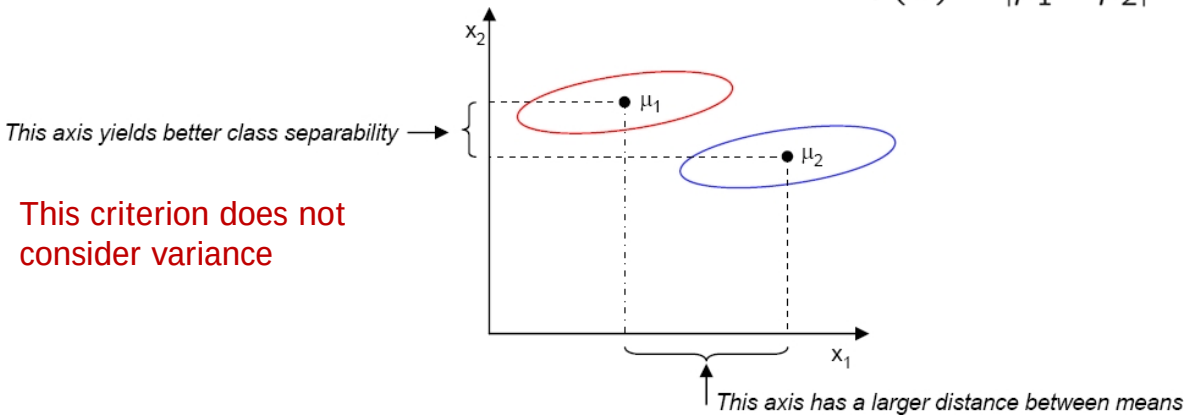
- Goal:
  - Reduce dimension while preserving class discriminatory information
- Strategy (2 classes):
  - We have a set of samples $x=\{x_1, x_2, \ldots, x_n\}$ where n_1 belong to class ω_1 and the rest n_2 to class ω_2. Obtain a scalar value by projecting x onto a line y: $y=w^T x$
  - **Challenge: find w that maximizes the separability of the classes**





A simple criterion function: 2 features and 2 classes

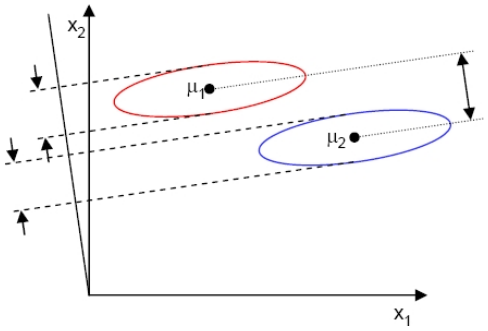
- To find a good projection vector, we need to define a measure of separation between the projections . This wil be the criterion function $J(w)$
- The mean vector of each class in the spaces spanned by x and y are
  $\mu_i = \frac{1}{n_i} \sum_{x \in \omega_i} x$
  $\tilde{\mu}_i = \frac{1}{n_i} \sum_{y \in \omega_i} y = \frac{1}{n_i} \sum_{x \in \omega_i} w^T x = w^T \mu_i$
- A naive choice would be projected mean difference, $J(w) = |\tilde{\mu}_1 - \tilde{\mu}_2|$



This criterion does not consider variance

A criterion function including variance: 2 features and 2 classes

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



Introducing general scatter matrices

- In M-dimensional space, let us now consider matrices describing the variance:
  - Variance INSIDE each class
  - Variance BETWEEN the classes (how well separated are the classes)
  - The total variance in the data set is constant and independent of any class labels

Scatter matrices – M classes

- Within-class scatter matrix:

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

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

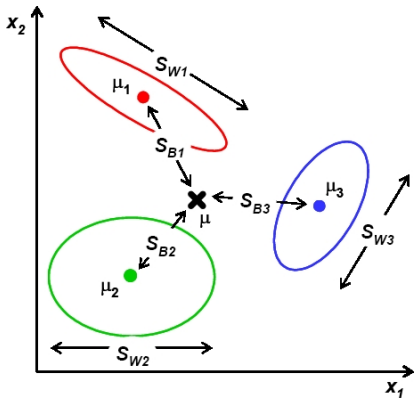
Variance within each class

- Between-class scatter matrix:

$$S_b = \sum_{i=1}^{M} P(\omega_i)(\mu_i-\mu_0)(\mu_i-\mu_0)^T$$

$$\mu_0 = \sum_{i=1}^{M} \mu_i$$

Distance between the classes



- Mixture or total scatter matrix:

$$S_m = E\left[(x-\mu_0)(x-\mu_0)^T\right]$$

Variance of feature with respect to the global mean

Some matrix algebra

- M is a symmetric l×l matrix
- |M| is the determinant of M. It is equal to the product of all eigenvalues of M.
- M can be expressed in terms of eigenvalues λ_i and eigenvectors v_i.
- |M| is nonzero only if the matrix has full rank (all eigenvalues are nonzero)
- trace(M) is equal to the sum of eigenvalues of M.

Relation between eigenvalues and the scatter of a matrix

- The eigenvalues associated with an eigenvector tells how strong the contribution along this direction is.
- A scalar measure of the scatter matrix M is its determinant (the product of the eigenvalues). This gives us ONE measure of the scatter in the matrix.
- If M is a covariance matrix, |M| is a measure of the l-dimensional hypervolume of the data. If the data lies in a subspace, |M| will be zero.
- For a covariance matrix M, trace(M) is the sum of the eigenvalues and thus a measure of the spread or scatter in A.

- The total scatter is
  $S_m = S_w + S_b$
- Consider the criterion function

$$J_1 = \frac{trace\{S_m\}}{trace\{S_w\}}$$ ← Sum of the diagonal elements in S_m

$$\Updownarrow$$

$$\frac{\text{sum of variance around global mean}}{\text{sum of variance inside class}}$$

- J_1 will be large when the variance among the global mean is large compared to the within-class variance.

Better scatter criteria functions – J2 and J3

$$J_2 = \frac{|S_m|}{|S_w|} = |S_w^{-1} S_m|$$

$$J_3 = trace\{S_w^{-1} S_b\}$$

- J_2 and J_3 are invariant to linear transformations.

Fisher's linear discriminant

- Fisher's linear discriminant is a transform that uses the information in the training data set to find a linear combination that best separates the classes.
- It is based on the criterion J_3:

$$J_3 = trace\left\{S_w^{-1} S_b\right\}$$
$$S_w = \sum_{i=1}^{M} P(\omega_i) S_i - \text{within-class scatter}$$
$$S_b = \sum_{i=1}^{M} P(\omega_i)(\mu_i - \mu_0)(\mu_i - \mu_0)^T - \text{between-class scatter}$$

- From the feature vector x, let S_{xw} and S_{xb} be the within-class and between-class scatter matrix.
- The scatter matrices for the transformed variable $y=A^Tx$ are:

$S_{yw}=A^TS_{xw}A$ $S_{yb}=A^TS_{xb}A$

- In subspace y, J_3 becomes:

$$J_3 = trace\left\{\left(A^T S_w A\right)^{-1}\left(A^T S_b A\right)\right\}$$

- Problem: find A such that J_3 is maximized.
- Solution: set

$$\frac{\partial J_3(A)}{\partial A} = 0$$
$$\Updownarrow$$
$$\frac{\partial J_3(A)}{\partial A} = -2S_{xw}A\left(A^T S_{xw}A\right)^{-1}\left(A^T S_{xb}A\right)\left(A^T S_{xw}A\right)^{-1} + 2S_{xb}A\left(A^T S_{xw}A\right)^{-1} = 0$$
$$\Updownarrow$$
$$S_{xw}^{-1}S_{xb}A = A(S_{yw}^{-1}S_{yb})$$

- The scatter matrices S_{yw} and S_{yb} are symmetric, and can thus be diagonalized by the linear transform (appendix B)

$$B^T S_{yw} B = I \text{ and } B^T S_{yb} B = D$$

- B is a l×l matrix, I the identity matrix, and D an l×l diagonal matrix. I and D are the scatter matrices of the transformed vector

$$\hat{y} = B^T y = B^T A^T x$$

- J_3 is invariant under linear transformations and:

$$J_3(\hat{y}) = trace\left\{\left(B^T S_{yw} B\right)^{-1}\left(B^T S_{yb} B\right)\right\} = trace\left\{B^{-1} S_{yw}^{-1} S_{yb} B\right\}$$
$$= trace\{S_{yw}^{-1} S_{yb} B B^{-1}\} = J_3(y)$$

- Furthermore,

$$\left(S_{xw}^{-1} S_{xb}\right) C = CD, \text{ where } C = AB$$

- Because D is diagonal, this is an eigenvalue-problem,
- D must have the eigenvalues of $\left(S_{xw}^{-1} S_{xb}\right)$ on the diagonal
- C must have the corresponding eigenvectors of $\left(S_{xw}^{-1} S_{xb}\right)$ as columns..

- Note that $S_{xb} = \sum_{i=1}^{M} P(\omega_i)(\mu_i - \mu_0)(\mu_i - \mu_0)^T$

  is a sum of M (M=nof. classes) matrices of rank 1, only m-1 of these elements are independent, meaning that the rank of S_{xb} is M-1 or less (no more than M-1 eigenvalues are nonzero).
- This means also that $S_{xw}^{-1}S_{xb}$ has rank M-1 or less.
- Fisher's discriminant transform can give us a l-dimensional projection, where l≤M-1.
  - Note: with 30 features (m=30) and 5 classes (M=5) this gives us a projection with dimension 4 or less.

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

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

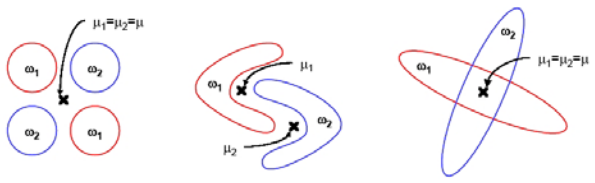
Comments on Fishers discriminant rule

- In general, projection of the original feature vector to a lower dimensional space is associated with some loss of information.
- Although the projection is optimal with respect to J_3, J_3 might not be a good criterion to optimize for a given data set. (Note that J_3 is a kind of sum of a product of between-class and within-class scatter, where the sum is over all classes)
- Minimizing J3 is not equivalent to minimizing the classification error.

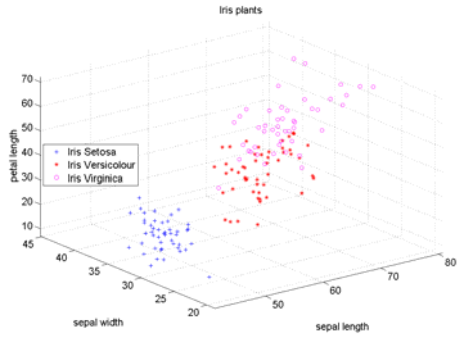
Limitations of Fisher's discriminant

- It produces at most *C-1* feature projections
- It is parametric, since it assumes unimodal gaussian likelihoods

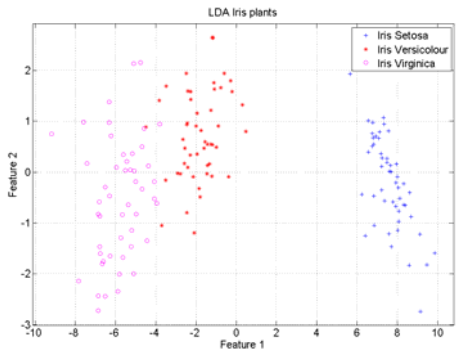
- It will fail when the discriminatory information is not in the mean but in the variance of the data



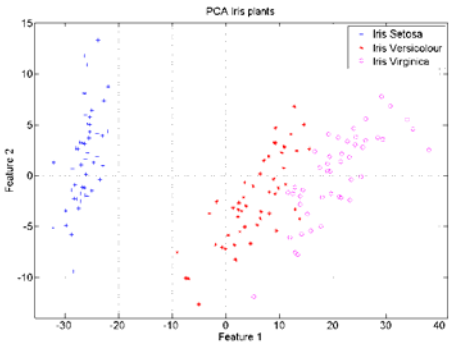
Fisher's discriminant example



Original data



Best 2 Fisher's



Best 2 PCA

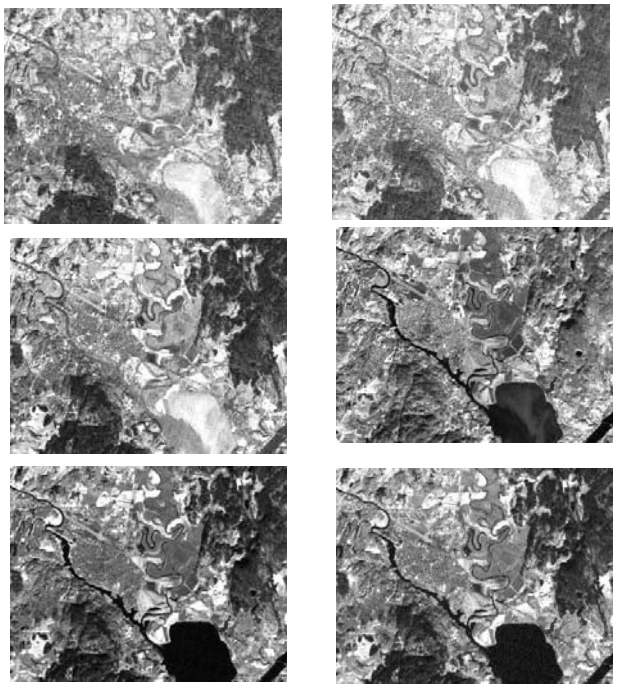
Literature on pattern recognition

- Updated review and statistical pattern recognition:
  - 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.

Data example

- Dimensionality reduction with PCA vs. Fisher

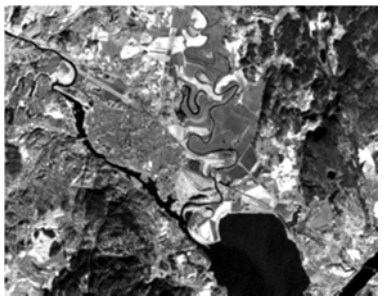
PCA example – original image



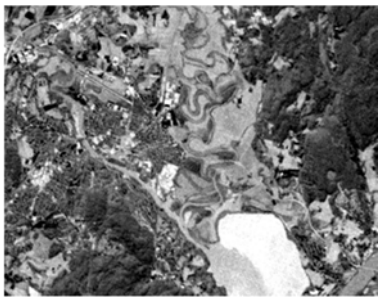
- Satellite image from Kjeller
- 6 spectral bands with different wavelengths

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

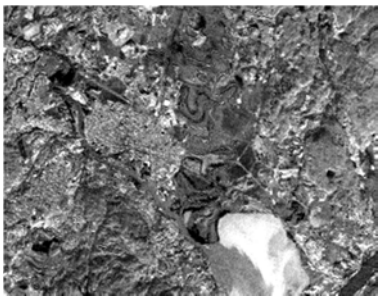
Principal component images



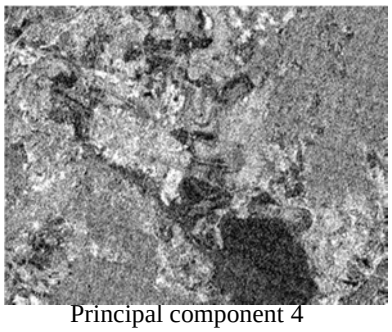
Principal component 1



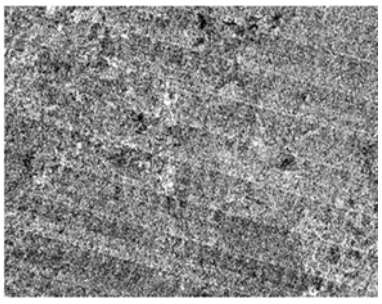
Principal component 2



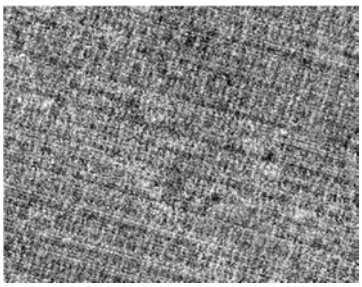
Principal component 3



Principal component 4



Principal component 5

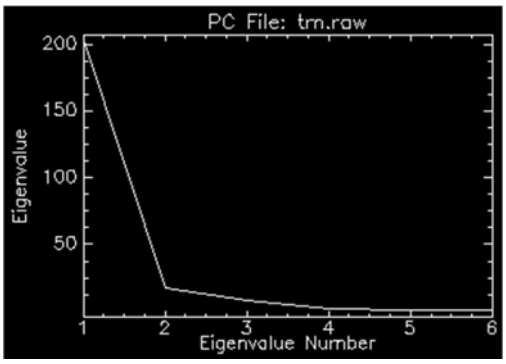


Principal component 6

Example: inspecting the eigenvalues

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

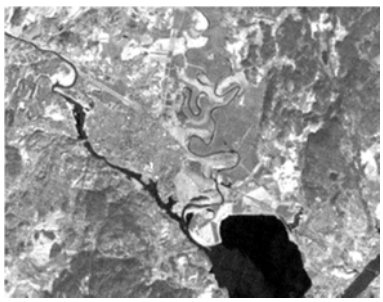
$$E\left[\|x-\hat{x}\|^2\right]=\sum_{i=1}^{N-1}\lambda_i-\sum_{i=1}^{m}\lambda_i=\sum_{i=m}^{N-1}\lambda_i$$



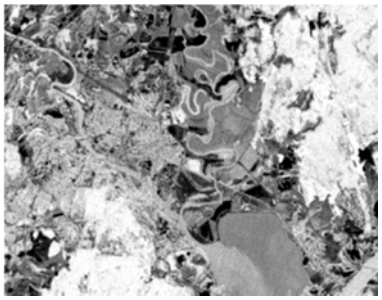
Plotting λ_i

will give indications on how many features are needed for representation

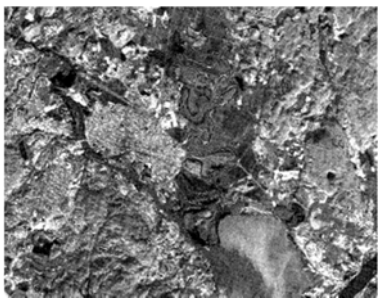
Fisher's linear discriminant on the Landsat image



1. Fisher feature

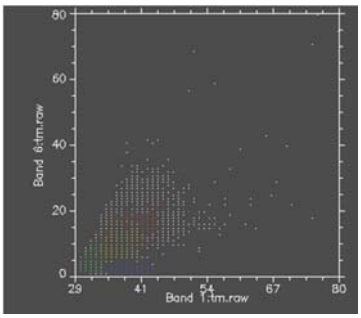


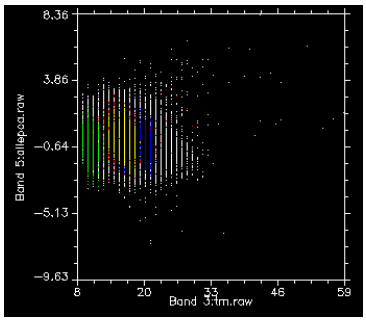
2. Fisher feature

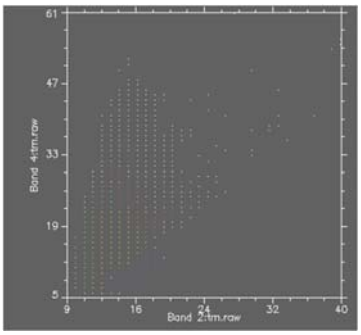


3. Fisher feature

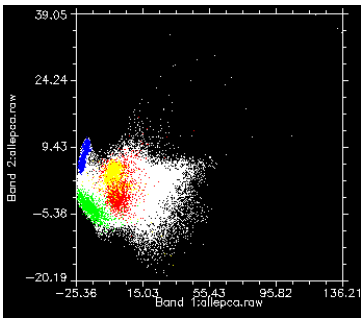
Scatter plots for the example – Original spectral bands

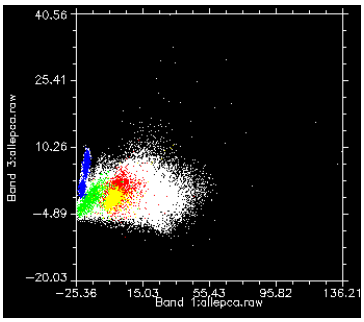


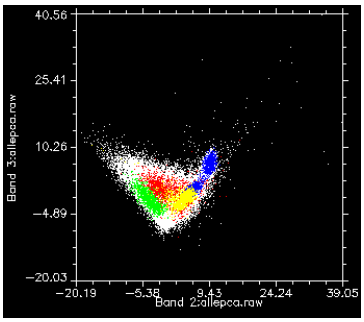




Scatter plots for the example – PCA-components

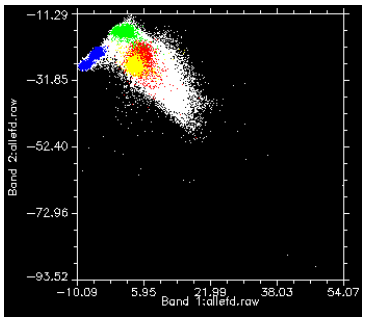




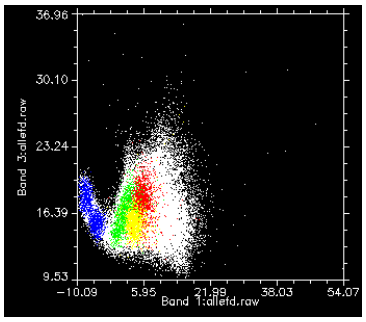


PCA 1 and 2 PCA 1 and 3 PCA 2 and 3

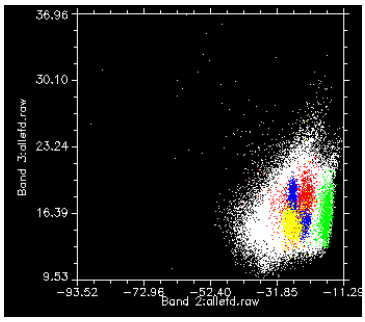
Scatter plots for the example – Fisher-components



Fisher 1 and 2



Fisher 1 and 3



Fisher 2 and 3

Comparison of overall classification accuracy

- All 6 original spectral bands: 91.9% correct classification
- PCA components 1-3: 90.8% correct
- Fisher components 1-3: 91.5% correct