Notiztitel 17.10.2008

28-5-2014

More advanced distance measures:

$$d(\underline{x}, \underline{y}) = (\underline{x} - \underline{y})^T \cdot C^{-1} \cdot (\underline{x} - \underline{y})$$

C → covariance matrix of training data

→ **Mahalanobis-distance**

Further options:

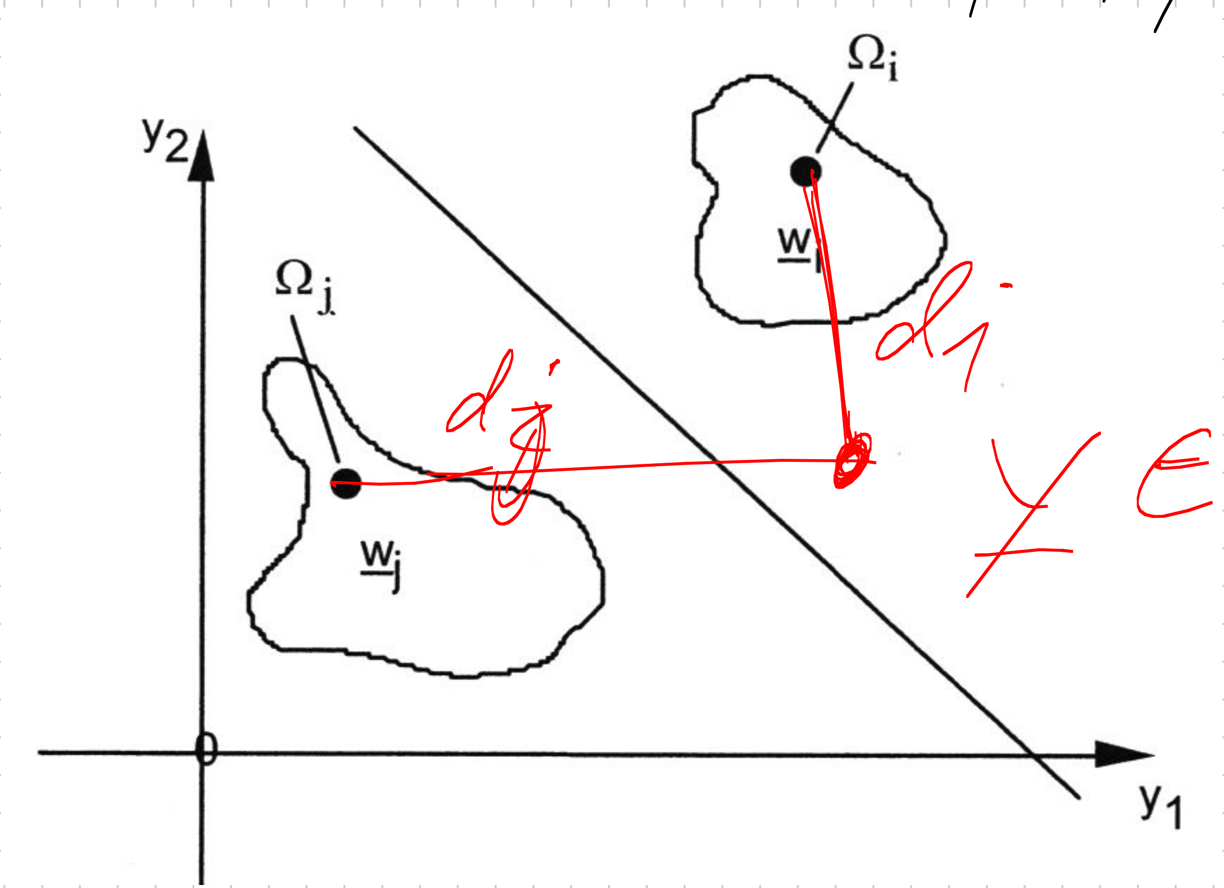
Scalar product of $\underline{v}$ and $\underline{y}$:

$$d(\underline{v}, \underline{y}) = \underline{v}^T \cdot \underline{y} = |\underline{v}| \cdot |\underline{y}| \cdot \cos(\angle(\underline{v}\underline{y}))$$

$\Rightarrow$ danger of misclassification because of different lengths for vectors $\underline{v}$

therefore: use normalized distance

$$d(\underline{w}, \underline{y}) = \frac{\underline{w}^T \cdot \underline{y}}{|\underline{w}| \cdot |\underline{y}|} = \cos(\measuredangle\, \underline{w}, \underline{y})$$



$\underline{y} \in \Omega_i$

$\Rightarrow$ NN-Rule (nearest neighbor-rule)

More advanced classification scheme:

KNN-Rule:

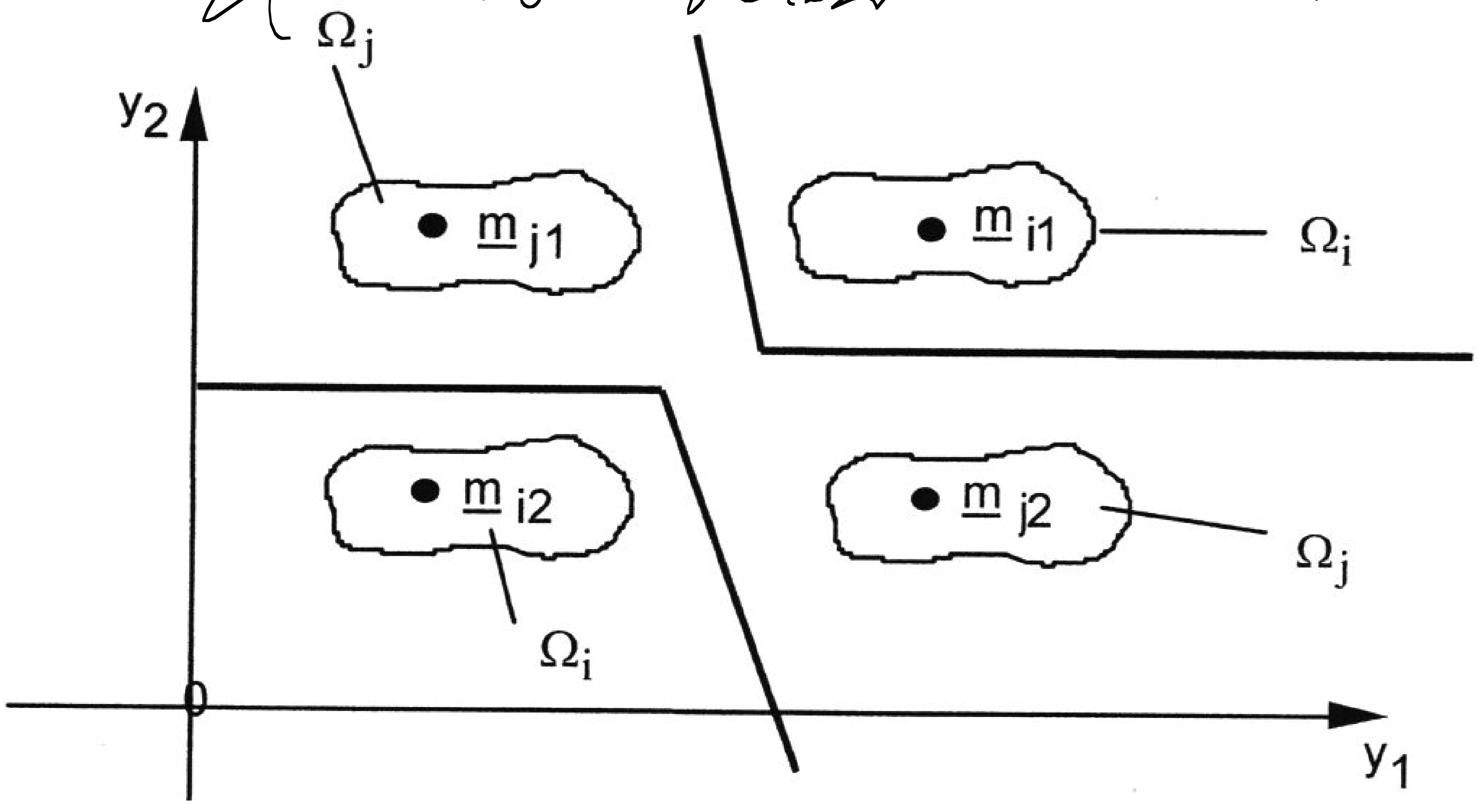
idea: represent each class not by one single reference vector, but instead by several reference vectors (L reference vectors)

Classification procedure:

- Compare unknown vector y with all $L \cdot M$ different reference vectors
- find the K nearest reference vectors (K nearest neighbors)
- determine class Ω_m^* that is most frequently represented in the K nearest neighbors. Count the number of occurrences of this class Ω_m^* and call this count K'

- assign vector $\underline{y}$ to class Ω_m^* ($\rightarrow$ majority vote)

- Option: reject vector $\underline{y}$ if k' is below a certain threshold



3.2.2 Classification with decision functions

re-consider formula for classical distance function:

$$d = (\underline{w} - \underline{y})^T \cdot (\underline{w} - \underline{y}) = \sum_{i=1}^{N} (w_i - y_i)^2$$

$$= \underline{w}^T \cdot \underline{w} - 2\underline{w}^T \cdot \underline{y} + \underline{y}^T \cdot \underline{y}$$

$$= \underline{y}^T \underline{y} - 2\left(\underline{u}^T \underline{y} - \frac{1}{2}\underline{u}^T \underline{u}\right)$$

Classification rule (NN-rule):

$$\Omega_m = \min_m \{ d(\underline{w}_m, \underline{y}) \}$$

winning class

$$\Rightarrow \text{now}: \Omega_m = \max_m \left\{ \underline{w}_m^T \underline{y} - \frac{1}{2}\underline{w}_m^T \underline{w}_m \right\}$$

$$\hat{=} \max_m \{ d_m(\underline{w}_m, \underline{y}) \}$$

decision function

Now, a more efficient notation can be introduced:

$$\underline{w}_m^* = \left[w_{m1}, w_{m2}, \ldots, w_{mN}, -\frac{1}{2}\underline{w}_m^T \cdot \underline{w}_m \right]$$

$$\underline{y}^* = \left[\underline{y}^T, 1 \right]^T$$

decision function: $d_m = \underline{w}_m^{*T} \cdot \underline{y}^*$

(mostly in future we will omit the #-notation)

$\Rightarrow$ novel classification rule, which is based on decision functions:

$$\Omega_m = \max_m \{ d(\underline{w}_m, \underline{y}) \}$$

winning class

$$d(\underline{w}_m, \underline{y}) = \underline{w}_m^T \cdot \underline{y}$$

Boundary between 2 classes, Ω_i and Ω_j;

$$d_i(\underline{w}_i, \underline{y}) = d_j(\underline{w}_j, \underline{y})$$

$$d_i(\underline{w}_i, \underline{y}) - d_j(\underline{w}_j, \underline{y}) = 0$$

$$\underline{w}_i^T \underline{y} - \underline{w}_j^T \underline{y} = 0$$

$$(\underline{w}_i^* - \underline{w}_j^*)^T \cdot \underline{y}^* = 0$$

$$\sum_{k=1}^{N} (w_{ik} - w_{jk}) \cdot y_k - \frac{1}{2}(\underline{w}_i^T \underline{w}_i - \underline{w}_j^T \underline{w}_j) = 0$$

$\Longrightarrow$ equation for decision boundary

Case 2 dimensions: $N = 2$

$$\alpha \cdot y_1 + \beta \cdot y_2 = \gamma \quad (\text{Line})$$

3.2.2.2 Generalized decision functions