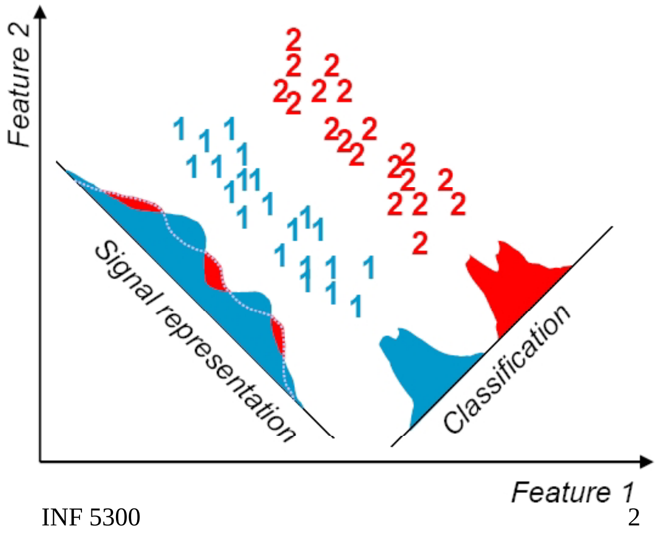
INF 5300
Introduction
Repetition
# of lectures by Anne

Signal representation vs classification

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



Mean square error approximation

- x can be expressed 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.

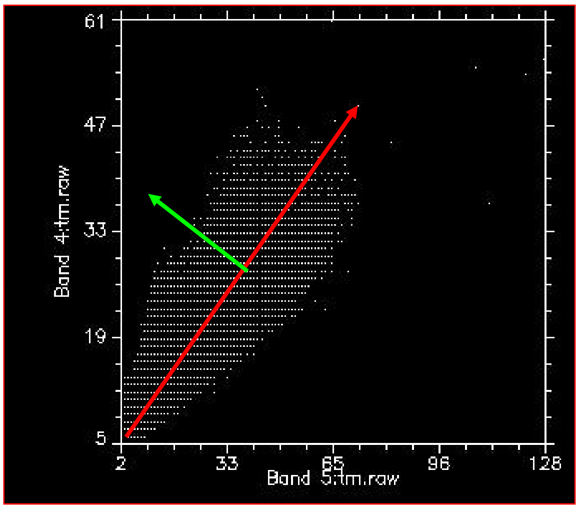
Principal components and total variance

- 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.

Scatter matrices – M classes

- Within-class scatter matrix:

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

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

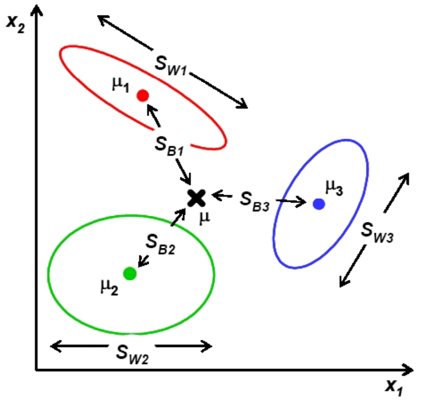
Variance within each class

- 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

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:

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

$$S_w = \sum_{i=1}^{M} P(\omega_i) S_i - \text{within - class scatter}$$

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

- 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

$$\frac{\partial J_3(A)}{\partial A} = 0$$

$$\Updownarrow$$

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

$$\Updownarrow$$

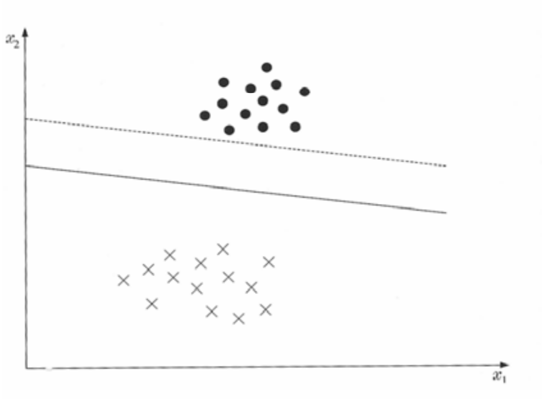
$$S_{xw}^{-1}S_{xb}A = A(S_{yw}^{-1}S_{yb})$$

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.

Can you explain SVM with this?

- There can be many such hyperplanes.
- Which of these two is best, and why?



The optimization problem with margins

- The class indicator for pattern i, y_i, is defined as 1 if y_i belongs to class ω_1 and -1 if it belongs to ω_2.
- The best hyperplane with margin can be found by solving the optimization problem with respect to w and w_0 :

$$\text{minimize} \quad J(w) = \frac{1}{2}\|w\|^2$$

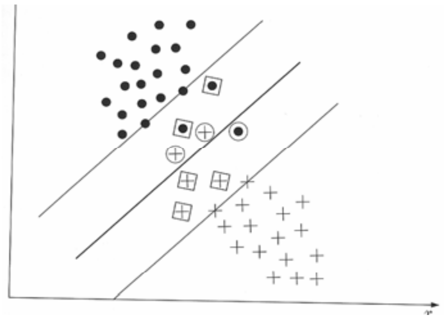
$$\text{subject to} \quad y_i(w^T x_i + w_0) \geq 1, \quad i = 1,2,\ldots N$$

- Checkpoint: do you understand this formulation?
- How is this criterion related to maximizing the margin?

The nonseparable case

- If the two classes are nonseparable, a hyperplane satisfying the conditions $w^Tx-w_0=\pm1$ cannot be found.
- The feature vectors in the training set are now either:

1. Vectors that fall outside the band and are correctly classified.
2. Vectors that are inside the band and are correctly classified. They satisfy $0 \le y_i(w^Tx+w_0)<1$ □
3. Vectors that are misclassified – expressed as $y_i(w^Tx+w_0)<0$ ○



□ Correctly classified

○ Erroneously classified

- The three cases can be treated under a single type of contraints if we introduce slack variables ξ_i:

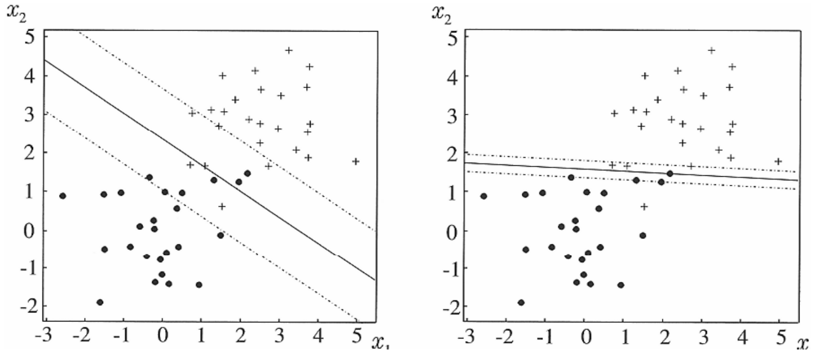
$$y_i[w^T x + w_0] \geq 1 - \xi_i$$

  - The first category (outside, correct classified) have $\xi_i=0$
  - The second category (inside, correct classified) have $0\le \xi_i \le 1$
  - The third category (inside, misclassified) have $\xi_i > 1$

- The optimization goal is now *to keep the margin as large as possible and the number of points with ξ_i >0 as small as possible.*

An example – the effect of C

- C is the misclassification cost.



C=0.2

C=100

- Selecting too high C will give a classifier that fits the training data perfect, but fails on different data set.
- The value of C should be selected using a separate validation set. Separate the training data into a part used for training, train with different values of C and select the value that gives best results on the validation data set. Then apply this to new data or the test data set. (explained later)

The optimization problem with margins

- The class indicator for pattern i, y_i, is defined as 1 if y_i belongs to class ω_1 and -1 if it belongs to ω_2.
- The best hyperplane with margin can be found by solving the optimization problem with respect to w and w_0 :

$$\text{minimize} \quad J(w) = \frac{1}{2}\|w\|^2$$

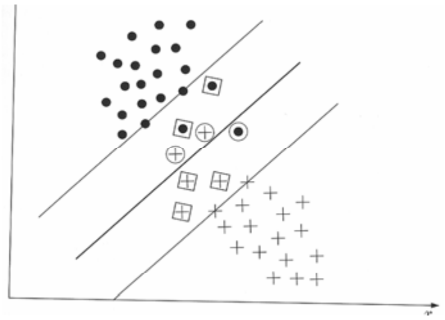
$$\text{subject to} \quad y_i(w^T x_i + w_0) \geq 1, \quad i = 1,2,\ldots N$$

- Checkpoint: do you understand this formulation?
- How is this criterion related to maximizing the margin?

The nonseparable case

- If the two classes are nonseparable, a hyperplane satisfying the conditions $w^Tx-w_0=\pm 1$ cannot be found.
- The feature vectors in the training set are now either:

1. Vectors that fall outside the band and are correctly classified.
2. Vectors that are inside the band and are correctly classified. They satisfy $0 \leq y_i(w^Tx+w_0)<1$ □
3. Vectors that are misclassified – expressed as $y_i(w^Tx+w_0)<0$ ○



□ Correctly classified

○ Erroneously classified

- The three cases can be treated under a single type of contraints if we introduce slack variables ξ_i:

$$y_i[w^T x + w_0] \geq 1 - \xi_i$$

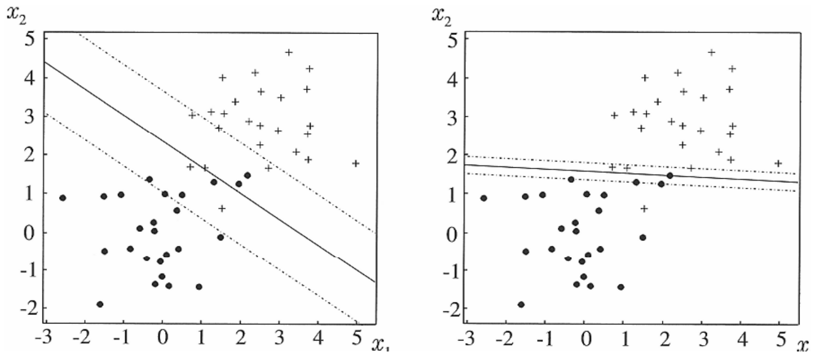
  - The first category (outside, correct classified) have $\xi_i=0$
  - The second category (inside, correct classified) have $0\leq \xi_i \leq 1$
  - The third category (inside, misclassified) have $\xi_i >1$

- The optimization goal is now *to keep the margin as large as possible and the number of points with ξ_i >0 as small as possible.*

An example – the effect of C

- C is the misclassification cost.

C=0.2



C=100

- Selecting too high C will give a classifier that fits the training data perfect, but fails on different data set.
- The value of C should be selected using a separate validation set. Separate the training data into a part used for training, train with different values of C and select the value that gives best results on the validation data set. Then apply this to new data or the test data set. (explained later)

Snakes - The energy function

$$E_{snake} = \int_{s=0}^{1} E_{\text{int}}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))ds$$

Internal deformation energy of the snake itself.
How it can bend and stretch.

Constraints on the shape of the snake. Enchourages the contour to be smooth. (Often omitted)

A term that relates to gray levels in the image, e.g. attracts the snake to points with high gradient magnitude.

The minimum values is found by derivation:

$$\frac{dE_{snake}}{dv} = 0$$

The internal deformation term

$$E_{\text{int}} = \alpha(s)\left|\frac{dv(s)}{ds}\right|^2 + \beta(s)\left|\frac{d^2v(s)}{ds^2}\right|^2$$

First derivative

Measures how stretched the contour is.
Keyword: point spacing.
Imposes tension.
The curve should be short if possible.
Physical analogy: v acts like a membrane.

Second derivative

Measures the curvature or bending energy.
Keyword: point variation.
Imposes rigidity.
Changes in direction should be smooth.
Physical analogy: v acts like a thin plate.

α and β are penalty parameters that control the weight of the two terms.
Low α values: the snake can stretch much.
Low β values: the snake can have high curvature.

The energy function

- Simple snake with only two terms (no termination energy):

$$E_{snake}(s) = E_{\text{int}}(v_s) + E_{image}(v_s)$$
$$= \alpha\left|\frac{dv_s}{ds}\right|^2 + \beta\left|\frac{d^2v_s}{ds^2}\right|^2 + \gamma E_{edge}$$

- We need to approximate both the first derivative and the second derivative of v_s, and specify how E_{edge} will be computed.
- How should the snake iterate from its initial position?

Approximating the first derivative of v_s

Average distance between points on the contour

Distance between this point and the next point

$$\left|\frac{dv_s}{ds}\right|^2 = \left|\sum_{i=0}^{S-1}\left\|v_i - v_{i+1}\right\| / S - \left\|v_s - v_{s+1}\right\|\right|$$

$$= \left|\sum_{i=0}^{S-1}\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \,/\, S - \sqrt{(x_s - x_{s+1})^2 + (y_s - y_{s+1})^2}\right|$$

Approximating the second derivative of v_s

Why is this correct?
Hint: Check the derivation of the Laplace operator

$$\left|\frac{d^2 v_s}{ds^2}\right|^2 = \left|(v_{s+1} - 2v_s + v_{s-1})\right|^2$$

$$= (x_{s+1} - 2x_s + x_{s-1})^2 + (y_{s+1} - 2y_s - y_{s-1})^2$$

The Kass differential equations

- The coordinates of the snake should be found by solving the differential equations iteratively:

$$-\frac{d}{ds}\left\{\alpha(s)\frac{d\hat{x}(s)}{ds}\right\}+\frac{d^2}{ds^2}\left\{\beta(s)\frac{d^2\hat{x}(s)}{ds^2}\right\}+\frac{1}{2}\int_{s=0}^{1}\left.\frac{\partial E_{edge}}{\partial x}\right|_{\hat{x},\hat{y}}=0$$

$$-\frac{d}{ds}\left\{\alpha(s)\frac{d\hat{y}(s)}{ds}\right\}+\frac{d^2}{ds^2}\left\{\beta(s)\frac{d^2\hat{y}(s)}{ds^2}\right\}+\frac{1}{2}\int_{s=0}^{1}\left.\frac{\partial E_{edge}}{\partial y}\right|_{\hat{x},\hat{y}}=0$$

- The iterative solution was given by

$$x^{<i+1>}=(A+\lambda I)^{-1}\left(\lambda x^{<i>}+f_x(x^{<i>},y^{<i>})\right)$$
$$y^{<i+1>}=(A+\lambda I)^{-1}\left(\lambda y^{<i>}+f_y(x^{<i>},y^{<i>})\right)$$

- λ is a step size

Capture range problems

- The problem stems from the short ”range” of the external fources.
- The inverse magnitude of the gradient will have significant values only in the vicinity of the salient edges.
- This basically forces us to initialize the snake very close to the target contour.
- This problem is know as the **capture range problem.**

Capture range problems

- Xu and Prince define the vector field:

$$\mathbf{v}(x,y) = (u(x,y), v(x,y))^T$$

- It is **v** that will be the GVF.
- The field **v** is the field that minimizes the following functional:

$$G = \iint \mu\left(u_x{}^2 + u_y^2 + v_x^2 + v_y^2\right) + \left|\nabla f\right|^2 \left|v - \nabla f\right|^2 dxdy$$

- v(x,y) is found by solving this equation.
- μ is a parameter that controls the amount of smoothing.

- Where have you seen the first term before (in this course)?

Capture range problems

$$G = \iint \mu\left(u_x{}^2 + u_y^2 + v_x^2 + v_y^2\right) + \left|\nabla f\right|^2 \left|v - \nabla f\right|^2 dxdy$$

- The goal is to minimize G.
- The second term will have a minimum if v=∇f.
- If ∇f is small, the first term will dominate.
- This can also be written as

$$\mu\nabla^2 u - \left(u - f_x\right)\left(f_x^2 + f_x^2\right) = 0$$
$$\mu\nabla^2 v - \left(v - f_y\right)\left(f_x^2 + f_x^2\right) = 0$$

- If ∇f is small, what remains is Lagrange's equation:

$$\mu\nabla^2 u = 0$$
$$\mu\nabla^2 v = 0$$

Capture range problems

- The first term will smooth the data, that is, far from edges the field will be kept as smooth as possible by imposing that the spatial derivatives be as small as possible.
- When $|\nabla f|$ is small, the vector field will be dominated by the partial derivatives of the vector field, yielding a smooth field.
- Close to edges (where $|\nabla f|$ is large) the field is forced to resemble the gradient of f itself.
- So **v** is smooth far from edges and nearly equal to the gradient of f close to edges.
- The term μ just defines the weight we give the different terms in the functional.
- The field **v** is computed iteratively

$$G = \iint \mu\left(u_x{}^2 + u_y^2 + v_x^2 + v_y^2\right) + \left|\nabla f\right|^2 \left|v - \nabla f\right|^2 dxdy$$

$$u_{i,j}^{n+1} = \left(1 - b_{i,j}\Delta t\right)u_{i,j}^{n} + r\left(u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1} - 4u_{i,j}\right) + c_{i,j}^{1}\Delta t$$

$$v_{i,j}^{n+1} = \left(1 - b_{i,j}\Delta t\right)v_{i,j}^{n} + r\left(v_{i+1,j} + v_{i,j+1} + v_{i-1,j} + v_{i,j-1} - 4v_{i,j}\right) + c_{i,j}^{2}\Delta t$$

Gradient magnitude

Laplacian term

A term in the gradient direction