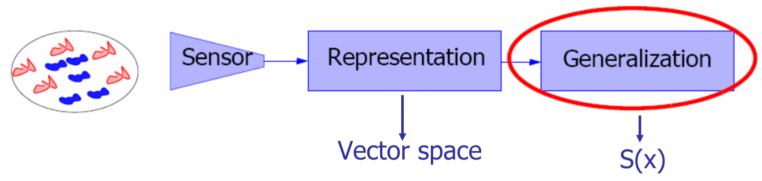
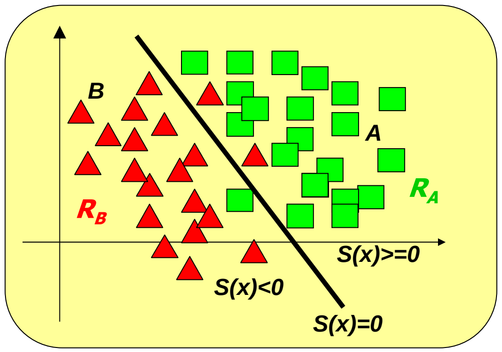
Classification, Discriminant Analysis





Supervised learning: how to learn S(x) from examples?

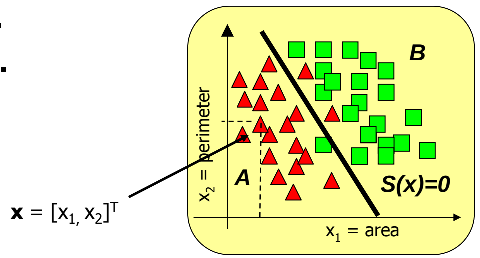
Contents

Some formalism

- Objects are observed by sensors → **numerical representation**.
- Numbers encode information on objects, e.g. their characteristics (partial, individual, combined) or degrees of pairwise similarities.
- We often derive features from raw measurements (perimeter, weight) or preprocessed measurements (curvature, response of filters in images).
- Features are dimensions in a (Euclidean) *feature vector space* **X**. Objects are described as feature vectors; *k*-dimensional **vector** is $\mathbf{x} = [x_1, x_2, \ldots, x_k]^T$.
- Given *training objects* and their class *labels* from Y, we look for a decision function that discriminates between the classes. For two classes: **S(x) = 0.**
- Classifier is a function **F: X → Y.**

  Two classes: **F(x) = sign(S(x)).**

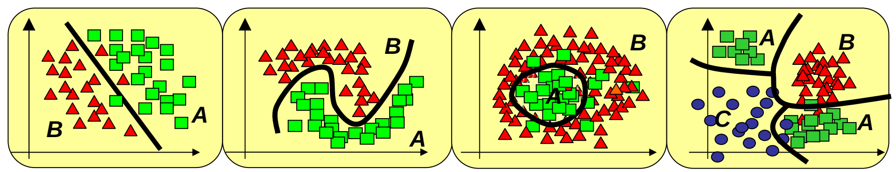
  Labels {1,-1}.



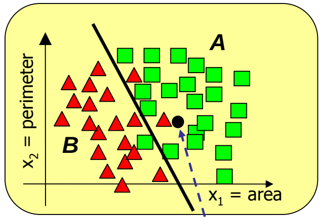
The problem of classification

Classification: learn a decision rule from the **training data** that assigns an object **x** to one of the K classes.

- ***K* classes:** ω_j, j=1,..,K, labeled by y_i (*+additional rejection class* ω_{rej} *if used*).
- **Training data: {x_i, y_i},** i=1,…,n. $\boldsymbol{x}_i$ in $\boldsymbol{X} = \mathbf{R}^k$ is a k-dimensional vector representing the *i*-th object, y_i is the label.
- **Decision rule S(x):** Partitions vector space into K (not necessarily compact) regions R_j, j=1,…,*K*, corresponding to the classes ω_j.
- **Decision boundaries:** Boundaries between the regions R_j.
- **Overlap:** If overlap exists, there is usually no perfect S(**x**). **There are various techniques to find the suboptimal decision rule.**



Classification principles



How to classify this object?

- **Class A**, as it has the highest density for A.
- **Class B**, as it is the closest to an object from B.
- **Class A**, as it is on the A-side of the linear minimum error classifier.

The task of classification is ill-posed!

Principles:

- **Generative classifiers:** focus on each class separately: model class conditional densities (likelihoods) and reason about discrimination
- **Discriminative classifiers:** focus on the discrimination directly, model decision function (or posterior probabilities)

How-to:

- Class conditional densities
- Distances
- Models about decision function
- Error minimization

Statistical decision theory: basics

- No information (no measurements). Assign object x to ω_j based on **prior probabilities:**

$$p(\omega_j) > p(\omega_k) \quad for\ \ all\ \ k \neq j, \quad k = 1,..,K$$

- Given information, i.e. a vector representation **x** of the object x, assign **x** to ω_j based on the maximum **a posterior probability (MAP):**

$$p(\omega_j \mid \mathbf{x}) > p(\omega_k \mid \mathbf{x}) \quad for\ \ all\ \ k \neq j, \quad k = 1,..,K$$

- **Bayes theorem:**

posterior = likelihood * prior / evidence

$$p(\omega_j \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_j)\, p(\omega_j)}{p(\mathbf{x})}$$

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x} \mid \omega_k)\, p(\omega_k)$$

Bayes decision rule

Assume two classes, A and B:

$$p(A \mid \mathbf{x}) \geq p(B \mid \mathbf{x}) \rightarrow \mathbf{x} \in A, \ \text{otherwise, } \mathbf{x} \in B$$

Bayes:
$$\frac{p(\mathbf{x} \mid \mathrm{A})\, p(\mathrm{A})}{p(\mathbf{x})} \geq \frac{p(\mathbf{x} \mid \mathrm{B})\, p(\mathrm{B})}{p(\mathbf{x})} \rightarrow \mathbf{x} \in A, \ \text{otherwise, } \mathbf{x} \in B$$

$$p(\mathbf{x} \mid \mathrm{A})\, p(\mathrm{A}) \geq p(\mathbf{x} \mid \mathrm{B})\, p(\mathrm{B}) \rightarrow \mathbf{x} \in A, \ \text{otherwise, } \mathbf{x} \in B$$

$$S(\mathbf{x}) = p(A)\, p(\mathbf{x} \mid \mathrm{A}) - p(B)\, p(\mathbf{x} \mid \mathrm{B})$$

2-class problem:

$$\boxed{S(x) = p(\mathbf{x} \mid \mathrm{A})\, p(\mathrm{A}) - p(\mathbf{x} \mid \mathrm{B})\, p(\mathrm{B}) >= 0 \rightarrow \mathbf{x} \in A, \ \text{otherwise, } \mathbf{x} \in B}$$

K-class problems:

$$\boxed{\text{class}(\mathbf{x}) = \arg\max_{\omega_i}\, p(\mathbf{x} \mid \omega_i)\, p(\omega_i)}$$

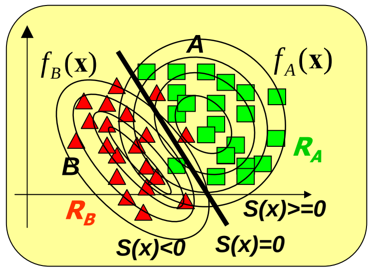
MAP: maximum a posterior

Classification error

Classification error ε is the probability that an arbitrary **x** is erroneously classified by a decision rule S(**x**):

If $S(\mathbf{x}) \geq 0$, then $\mathbf{x} \rightarrow \mathrm{A}$

If $S(\mathbf{x}) < 0$, then $\mathbf{x} \rightarrow \mathrm{B}$



$$\varepsilon = P(S(\mathbf{x}) < 0, \mathbf{x} \in A) + P(S(\mathbf{x}) \geq 0, \mathbf{x} \in B)$$

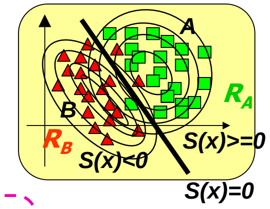
$$\varepsilon = P(S(\mathbf{x}) < 0 \mid \mathbf{x} \in A)\, p(A) + P(S(\mathbf{x}) \geq 0 \mid \mathbf{x} \in B)\, p(B)$$

$$\varepsilon = \underbrace{\int_{S(\mathbf{x})<0} p(A)\, f_A(\mathbf{x})\, d\mathbf{x}}_{\varepsilon_A} + \underbrace{\int_{S(\mathbf{x})\geq 0} p(B)\, f_B(\mathbf{x})\, d\mathbf{x}}_{\varepsilon_B} \qquad p(A) + p(B) = 1$$

$f_A(\mathbf{x})$ and $f_B(\mathbf{x})$ are the probability density functions of A and B.

The optimal rule is the Bayes decision rule

Determine the optimal S(**x**) such that



$$\varepsilon = \underbrace{\int_{S(\mathbf{x})<0} p(A) f_A(\mathbf{x}) d\mathbf{x}}_{\varepsilon_A} + \underbrace{\int_{S(\mathbf{x})\geq 0} p(B) f_B(\mathbf{x}) d\mathbf{x}}_{\varepsilon_B}$$ is **minimum**.

$$\varepsilon = \int_{R_B: S(\mathbf{x})<0} p(A) f_A(\mathbf{x}) d\mathbf{x} + \underbrace{\int_{R_A: S(\mathbf{x})\geq 0} p(B) f_B(\mathbf{x}) d\mathbf{x} + \int_{R_B: S(\mathbf{x})<0} p(B) f_B(\mathbf{x}) d\mathbf{x}}_{P(B)} - \int_{R_B: S(\mathbf{x})<0} p(B) f_B(\mathbf{x}) d\mathbf{x}$$

$$\varepsilon = p(B) + \int_{R_B: S(\mathbf{x})<0} [p(A) f_A(\mathbf{x}) - p(B) f_B(\mathbf{x})] d\mathbf{x}$$

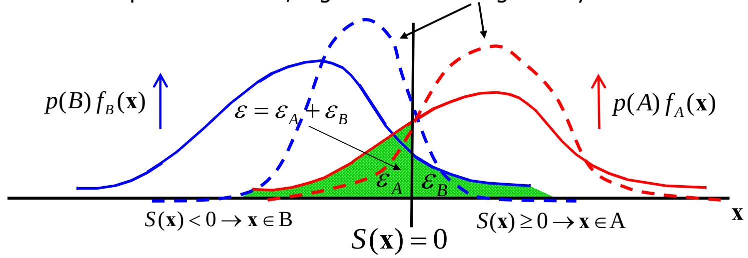
This is minimum if $[p(A) f_A(\mathbf{x}) - p(B) f_B(\mathbf{x})] < 0$ over R_B: S(x) <0.

So, the **optimal rule** is the **Bayes decision rule.**

$$S^*(x) = p(A) f_A(\mathbf{x}) - p(B) f_B(\mathbf{x})$$

Classification error

Sub-optimal classifier, e.g. based on wrong density estimates

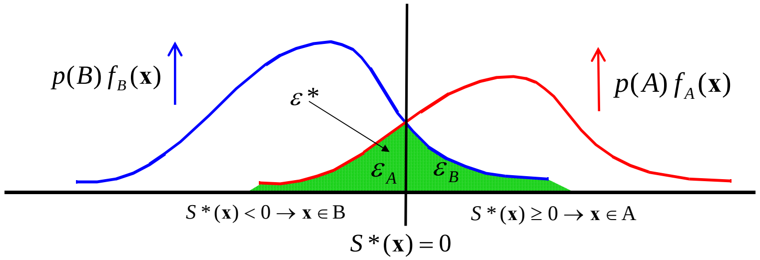


$$\varepsilon = \underbrace{\int_{S(\mathbf{x})<0} p(A) f_A(\mathbf{x}) d\mathbf{x}}_{\varepsilon_A} + \underbrace{\int_{S(\mathbf{x})\geq 0} p(B) f_B(\mathbf{x}) d\mathbf{x}}_{\varepsilon_B}$$

$p(A), p(B), f_A(\mathbf{x}), f_B(\mathbf{x})$ estimated by parametric or non-parametric approaches

S(**x**)=0 discriminant function, e.g. piece-wise linear

Optimal classification error = Bayes error



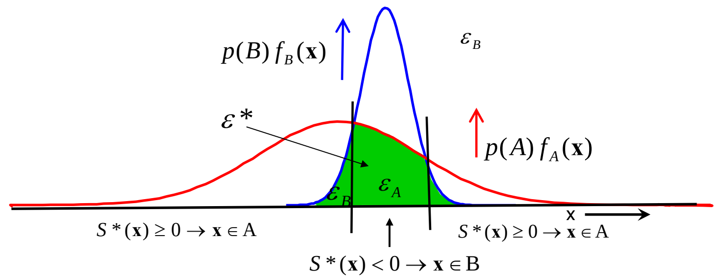
Classification error is minimal, ε*, if the decision function is optimal.
This is the **Bayes error**, the lowest achievable error!

Bayes decision rule: $S^*(\mathbf{x}) = p(A) f_A(\mathbf{x}) - p(B) f_B(\mathbf{x})$

Bayes error: $\varepsilon^* = \int \min\{p(A) f_A(\mathbf{x}),\ p(B) f_B(\mathbf{x})\} d\mathbf{x}$

Bayes error is only reachable if true distributions are known.

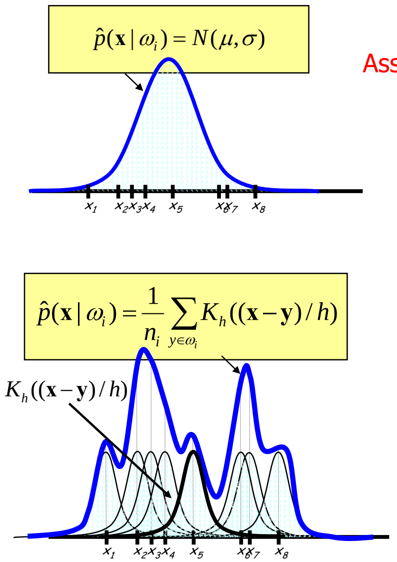
Bayes rule for different distributions



$$S^*(\mathbf{x}) = p(A) f_A(\mathbf{x}) - p(B) f_B(\mathbf{x})$$

$$\varepsilon^*(\mathbf{x}) = \int \min\{p(A) f_A(\mathbf{x}),\ p(B) f_B(\mathbf{x})\} d\mathbf{x}$$

Bayes decision making (how-to)



$$\text{class}(\mathbf{x}) = \arg\max_{\omega_i} p(\mathbf{x}|\omega_i)\,p(\omega_i)$$

Assumes knowledge about class density functions.

$$f_i(\mathbf{x}) = \hat{p}(\mathbf{x}|\omega_i) \qquad \hat{p}(\omega_i) = \frac{|\omega_i|}{\text{Total}}$$

Two approaches to estimate pdf's:

1) Parametric

- Assume a particular model for each pdf (restrict the shape)
- Estimate parameters from the training set
- E.g. Gaussian, Mixtures of Gaussians

2) **Non-parametric**

- No assumption about the underlying pdf (restrict the smoothness of the estimate)
- Estimate locally and combine globally
- E.g. Histogram, k-NN, Parzen

Bayes rule: summary

- **Bayes decision rule** is optimal when both class priors and pdfs are known.
- Usually, we have to approximate the priors and pdfs from the data. This leads to estimation errors. Only for very large training sets we may approach the Bayes error.
- In other cases additional costs or risk are involved. E.g:
  - it is very risky to classify an ill patient as healthy
  - it is less risky to classify a healthy patient as ill (extra tests)

  In this situation we have to adapt the formulation to the minimum cost classification.

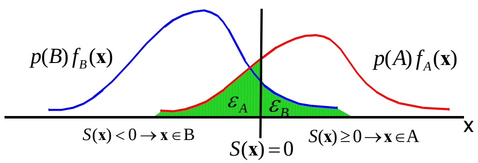
Minimum cost classification

- Costs related to erroneous classification:

	$\mathbf{x}\in$ A	$\mathbf{x}\in$ B
R_A: S(**x**) ≥ 0	Correct	$\mathbf{C_B}$
R_B: S(**x**) < 0	$\mathbf{C_A}$	Correct

$C_B = \text{cost}\,\{S(\mathbf{x}) \geq 0,\ \mathbf{x}\in\text{B}\}$

$C_A = \text{cost}\{S(\mathbf{x}) < 0,\ \mathbf{x}\in\text{A}\}$



- Total expected cost:

$$E[C] = C_A P(S(\mathbf{x})<0,\ \mathbf{x}\in\text{A}) + C_B P(S(\mathbf{x})\geq 0,\ \mathbf{x}\in\text{B})$$

- This is minimized by:

$$\boxed{S(\mathbf{x}) = C_A\,p(A)\,f_A(\mathbf{x}) - C_B\,p(B)\,f_B(\mathbf{x})}$$

Min-Max classification

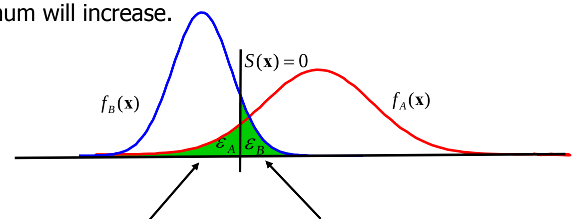
- If p(A), p(B) are unknown, find S(**x**) that **minimizes** the **maximum** possible error.

$$\min_{S(\mathbf{x})}\ \max_{p(A),p(B)} \int_{S(\mathbf{x})<0} p(A)\,f_A(\mathbf{x})d\mathbf{x} + \int_{S(\mathbf{x})\geq 0} p(B)\,f_B(\mathbf{x})d\mathbf{x}$$

- p(A) + p(B) =1 → maximum reached for p(A)=0, p(B)=1, or p(A)=1, p(B)=0.

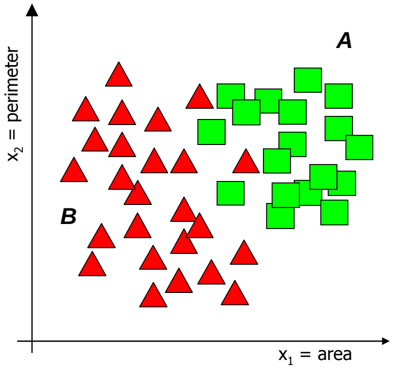
$$\min_{S(\mathbf{x})}\ \max\left\{\int_{S(\mathbf{x})<0} f_A(\mathbf{x})d\mathbf{x},\ \int_{S(\mathbf{x})\geq 0} p(B)\,f_B(\mathbf{x})d\mathbf{x}\right\}$$

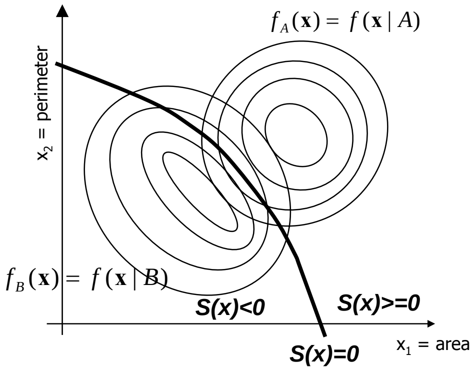
- This is minimum if S(**x**) is such that the two terms are equal. Other S(**x**) will increase one of them → the maximum will increase.



$$S(\mathbf{x}):\ \int_{S(\mathbf{x})<0} f_A(\mathbf{x})d\mathbf{x} = \int_{S(\mathbf{x})\geq 0} p(B)\,f_B(\mathbf{x})d\mathbf{x}$$

Discriminant Analysis





Probability density estimates of the classes

Quadratic discriminant=Bayes rule for Normal Distributions [G]

Bayes rule $S(\mathbf{x}) = p(A)\,p(\mathbf{x} \mid \mathrm{A}) - p(B)\,p(\mathbf{x} \mid \mathrm{B}) = 0$

logs don't matter

$$p(A)\,p(\mathbf{x} \mid \mathrm{A}) = p(B)\,p(\mathbf{x} \mid \mathrm{B})$$

$$\log[\,p(A)\,p(\mathbf{x} \mid \mathrm{A})] = \log[\,p(B)\,p(\mathbf{x} \mid \mathrm{B})]$$

$$R(\mathbf{x}) = \log(\,p(A)\,p(\mathbf{x} \mid \mathrm{A})) - \log(\,p(B)\,p(\mathbf{x} \mid \mathrm{B}))$$

R(x) and S(x) have the same signs

$$R(\mathbf{x}) = \log(\,p(\mathbf{x} \mid \mathrm{A})) - \log(\,p(\mathbf{x} \mid \mathrm{B}) + \log[\,p(\mathrm{A})/\,p(\mathrm{B})]$$

Normal distribution

$$p(\mathbf{x} \mid \mathrm{A}) = \frac{1}{\sqrt{2\pi^k \det(\boldsymbol{\Sigma}_\mathrm{A})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_\mathrm{A})^\mathrm{T} \Sigma_\mathrm{A}^{-1} (\mathbf{x} - \boldsymbol{\mu}_\mathrm{A})\right)$$

$$\log(\,p(\mathbf{x} \mid \mathrm{A})) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_\mathrm{A})^\mathrm{T} \Sigma_\mathrm{A}^{-1} (\mathbf{x} - \boldsymbol{\mu}_\mathrm{A}) - \log(\sqrt{2\pi^k \det(\boldsymbol{\Sigma}_\mathrm{A})})$$

Quadratic expression

Substitute

$$\boxed{R(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{A})^\mathrm{T} \hat{\Sigma}_\mathrm{A}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{A}) + \frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{B})^\mathrm{T} \hat{\Sigma}_\mathrm{B}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{B}) + \mathrm{const}}$$

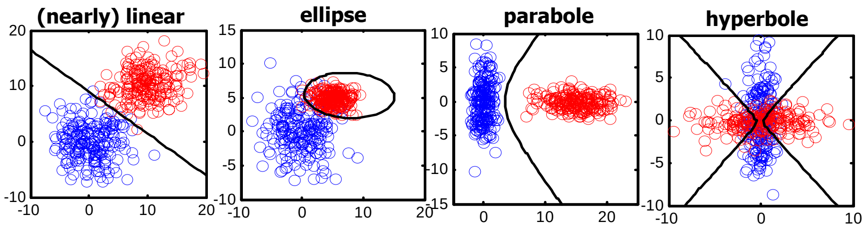
$$\mathrm{const} = \log\{\,p(A)/(\,p(B)\} + \frac{1}{2}\log\{\det(\hat{\Sigma}_\mathrm{B})/\det(\hat{\Sigma}_\mathrm{A})\}$$

qdc udc

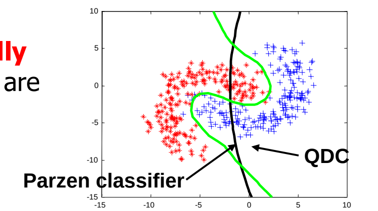
Quadratic discriminant functions

$$\boxed{R(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{A})^\mathrm{T} \hat{\Sigma}_\mathrm{A}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{A}) + \frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{B})^\mathrm{T} \hat{\Sigma}_\mathrm{B}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{B}) + \mathrm{const}}$$

$$\mathrm{const} = \log\{\,p(A)/(\,p(B)\} + \frac{1}{2}\log\{\det(\hat{\Sigma}_\mathrm{B})/\det(\hat{\Sigma}_\mathrm{A})\}$$



QDC assumes that classes are normally distributed. Wrong decision boundaries are estimated if this does not hold.



Bayes rule for Normal Distributions with Equal Covariances

QDC $\boxed{R(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{A})^\mathrm{T} \hat{\Sigma}_\mathrm{A}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{A}) + \frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{B})^\mathrm{T} \hat{\Sigma}_\mathrm{B}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_\mathrm{B}) + \mathrm{const}}$

Assume Σ_A and Σ_B are equal: $\Sigma = \Sigma_A = \Sigma_B$. Quadratic term disappears.

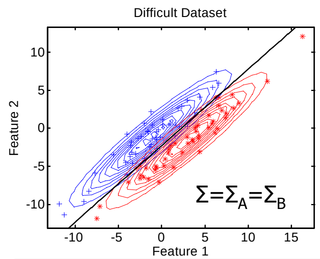
Linear expression

LDC $\boxed{R(\mathbf{x}) = (\hat{\boldsymbol{\mu}}_\mathrm{A} - \hat{\boldsymbol{\mu}}_\mathrm{B})^\mathrm{T} \hat{\Sigma}^{-1} \mathbf{x} + \mathrm{const}}$

$$const = -\frac{1}{2}\hat{\boldsymbol{\mu}}_\mathrm{A}{}^\mathrm{T} \hat{\Sigma}^{-1} \hat{\boldsymbol{\mu}}_\mathrm{A} + \frac{1}{2}\hat{\boldsymbol{\mu}}_\mathrm{B}{}^\mathrm{T} \hat{\Sigma}^{-1} \hat{\boldsymbol{\mu}}_\mathrm{B} + \log[\mathrm{p(A)/p(B)}]$$

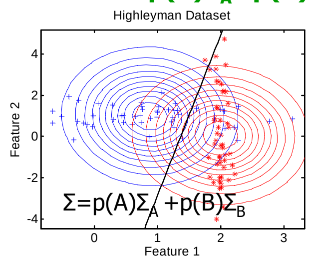
ldc

Unequal covariance matrices → use linear approximation $\boldsymbol{\Sigma = p(A)\Sigma_A + p(B)\Sigma_B}$

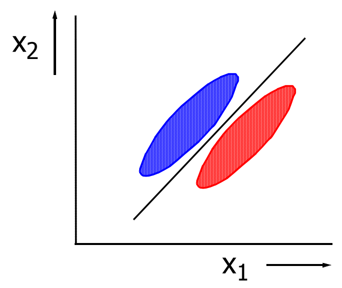


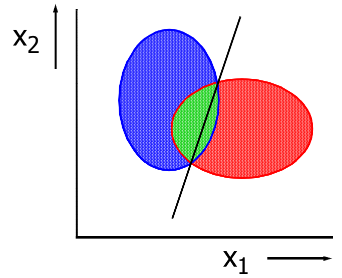
gendath

gendatd



Linear discriminant function (summary) [G]





Normal distributions with equal covariance matrices Σ are optimally separated by a linear classifier

$$S(\mathbf{x}) = (\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)^T \Sigma^{-1} \mathbf{x} + \text{const}$$

The optimal classifier for normal distributions with unequal covariance matrices Σ_A and Σ_B can be approximated by:

$$S(\mathbf{x}) = (\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)^T (p(A)\Sigma_A + p(B)\Sigma_B)^{-1} \mathbf{x} + \text{const}$$

ldc

Fisher linear discriminant (I)

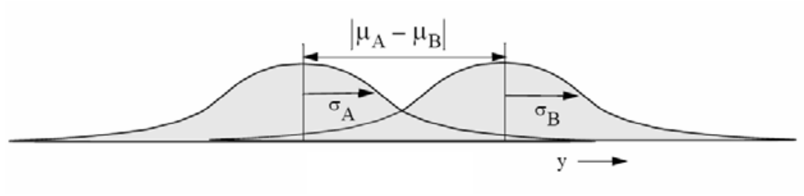
Assume a two-class problem. We look for a linear discriminant:

$$S(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

such that the **separability** between the classes **is maximized** along **w.**

Fisher criterion:

$$J_F = \frac{\sigma^2_{\text{Between-class}}}{\sigma^2_{\text{Within-class}}} = \frac{|\mu_A - \mu_B|^2}{\sigma_A^2 + \sigma_B^2}$$



Fisher linear discriminant (II)

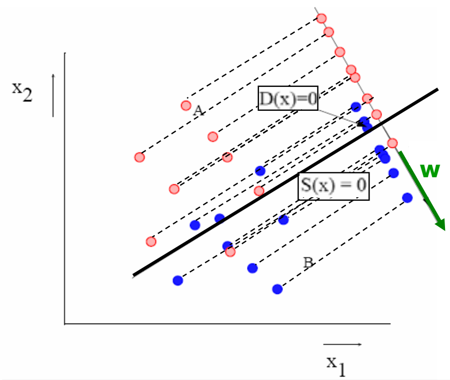
fisherc

Fisher criterion along the direction w:

$$J_F = \frac{|\mathbf{w}^T\boldsymbol{\mu}_A - \mathbf{w}^T\boldsymbol{\mu}_B|^2}{\mathbf{w}^T p_A \Sigma_A \mathbf{w} + \mathbf{w}^T p_B \Sigma_B \mathbf{w}} = \frac{\mathbf{w}^T(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)^T(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)\mathbf{w}}{\mathbf{w}^T(p_A\Sigma_A + p_B\Sigma_B)\mathbf{w}} = \frac{\mathbf{w}^T\Sigma_B\mathbf{w}}{\mathbf{w}^T\Sigma_W\mathbf{w}}$$

Σ_B is the between-class covariance matrix.

Σ_W is the within-class covariance matrix.



Solution for $\Sigma_W = \Sigma$:

$$\mathbf{w} = \hat{\Sigma}_W^{-1}(\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)$$

$$\boxed{S(\mathbf{x}) = (\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)^T \hat{\Sigma}^{-1} \mathbf{x} + \text{const}}$$

Same as LDC up to a constant.

No assumption is made about normality of the data.

Nearest mean classifier (NMC) [G]

Assume $\Sigma = \Sigma_A = \Sigma_B = I$. Linear discriminant becomes the nearest mean classifier.

NMC

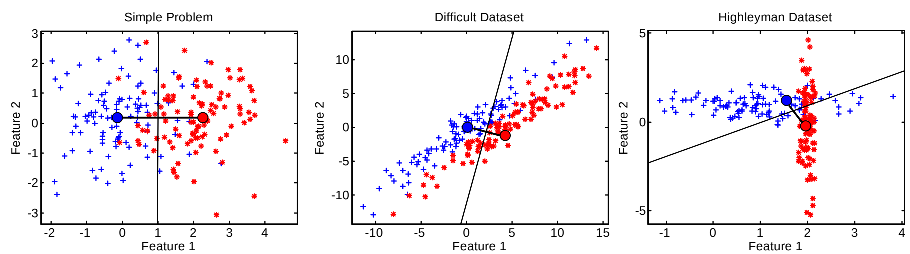
$$\boxed{R(\mathbf{x}) = (\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)^T \mathbf{x} - (\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)^T (\hat{\boldsymbol{\mu}}_A + \hat{\boldsymbol{\mu}}_B)/2}$$

nmc

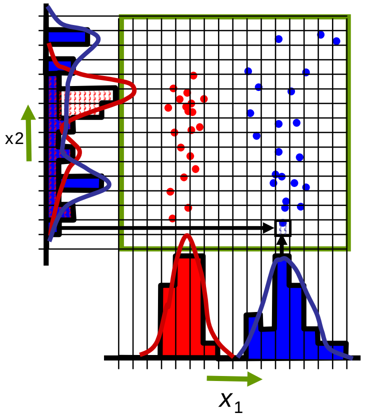
LDC, FisherC

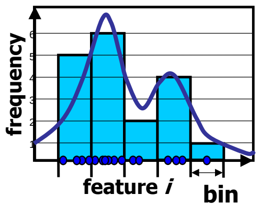
$$\boxed{R(\mathbf{x}) = (\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)^T \hat{\Sigma}^{-1} \mathbf{x} + \text{const}}$$

ldc fisherc



Naïve-Bayes classifier [G]





Assume class-independent features.

$$p(\mathbf{x}|A) = \prod_{i=1}^{k} p(x_i \mid A)$$

$$p(\mathbf{x}|B) = \prod_{i=1}^{k} p(x_i \mid B)$$

Estimate class probability density functions per feature: 1D histograms, 1D normal distributions, 1D Parzen estimates, etc. Multiply estimates.

Use Bayes decision rule: $S(\mathbf{x}) = p(A)\, f_A(\mathbf{x}) - p(B)\, f_B(\mathbf{x})$

$$\text{class}(\mathbf{x}) = \arg\max_{\omega_k} p(\mathbf{x} \mid \omega_k)\, p(\omega_k)$$

naivebc

Logistic model – logistic classifier [D]

- It holds for the Bayes discriminant:

$$p(A)\,p(\mathbf{x}\mid \mathrm{A}) = p(B)\,p(\mathbf{x}\mid \mathrm{B}) \rightarrow \log\{p(A)\,p(\mathbf{x}\mid \mathrm{A})\} = \log\{p(B)\,p(\mathbf{x}\mid \mathrm{B})\}$$

$$\rightarrow \log\left(\frac{p(A)\,p(\mathbf{x}\mid \mathrm{A})}{p(B)\,p(\mathbf{x}\mid \mathrm{B})}\right) = 0$$

- For linear discriminants, we have:

$$\log\left(\frac{p(A\mid \mathbf{x})}{p(B\mid \mathbf{x})}\right) = \log\left(\frac{p(A)\,p(\mathbf{x}\mid \mathrm{A})}{p(B)\,p(\mathbf{x}\mid \mathrm{B})}\right) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

Given that $p(B\mid \mathbf{x}) = 1 - p(A\mid \mathbf{x})$

$$p(A\mid \mathbf{x}) = \frac{p(A)\,p(\mathbf{x}\mid \mathrm{A})}{p(A)\,p(\mathbf{x}\mid \mathrm{A}) + p(B)\,p(\mathbf{x}\mid \mathrm{B})} = \frac{e^{\mathbf{w}^{\mathrm{T}}\mathbf{x}+w_0}}{1+e^{\mathbf{w}^{\mathrm{T}}\mathbf{x}+w_0}} = \frac{1}{1+e^{-\mathbf{w}^{\mathrm{T}}\mathbf{x}-w_0}}$$

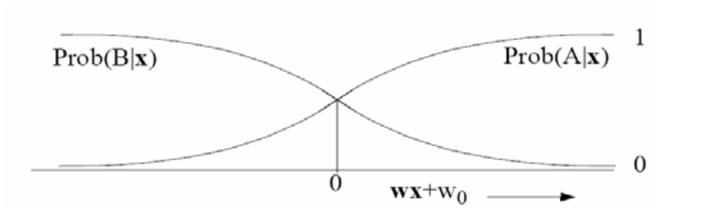
$$p(B\mid \mathbf{x}) = \frac{p(B)\,p(\mathbf{x}\mid \mathrm{B})}{p(A)\,p(\mathbf{x}\mid \mathrm{A}) + p(B)\,p(\mathbf{x}\mid \mathrm{B})} = \frac{1}{1+e^{\mathbf{w}^{\mathrm{T}}\mathbf{x}+w_0}}$$

loglc

Logistic function

- It appears that $\log\left(\frac{p(A)\,p(\mathbf{x}\mid \mathrm{A})}{p(B)\,p(\mathbf{x}\mid \mathrm{B})}\right)$ is linear for many distributions.
- E.g. normal, binary, multimodal and mixtures of them.

$$f(\mathbf{x}) = p(A\mid \mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^{\mathrm{T}}\mathbf{x}-w_0}}$$ is called the logistic function.



See: Anderson, Logistic Discrimination, in : Handbook of Statistics, vol. 2, Krishnaiah and Kanal (eds.), North Holland, 1982, pp. 169 - 191

The Logistic Model, ML Estimation

- Observations X={$\mathbf{x}_1$,..,$\mathbf{x}_n$} depend on the unknown parameter θ.
- **Assumption:** data samples are **independent, identically distributed (iid):** f($\mathbf{x}_1$,..,$\mathbf{x}_n$| θ) = ∏ f($\mathbf{x}_i$| θ).
- Likelihood is a function of θ, samples $\mathbf{x}_i$ are fixed. L(θ|X)=f($\mathbf{x}_1$,..,$\mathbf{x}_n$|θ)=∏ f($\mathbf{x}_i$|θ).
- **Maximum Likelihood**: θ_{ML}=argmax$_\theta$ L(θ|X) =argmax$_\theta$ log L(θ|X).

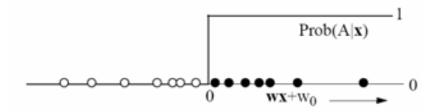
In the logistic model, we maximize the conditional log-likelihood:

$$\log L(\mathbf{w}) = \log\{\prod_{\mathbf{x}_i \in A} p(A\mid \mathbf{x}_i;\mathbf{w}) \prod_{\mathbf{x}_i \in B} p(B\mid \mathbf{x}_i;\mathbf{w})\}$$

by using a gradient-descent method (steepest ascent or Newton) :

$$0 = \frac{\partial \log L(\mathbf{w})}{\partial w_j} = \sum_{\mathbf{x}\in A} x_j\, p(A\mid \mathbf{x};\mathbf{w}) - \sum_{\mathbf{x}\in B} x_j\, p(B\mid \mathbf{x};\mathbf{w})$$

For separable classes, maximum is at ∞, as p(A|**x**)=1 for x in A, and p(B|**x**)=1 for x in B.

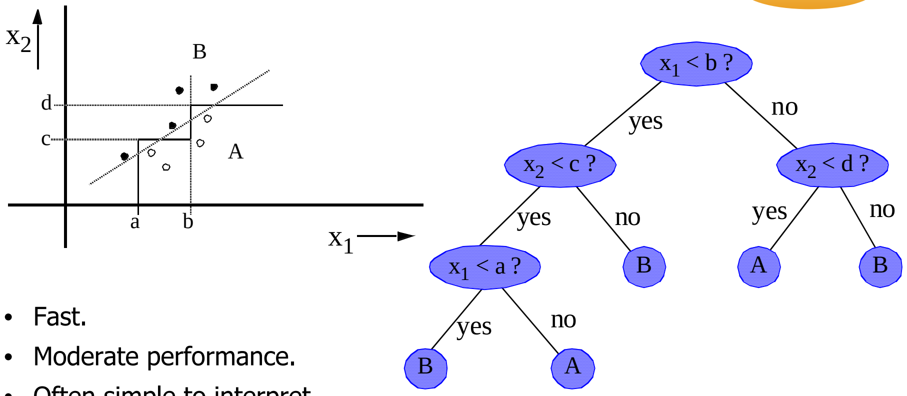


loglc

Decision trees [D]

Implementation of a piece-wise linear classifier:

treec



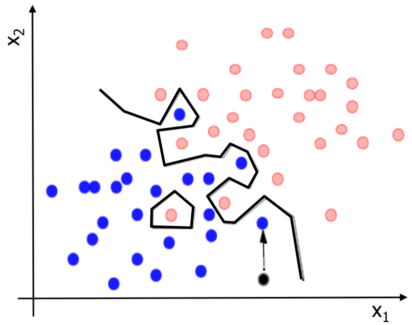
- Fast.
- Moderate performance.
- Often simple to interpret.
- Can handle numerical and categorical variables.

C4.5*-decision tree. An algorithm used to generate a decision tree developed by Ross Quinlan.*
See: J.R.Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

Nearest neighbor rule (1-NN rule) [D]

Assign a new object to the class of the nearest neighbor in the training set.

knnc

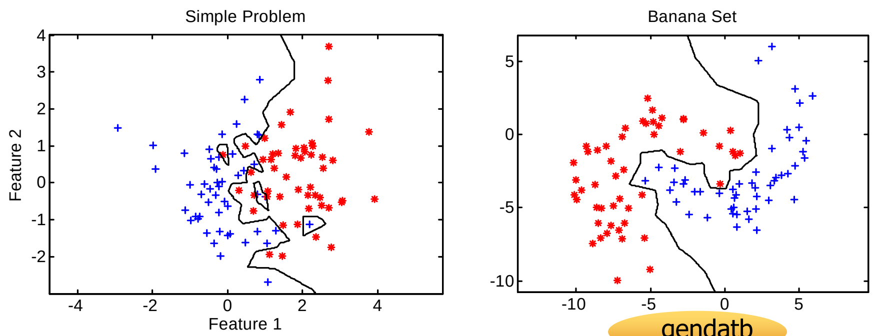


1-NN rule:

- Often relies on the Euclidean distance. Other distance measures can be used.
- Insensitive to prior probabilities!
- Scaling dependent. Features should be scaled properly.

There are **no** errors on the training set. The classifier is overtrained.

1-NN rule: examples



gendatb

Advantages:

- Simple.
- Works well for almost separable classes.
- Useful to shape non-linear decision functions.

Disadvantages:

- No training time. Long execution time.
- All data should be stored.

1-NN classification error

Asymptotically (for very large training sets):

$$\varepsilon^* \leq \varepsilon_{\substack{1-NN \\ n\to\infty}} \leq 2\varepsilon^*(1-\varepsilon^*) \leq 2\varepsilon^*$$

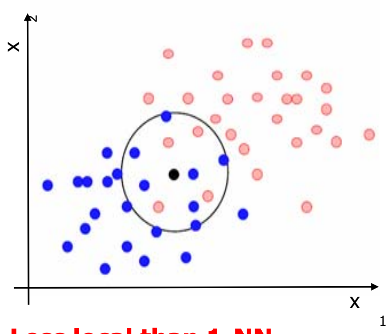
The nearest neighbor rule will not perform worse than twice the best possible classifier.

1-NN is often a very good classifier!!!!

k-nearest neighbor rule (k-NN) [D]

Assign an object to the class that is most frequently represented among k nearest neighbors in the training set of n objects.

knnc



Less local than 1-NN.
More smooth.
Very global when k→n.

- k-NN class density estimates

$$\hat{p}(\mathbf{x}\,|\,\omega_j)=\frac{k_j}{n_j\mathrm{Vol}(\mathbf{x})}$$

- Priors

$$\hat{p}(\omega_j)=\frac{n_j}{n}$$

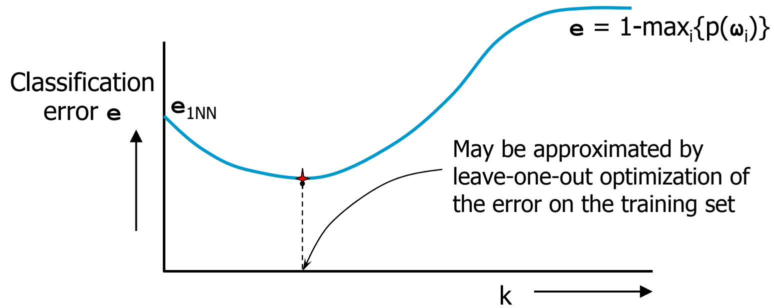
- Decision rule

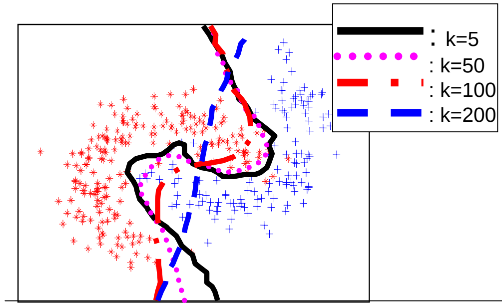
$$\frac{k_k}{n_k\mathrm{Vol}(\mathbf{x})}\frac{n_k}{n} > \underbrace{\frac{k_j}{n_j\mathrm{Vol}(\mathbf{x})}\frac{n_j}{n}}_{\hat{p}(\omega_j)\hat{p}(\mathbf{x}|\omega_j)} \quad \forall j\neq k$$

- Simplifies to **majority vote:**

$$k_k > k_j \quad \forall j\neq k$$

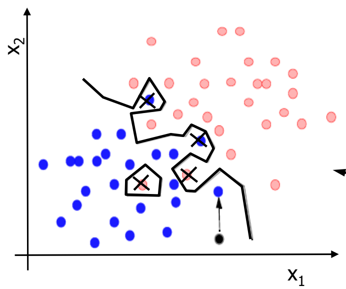
k-NN decision boundaries: optimal k in the k-NN rule





Rule of thumb:
choose k= sqrt(n)

Nearest prototype rule: editing and condensing

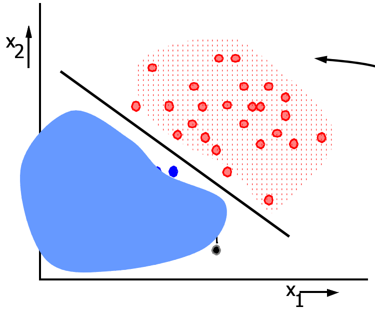


k-NN rule distances to all training objects have to be computed.

Editing-and-condensing reduces the complexity while aiming at similar classification accuracy.

edicon

Editting: remove objects that are misclassified by the k-NN rule.



Condensing: select a subset of prototypes such that the 1-NN rule performs similarly as on the complete training set.

Parzen classifier [G]

- Bayes decision rule

$$S(\mathbf{x})=p(A)\,f_A(\mathbf{x})-p(B)\,f_B(\mathbf{x})$$

$$\mathrm{class}(\mathbf{x})=\arg\max_{\omega_k}\,p(\mathbf{x}\,|\,\omega_k)\,p(\omega_k)$$

- Substitute Parzen density estimates

$$\hat{p}(\mathbf{x}\,|\,\omega_k)=\frac{1}{n_k}\sum_{\mathbf{x}_i\in\omega_k}K_h\!\left(\frac{\mathbf{x}_i-\mathbf{x}}{h}\right)$$

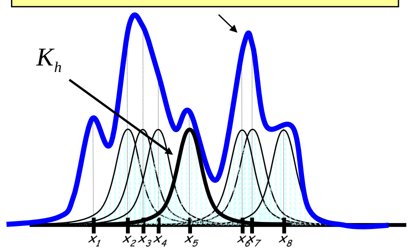
- Parzenc: optimize *h* for classification

parzenc

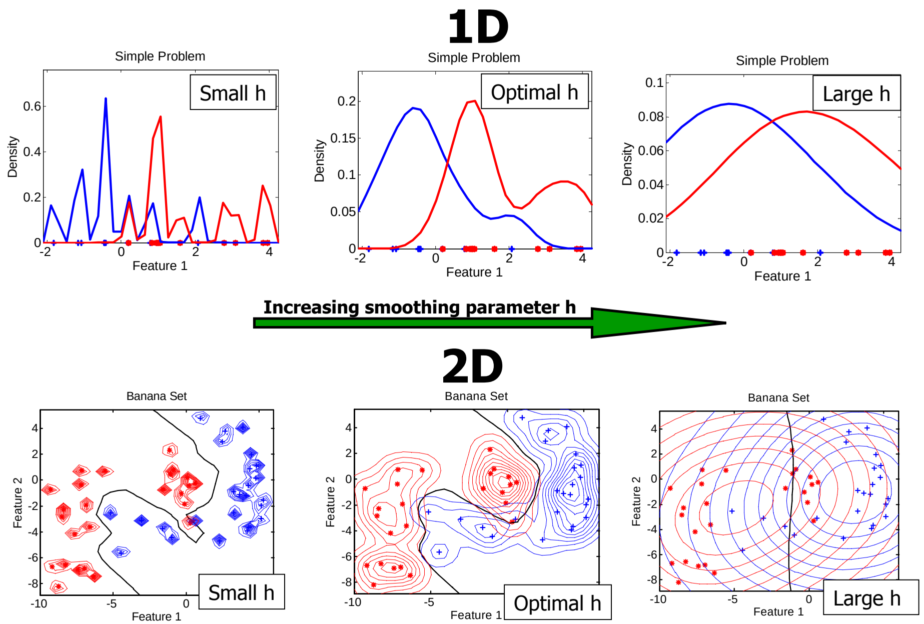
- Parzendc: optimize *h* for density estimation per class

parzendc

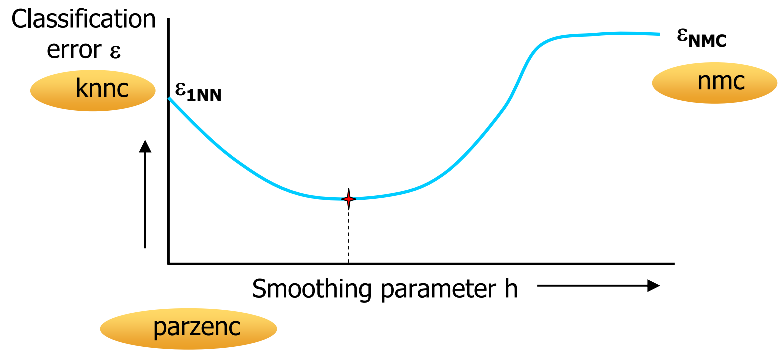
$$\boxed{\hat{p}(\mathbf{x}\,|\,\omega_k)=\frac{1}{n_k}\sum_{\mathbf{x}_i\in\omega_k}K_h\!\left(\frac{\mathbf{x}_i-\mathbf{x}}{h}\right)}$$



Parzen: density estimates vs the smoothing parameter h



Parzen classifier performance



Parzen classifier:

- Small smoothing parameter: 1-NN performance, $\varepsilon \rightarrow \varepsilon_{1NN}$
- Large smoothing parameter: Nearest mean performance, $\varepsilon \rightarrow \varepsilon_{NMC}$

Perceptron [D]

perlc

Linear classifier: $S(\mathbf{x}') = \mathbf{w}^T\mathbf{x}' \qquad \mathbf{x}_i' = [y_i\mathbf{x}_i \quad y_i]^T$

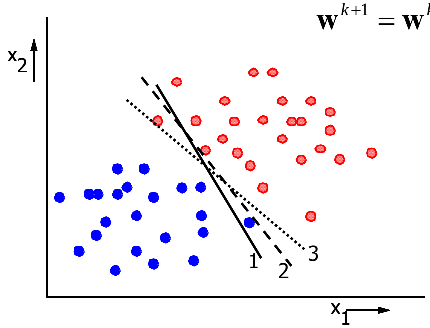
$$y_i = 1, \text{if } \mathbf{x}_i \in A; \quad y_i = -1, \text{if } \mathbf{x}_i \in B$$

Linear separability: $\mathbf{w}^T\mathbf{x}'_j > 0 \quad \forall \mathbf{x}'_j$

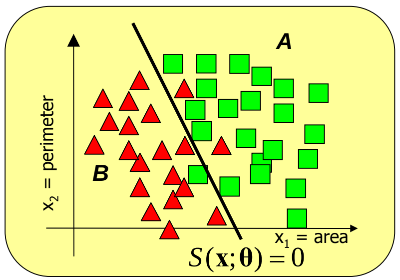
The weights are iteratively updated *only* for erroneously classified objects, i.e.

$$\mathbf{w}^T\mathbf{x}'_i < 0$$

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \alpha\,\Delta\mathbf{w}(\mathbf{w}^k, \mathbf{x}_i') = \mathbf{w}^k + \alpha\mathbf{x}_i'$$



Classifiers based on error optimization [D]



perlc

lmnc

bpxnc

If $S(\mathbf{x};\boldsymbol{\theta}) \geq 0$, then $\mathbf{x} \rightarrow$ A

If $S(\mathbf{x};\boldsymbol{\theta}) < 0$, then $\mathbf{x} \rightarrow$ B

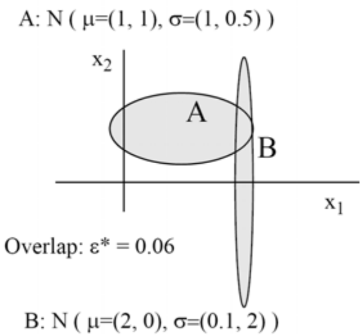
Change parameters **θ** of the decision function such that the classification error is minimized. Usually, gradient-based techniques are used to solve nonlinear equations.

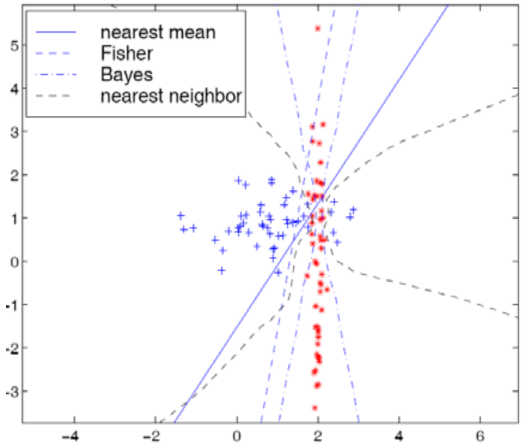
Error function: $J(\boldsymbol{\theta}) = \sum_{\mathbf{x} \text{ in Training set}} F(S(\mathbf{x};\boldsymbol{\theta}))$

E.g. error count, average error, sum of distances to the boundary

Example: Highleyman's classes

gendath

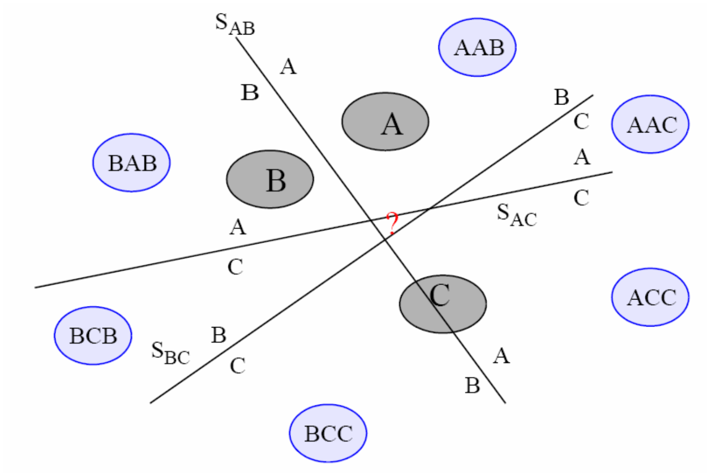




See: W.H. Highleyman, Linear Decision Functions with Applications to Pattern Recognition, Proc. IRE - 50, 1962, 1501

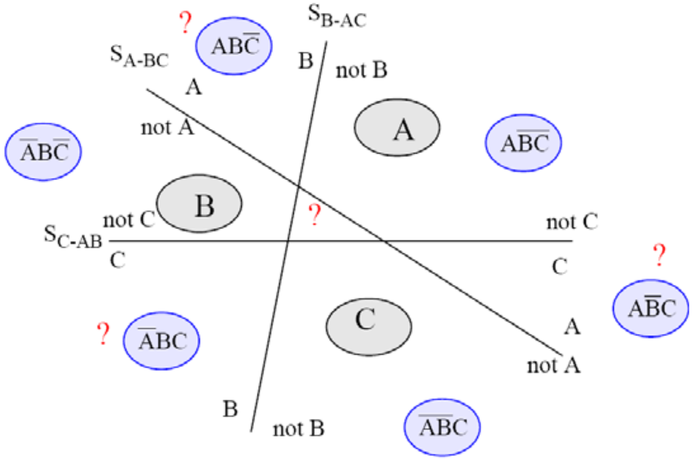
Multiple classifiers (I)

Undecidable region in case of multiple 2-class discriminants.



Multiple classifiers (II)

Undecidable regions in case of multiple one-vs-all-other discriminants.

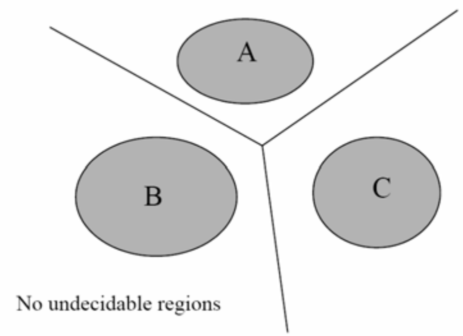


Multiple classifiers (III)

PRTools

Instead of discriminants, use class description functions: class probability density functions, Euclidean or Mahalanobis distances.

If D($\mathbf{x}$,ω_k) > D($\mathbf{x}$,ω_i)   for all i ≠ k   then x → ω_k

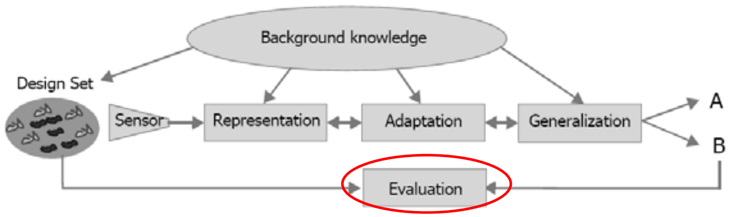


Summary on the statistical approach to classification

- Objects are vectors in a Euclidean space. Classes are groups of vectors.
- **Classification:** find a decision function that discriminates between classes. Additional assumptions or models are necessary because of finite data.
- **Bayes decision rule is the basis of probabilistic classification.**
- Two major approaches:

  Generative classifiers: estimate class conditional densities by parametric / non-parametric approaches. Derive posterior probabilities via Bayes theorem.

  Discriminative classifiers: estimate either posterior probabilities directly or determine a decision function.
- We know how to construct classifiers. **Evaluation is crucial to find the best one.**



Some classifiers in PRTools

Generative classifiers:
- NMC - nearest mean classifier
- NMSC - nearest mean scaled classifier
- FISHERC - Fisher linear discriminant
- LDC - linear discriminant
- QDC - quadratic discriminant
- UDC – quadratic discriminant with diagonal covariance matrices
- MOGC - mixture of Gaussians classification
- PARZENC - Parzen classifier
- NAIVEBC - naïve Bayes classifier

Discriminative classifiers:
- TREEC - binary decision tree classifier
- BPXNC - feed forward neural network classifier by backpropagation
- LMNC - feed forward neural network by Levenberg-Marquardt rule
- PERLC - linear perceptron
- RBNC - radial basis neural network classifier
- SUBSC - subspace classifier
- SVC - support vector machine
- KNNC - k-nearest neighbor rule