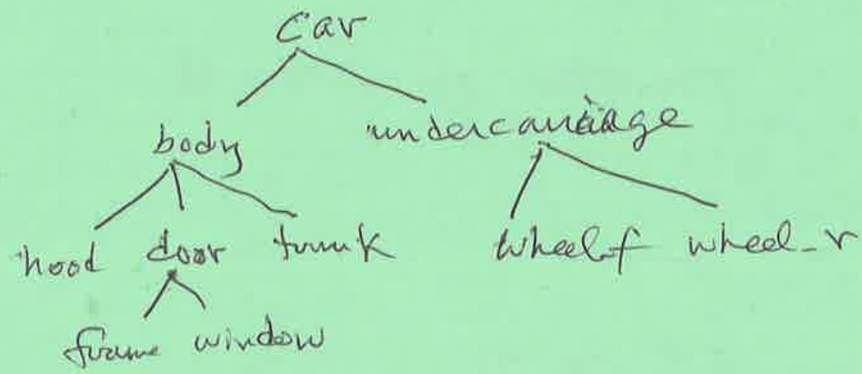
CS6640 Week 13

pattern vectors : points in space

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

structural patterns : grammars



Min distance classifier

prototypes: $\bar{m}_j = \frac{1}{n_j} \sum_{\bar{x} \in c_j} \bar{x} \quad j = 1, 2, \ldots, N_c$

$D_j(\bar{x}) = \|\bar{x} - \bar{m}_j\| \quad j = 1, \ldots, N_c$

$\|\bar{a}\| = (\bar{a}^T \bar{a})^{1/2}$

$\bar{x} \rightarrow c_i$ if $D_i(\bar{x}) \leq D_j(\bar{x}) \; \forall j$

or use

$d_j(\bar{x}) = \bar{m}_j^T \bar{x} - \frac{1}{2} \bar{m}_j^T \bar{m}_j \quad j = 1, \ldots, N_c$

& $\bar{x} \rightarrow c_i$ if $d_i(\bar{x}) \geq d_j(\bar{x}) \; \forall j \neq i$

decision or discriminant functions

<u>decision boundary</u>: between c_i & c_j:

$$d_i(\bar{x}) = d_j(\bar{x})$$

$$\Rightarrow \quad d_{ij}(\bar{x}) = d_i(\bar{x}) - d_j(\bar{x})$$

$$= (\bar{m}_i - \bar{m}_j)^T \bar{x} - \frac{1}{2}(\bar{m}_i - \bar{m}_j)^T(\bar{m}_i + \bar{m}_j) = 0$$

This is a hyperplane (line in 2D, plane in 3D

see example 13\2a

Optimum statistical classifier (Bayes)

<u>Conditional probability</u> $p(c_i | \bar{x})$

probability of class c_i given $\bar{x}$

if classifier makes a mistake:

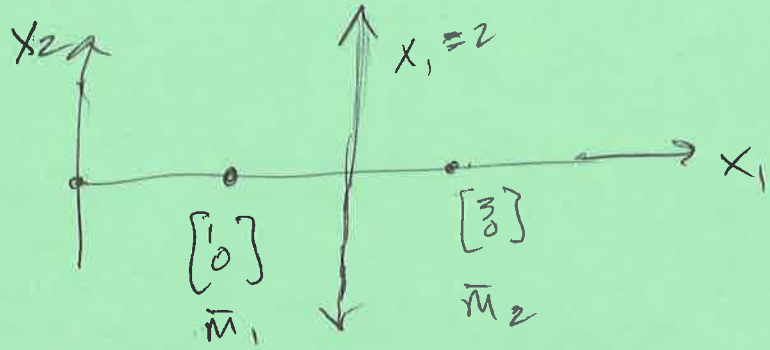
picks c_j when c_i was correct:

incurs loss L_{ij}

conditional average risk (loss):

$$r_j(\bar{x}) = \sum_{k=1}^{N_c} L_{kj}\, p(c_k | \bar{x})$$

13|2a



$$d_{12}(\bar{x}) = \left(\begin{bmatrix}1\\0\end{bmatrix} - \begin{bmatrix}3\\0\end{bmatrix}\right)^T \bar{x} - \frac{1}{2}\left(\begin{bmatrix}1\\0\end{bmatrix} - \begin{bmatrix}3\\0\end{bmatrix}\right)^T \left(\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}3\\0\end{bmatrix}\right) = 0$$

$$0 = [-2 \ \ 0]\,\bar{x} - \frac{1}{2}\left([-2 \ \ 0]\begin{bmatrix}4\\0\end{bmatrix}\right) = 0$$

$$0 = -2x_1 - \frac{1}{2}(-8)$$

$$0 = -2x_1 + 4$$

$$\Rightarrow \quad x_1 = 2$$

Using Bayes Rule:

$$P(A|B) = \frac{P(A|B)\,P(A)}{P(B)}$$

get:

$$r_j(\bar{x}) = \frac{1}{p(\bar{x})} \sum_{k=1}^{N_c} L_{kj}\, P(\bar{x}|C_k)\, \underbrace{P(C_k)}_{\substack{\text{a priori} \\ \text{prob. of class } C_k}}$$

To compare r_j's (i.e., to choose lowest risk class):
choose lowest:

$$r_j(\bar{x}) = \sum_{k=1}^{N_c} L_{kj}\, P(\bar{x}|C_k)\, P(C_k)$$

get rid of $\frac{1}{p(\bar{x})}$ since it is common factor

if each decision chooses lowest risk, then total average loss over all decisions is minimal

<u>Bayes classifier</u>

$$\bar{x} \to C_i \text{ if } r_i(\bar{x}) \leq r_j(\bar{x}) \quad \forall j \neq i$$

Use special loss function:

$$L_{ij} = 1 - \delta_{ij} \quad \begin{cases} \delta_{ij} = 1 & \text{if } i = j \\ \delta_{ij} = 0 & \text{else} \end{cases}$$

So,

$$r_j(\bar{x}) = \sum_{k=1}^{N_c} (1-\delta_{kj})\, p(\bar{x}|C_k)\, P(C_k)$$

$$= p(\bar{x}) - p(\bar{x}|C_j)\, P(C_j) \qquad ?$$

Because:

$$p(\bar{x}) = \sum_{k=1}^{N_c} p(\bar{x}|C_k)\, P(C_k)$$

and
$$r_j(\bar{x}) = \sum_{k=1}^{j-1} p(\bar{x}|C_k)\, P(C_k) + \sum_{k=j+1}^{N_c} p(\bar{x}|C_k)\, P(C_k)$$

$$\Rightarrow \; r_j(\bar{x}) = p(\bar{x}) - p(\bar{x}|C_j)\, P(C_j)$$

Bayes classifier:

$$\bar{x} \rightarrow C_i \quad \text{if } \forall j \neq i$$

$$p(\bar{x}) - p(\bar{x}|C_i)\, P(C_i) \leq p(\bar{x}) - p(\bar{x}|C_j)\, P(C_j)$$

$$\Rightarrow \; p(\bar{x}|C_i)\, P(C_i) \geq p(\bar{x}|C_j)\, P(C_j)$$

Must know probabilities ; Assume Gaussian

Consider 1D with 2 classes $(N_c = 2)$

means: m_1, m_2 variances: σ_1^2, σ_2^2

$$d_j(x) = p(x|C_j)\, P(C_j)$$

$$= \frac{1}{\sqrt{2\pi}\,\sigma_j}\, e^{-\frac{(x-m_j)^2}{2\sigma_j^2}}\, P(C_j)$$

when $p(c_1) = p(c_2)$, then

↑ decision point (like hyperplane)

N - D

$$p(\bar{x} \mid c_j) = \frac{1}{(2\pi)^{n/2} |C_j|^{1/2}} e^{-\frac{1}{2}(\bar{x} - \bar{m}_j)^T C_j^{-1} (\bar{x} - \bar{m}_j)}$$

$\bar{m}_j$ mean for class j $= \frac{1}{n_j} \sum_{\bar{x} \in c_j} \bar{x}$

C_j covariance matrix

$$\bar{C}_j = \frac{1}{n_j} \sum_{\bar{x} \in c_j} \overline{x x^T} - \bar{m}_j \bar{m}_j^T$$

use ln instead of e

$$d_j(\bar{x}) = \ln \left[p(\bar{x} \mid c_j) p(c_j) \right]$$

$$= \ln p(\bar{x} \mid c_j) + \ln p(c_j)$$

$$= \ln p(c_j) - \underbrace{\frac{n}{2} \ln 2\pi}_{\text{constant} \to \text{eliminate}} - \frac{1}{2} \ln |\bar{C}_j| - \frac{1}{2} \left[(\bar{x} - \bar{m}_j)^T \bar{C}_j^{-1} (\bar{x} - \bar{m}_j) \right]$$

$$d_j(\bar{x}) = \ln(P(C_j)) - \frac{1}{2}\ln|C_j| - \frac{1}{2}\left[(\bar{x}-\bar{m}_j)^T C_j^{-1}(\bar{x}-\bar{m})\right]$$

give: Bayes decision functions under 0-1 loss
quadratic decision surfaces

if $C_i = C_j \quad \forall i, j$

$$d_j(\bar{x}) = \ln P(c_j) + \bar{x}^T C^{-1}\bar{m}_j - \frac{1}{2}\bar{m}_j^T C^{-1}\bar{m}_j$$

linear decision surfaces

if $C = I$ and $P(c_i) = P(c_j)$

$$\Rightarrow \quad d_j(\bar{x}) = \bar{m}_j^T \bar{x} - \frac{1}{2}\bar{m}_j^T \bar{m}_j$$ (see p. 13\1)

Pattern Recognition (from Duda + Hart)

input
↓
sensing
↓
segmentation
↓
feature extraction
↓
classification ← missing features
context
↓
post-processing ← costs

Feature ~~Extraction~~

distinguish object
invariant to sensing modality + presentation
translation, rotation, scale, occlusion
projective distortion

Classification

variability
noise
robustness

Post-Processing

Error Rate
Risk
context
multiple classifiers

PR Design Cycle

Data collection

typical examples of what classes
prototype robustness & effectiveness

Feature Choice

stable
distinguishing

Model Choice

validity
ease of use

Training

e.g., find means & covariances

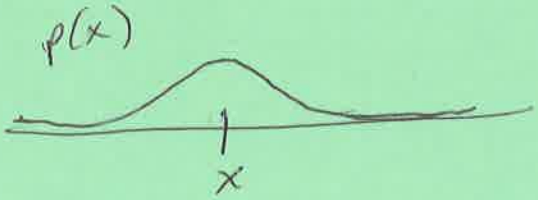
Evaluation

Apply to test data & gather statistics

start
↓
collect data
↓
choose features
↓
choose model
↓
train classifier
↓
evaluate classifier
↓
end

State Estimation

represent beliefs by parameterized method



$\mathcal{N}(\mu, \sigma^2)$

At time t, state $\vec{x}$ is : $\bar{\mu}_t$ Σ_t

multi-dimensional

for linear system:

$$\bar{x}_{t+1} = \sum a_i x_{i,t}$$

1. state transition probability $p(\bar{x}_t \mid \bar{u}_t, \bar{x}_{t-1})$:

$$\underset{\text{new state vector}}{\bar{x}_t} = A_t \underset{\text{old state vector}}{\bar{x}_{t-1}} + \underbrace{B\bar{u}_t}_{\text{control}} + \underbrace{\varepsilon_t}_{\text{noise}}$$

$$\bar{x}_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ \vdots \\ x_{n,t} \end{bmatrix} \qquad \bar{u}_t = \begin{bmatrix} u_{1,t} \\ u_{2,t} \\ \vdots \\ u_{m,t} \end{bmatrix}$$

A is $n \times n$, B is $n \times m$

$$\varepsilon \sim \mathcal{N}(0, R_t)$$

process noise variance

13\10

mean of posterior state:

$$p(\bar{x}_t \mid \bar{u}_t, \bar{x}_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}}$$
$$\exp\left\{-\frac{1}{2}(\bar{x}_t - A_t \bar{x}_{t-1} - B_t \bar{u}_t)^T R_T^{-1} (\bar{x}_t - A_t \bar{x}_{t-1} - B_t \bar{u}_t)\right\}$$

2. measurement prob. linear:

$$\bar{z}_t = C_t \bar{x}_t + \bar{\delta}_t$$

C_t is $k \times n$ $\qquad \delta \sim N(0, Q_t)$

measurement prob:

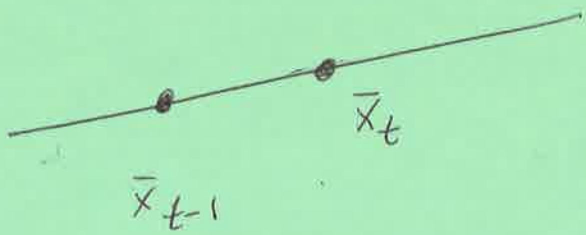
$$p(\bar{z}_t \mid \bar{x}_t) = \det(2\pi Q_t)^{-1/2} \exp\left\{-\frac{1}{2}(\bar{z}_t - C_t \bar{x}_t)^T Q_t^{-1} (\bar{z}_t - C_t \bar{x}_t)\right\}$$

3. Initial belief is normal: μ_0, Σ_0

$$bel(\bar{x}_0) = p(\bar{x}_0) = \det(2\pi \Sigma_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\bar{x}_0 - \bar{\mu}_0)^T (\bar{x}_0 - \bar{\mu}_0)\right\}$$

Example

Track linear motion



Suppose constant velocity
v_x, v_y
Time step Δt

$$\bar{x}_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

View V_x, V_y as control

$$x_{1,t+1} = x_{1,t} + V_x \Delta t$$
$$x_{2,t+1} = x_{2,t} + V_y \Delta t$$

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}$$

$$\bar{x}_{t+1} = A \quad \bar{x}_t \quad + \quad B \quad \bar{u}_t \quad + \quad \varepsilon$$

$$\begin{bmatrix} x_{1,t+1} \\ x_{2,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,t} \\ y_{1,t} \end{bmatrix} + \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \varepsilon$$

$\sim N(0, R)$

Need a sensor model:

$$\bar{z}_t = C \bar{x}_t + \delta$$

$$\begin{bmatrix} z_{1,t} \\ z_{2,t} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} + \delta$$

$\sim N(0, Q)$

Main idea

$\bar{X}_0 \leftarrow$ initialize (how? $\rightarrow$ sensors 1^{st} frames)

while <tracking>

<get data>

$\bar{z}_t \leftarrow$ from sensor data

$\bar{x} \leftarrow$ KalmanFilter(.)

end

KalmanFilter$(\bar{\mu}_{t-1}, \Sigma_{t-1}, \bar{u}_t, \bar{z}_t)$

$$\bar{\mu}_t = A_t \bar{u}_{t-1} + B_t \bar{u}_t$$

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t (\vec{z}_t - C_t \bar{\mu}_t)$$

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

$\Rightarrow$ need A_t, B_t, R_t, C_t, Q_t as inputs