10th VIPS Advanced School on Computer Vision and Pattern Recognition

Dissimilarity-based Representation for Pattern Recognition

Robert P.W. Duin, Delft University of Technology

Pattern Recognition Lab
Delft University of Technology
The Netherlands

//rduin.nl

TUDelft

Participants

Mohamed Lamine Mekhalfi
Attaullah Buriro
Mohammad Bilal
John Kenneth Thiessen
Andrea Gasparetto
Andres Mendez
Cristina Segalin
Davide Conigliaro

Sami Abduljalil Abdulhak Naji
Pietro Lovato
Ricardo Henrique Gracini Guiraldelli
Anna Pesarin
Filippo Bistaffa
Valeria Garro
Denis Peruzzo

Robert (Bob) Duin
Manuele Bicego
Umberto Castellani

TUDelft

Program

1. Representation and Generalization
2. The Dissimilarity Space
3. Pseudo-Euclidean embedding
4. Applications

TUDelft

Daily Schedule

10:00 - 12:00 Lectures
http://www.37steps.com/disrep-course/

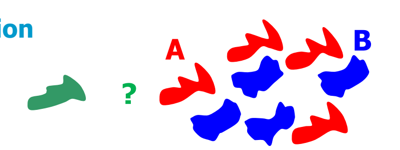
13:30 – 16:30 Lab course
http://www.37steps.com/disrep_exercises/

TUDelft

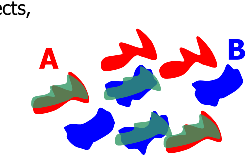
Pattern Recognition Problems

TUDelft

Question



How to represent real world objects,
(with a size and a shape)
given a set of examples
such that we can generalize?



TUDelft

Real world objects and events

Images
Spectra
Time signals
Gestures
→ shapes

How to build a representation?
Features ←→ Structure

Blob Recognition

BACK
BREAST
DRUMSTICK
THIGH-AND-BACK
WING

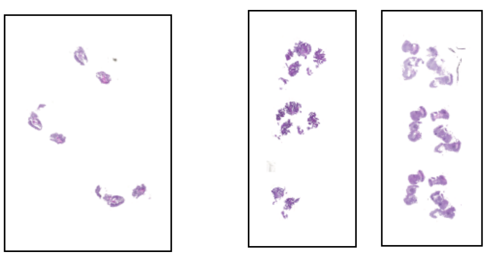
446 binary images, varying size, e.g.: 100 x 130

Andreu, G., Crespo, A., Valiente, J.M.: Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recogn. In: ICNN. (1997) 1341–1346.

Shape classification by weighted-edit distances (Bunke)

*Bunke, H., Buhler, U.: Applications of approximate string matching to 2D shape recognition. Pattern recognition **26** (1993) 1797–1812*

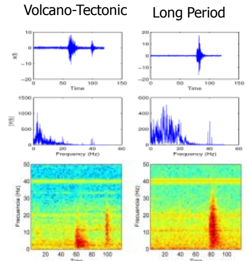
Colon Tissue Recognition



??? normal pathological

Volcano / Seismic Signal Classification

Volcano-Tectonic Long Period



150 000 events (1994 – 2008)
5 volcanos
40 stations
15 classes

J. Makario,
INGEOMINAS, Manizales, Colombia

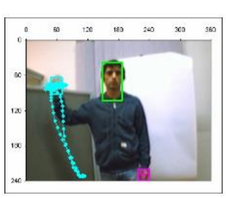
M. Orozco-Alzate,
Nat. Univ. Colombia, Manizales

R. Duin, TUDelft

M. Bicego, Univ. of Verona, Italy

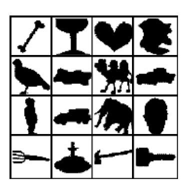
Cenatav, Havana, Cuba

Gesture Recognition

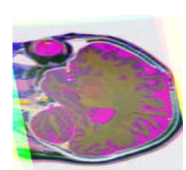


Is this gesture in the database?

Pattern Recognition Problems



To which class belongs an image



To which class (segment) belongs every pixel?

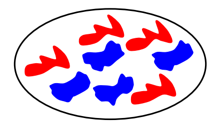
Where is an object of interest (detection); What is it (classification)?

Pattern Recognition: Shape Recognition

Pattern Recognition is very often Shape Recognition:

- Images: B/W, grey value, color, 2D, 3D, 4D
- Time Signals
- Spectra

Pattern Recognition: Shapes



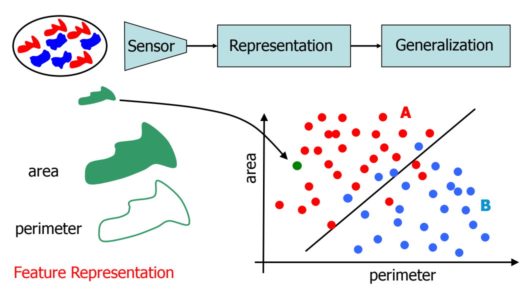
Examples of objects for different classes

Object of unknown class to be classified

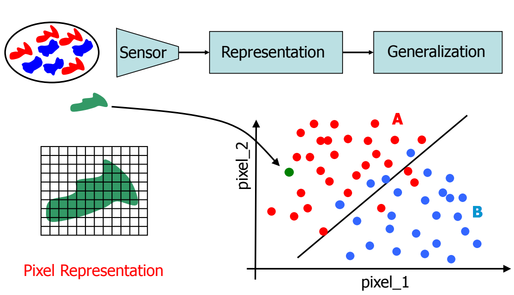
A ? B

Vector Representation

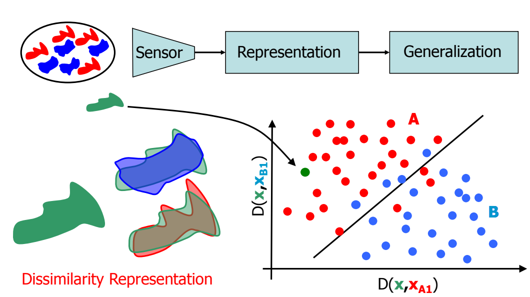
Pattern Recognition System



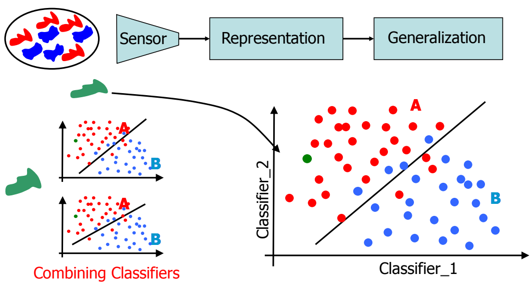
Pattern Recognition System



Pattern Recognition System



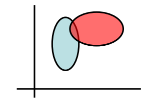
Pattern Recognition System



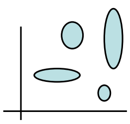
TUDelft

Good Representations

- Class specific
  Different classes should be represented in different positions in the representation space.



- Compact
  Every class should be represented in a small set of finite domains.

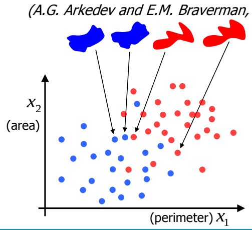


TUDelft

Compactness

Representations of real world similar objects are close.
There is no ground for any generalization (induction) on representations that do not obey this demand.

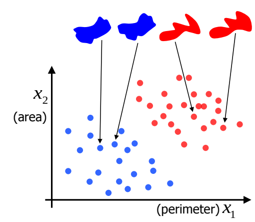
(A.G. Arkedev and E.M. Braverman, Computers and Pattern Recognition, 1966.)



The compactness hypothesis is not sufficient for perfect classification as dissimilar objects may be close.
→ class overlap
→ probabilities

TUDelft

True Representations



Similar objects are close
and
Dissimilar objects are distant.

→ no probabilities needed, domains are sufficient!

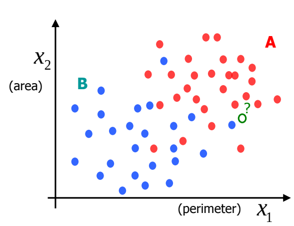
TUDelft

Distances and Densities

? to be classified as

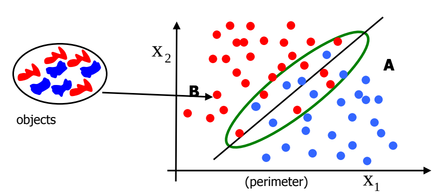
B – because it is most close to an object B

A – because the local density of A is larger.



TUDelft

Features Reduce



Due to reduction essentially different objects are represented identically.

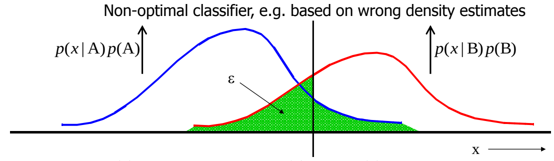
→ The feature space representation needs a statistical, probabilistic generalization

TUDelft

Classification

Classification error

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



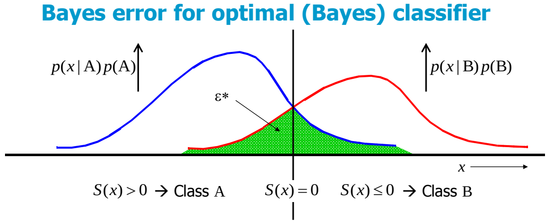
$S(x) > 0$ → Class A $S(x) = 0$ $S(x) \leq 0$ → Class B

$$\varepsilon = p(S(x) > 0, \mathrm{B}) + p(S(x) < 0, \mathrm{A})$$

$$\varepsilon = p(S(x) > 0 | \mathrm{B})p(\mathrm{B}) + p(S(x) < 0 | \mathrm{A})p(\mathrm{A})$$

$$\varepsilon = \int_{S(x)>0} p(x|\mathrm{B})p(\mathrm{B})dx + \int_{S(x)\leq 0} p(x|\mathrm{A})p(\mathrm{A})dx$$

Bayes error for optimal (Bayes) classifier



$S(x) > 0$ → Class A $S(x) = 0$ $S(x) \leq 0$ → Class B

Classification error is minimal, ε*, if the decision function is optimal:

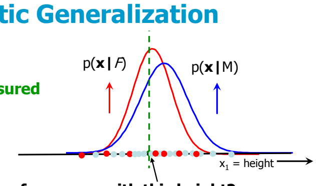
$$S(x)^* = p(x|\mathrm{A})p(\mathrm{A}) - p(x|\mathrm{B})p(\mathrm{B})$$

$$\varepsilon^* = \int \min\{p(x|\mathrm{A})p(\mathrm{A}), p(x|\mathrm{B})p(\mathrm{B})\}dx$$

Only possible if true distributions are known

Probabilistic Generalization

x = height measured



What is the gender of a person with this height?

Best guess is to choose the most 'probable' class (→ small error).

⇒ Good for overlapping classes.

⇒ Assumes the existence of a probabilistic class distribution and a representative set of examples.

Bayes decision rule, formal

	p(A|x)	>	p(B|x)	→ A else B
Bayes:	$\frac{p(x\|A)\ p(A)}{p(x)}$	>	$\frac{p(x\|B)\ p(B)}{p(x)}$	→ A else B
	p(x|A) p(A)	>	p(x|B) p(B)	→ A else B

2-class problems: S(x) = p(x|A) p(A) - p(x|B) p(B) > 0 → A else B

n-class problems: Class(x) = argmax_{ω}(p(x|ω) p(ω))

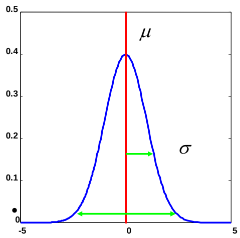
Density estimation

- The density is defined on the whole feature space.
- Around object x, the density is defined as:

$$p(x) = \frac{dP(x)}{dx} = \left(\frac{\text{fraction of objects}}{\text{volume}}\right)$$

- Given n measured objects, e.g. person's height (m) how can we estimate $p(x)$?

The Gaussian distribution (3)

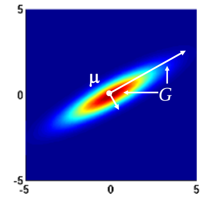


- Normal distribution = Gaussian distribution
- Standard normal distribution: $\mu = 0,\ \sigma^2 = 1$
- 95% of data between $[\,\mu - 2\sigma, \mu + 2\sigma\,]$ (in 1D!)

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2} \right)$$

TUDelft

Multivariate Gaussians



$$G = \begin{bmatrix} 3 & 1½ \\ 1½ & 2 \end{bmatrix}$$

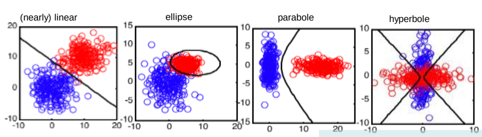
- k - dimensional density:

$$p(x) = \frac{1}{\sqrt{2\pi^k \det(G)}} \exp\left(-\frac{1}{2} (x-\mu)^{\mathrm{T}} G^{-1} (x-\mu) \right)$$

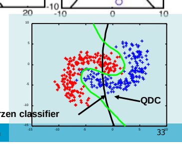
TUDelft

Quadratic discriminant functions

$$\boxed{\mathrm{R(x)} = -\frac{1}{2}(\mathrm{x}-\hat{\mu}_A)^{\mathrm{T}}\hat{\Sigma}_A^{-1}(\mathrm{x}-\hat{\mu}_A) + \frac{1}{2}(\mathrm{x}-\hat{\mu}_B)^{\mathrm{T}}\hat{\Sigma}_B^{-1}(\mathrm{x}-\hat{\mu}_B) + \mathrm{const}}$$

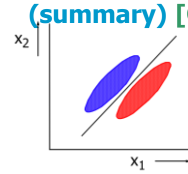


QDC assumes that classes are normally distributed. Wrong decision boundaries are estimated if this does not hold.



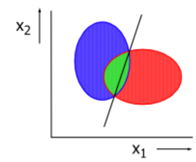
TUDelft

Linear discriminant function (summary) [G]



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

$$\mathrm{R(x)} = (\mu_A - \mu_B)^{\mathrm{T}} \Sigma^{-1} \mathrm{x} + \mathrm{const}$$



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

$$\mathrm{R(x)} = (\mu_A - \mu_B)^{\mathrm{T}} (p(A)\Sigma_A + p(B)\Sigma_B)^{-1} \mathrm{x} + \mathrm{const}$$

TUDelft

Parzen density estimation (1)

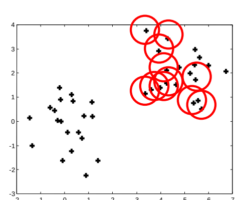
Fix volume of bin, vary positions of bins, add contribution of each bin
Define 'bin'-shape (kernel):

$$K(\mathbf{r}) > 0$$

$$\int K(\mathbf{r})\, d\mathbf{r} = 1$$

For test object $\mathbf{z}$ sum all bins

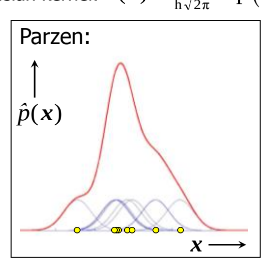
$$p(\mathbf{z}) = \frac{1}{hn} \sum_i K\left(\frac{\mathbf{z} - \mathbf{x}_i}{h} \right)$$



TUDelft

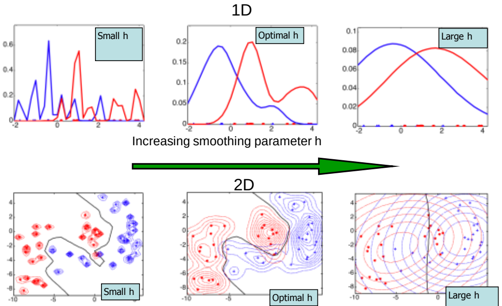
Parzen density estimation (2)

- With Gaussian kernel: $\mathrm{K(x)} = \frac{1}{h\sqrt{2\pi}} \exp\left(-\frac{x^2}{2h^2} \right)$



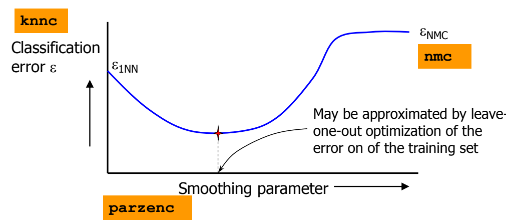
TUDelft

Parzen: density estimates vs the smoothing parameter



TUDelft

Parzen classifier performance

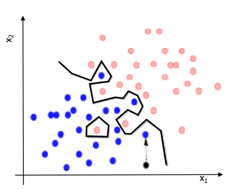


Small smoothing parameters: 1-NN performance
Large smoothing parameters: Nearest mean performance

TUDelft

Nearest neighbor rule (1-NN rule)

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

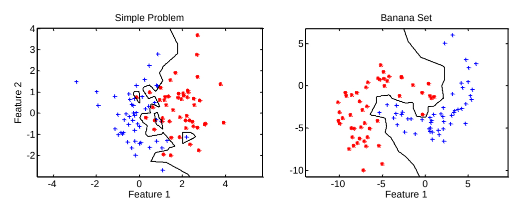


1-NN rule:
- Often relies on the Euclidean distance. Other distance measures can be used.
- Insensitive to prior probabilities!
- Scaling dependent. Features should be scaled properly.

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

TUDelft

Nearest neighbor examples



Good for almost separable classes.
Useful to shape non-linear decision functions.
No training time. Long execution time.
All data should be stored.

TUDelft

Nearest neighbor error

Asymptotically (very large training sets):

$$\varepsilon \le 2\varepsilon^*(1-\varepsilon^*)$$

$$\varepsilon \le 2\varepsilon^*$$

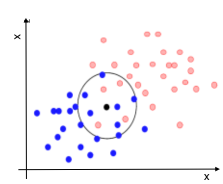
The nearest neighbor classifier will not perform worse than twice the best possible classifier

TUDelft

K-nearest neighbor classifier

knnc

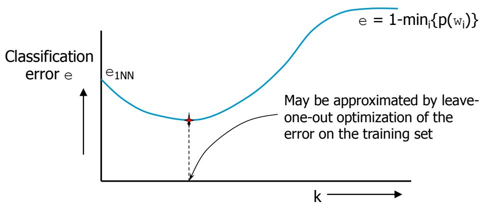
Assign new objects to the class of the majority of the k nearest neighbors in the training set.



More smooth.
Less local.

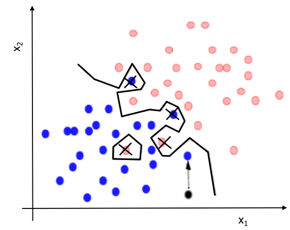
TUDelft

K-nearest-neighbor performance



Prototype selection

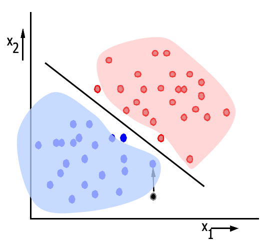
`edicon`



Editting:
Removing some objects may be more accurate.

Prototype selection (2)

`edicon`

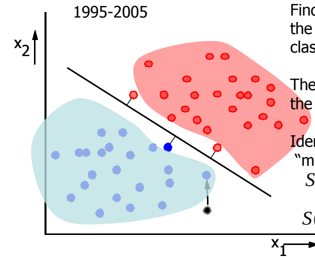


Condensing:
Removing more objects may be faster.

Support vector machine (SVM)

`svc`

1995-2005



Find for linear separable classes the few objects that determine the classifier: the support objects.

They have the same distance to the classifier: the margin.

Identical to "maximum margin classifier"

$$S(x) = \sum_i \alpha_i (x_i^T x)$$

$$S(x) = w^T x, \min(w^T w)$$

Support vector machine (2)

`svc`

$S(x) = \sum_{x_i \in S} \alpha_i (x_i^T x)$ Depends on support objects S only

$S(x) = w^T x, \min(w^T w)$ Minimum norm → maximum margin

$S(x) = w^T x, \min(w^T w) + \sum_{x_j \in E} \xi(x_j)$ Allow some errors E

`svc`: Linear support vector classifier

Non-linear support vector machine: The kernel trick

$S(x) = \sum_{x_i \in S} \alpha_i (x_i^T x) \longrightarrow S(x) = w^T x$ Linear classifier `svc`

$S(x) = \sum_{x_i \in S} \alpha_i K(x_i^T x) \longrightarrow$ Non-linear classifier

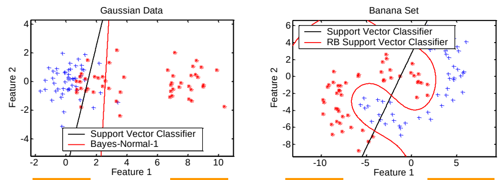
$K(\bullet)$ is a non-linear function of an inner product. A linear classifier in a high-dimensional 'kernel space' is computed, resulting in a non-linear classifier in the feature space.

$K(x_i^T x) = (x_i^T x)^p$ Polynomial classifier `svc`

$K(x_i^T x) = \Phi(\frac{x - x_i}{s})$ Radial basis SVM (about Parzen)

`rbsvc` `parzenc`

SVM: Examples



`svc` `ldc` `svc` `rbsvc`

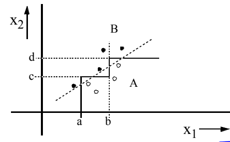
`svc` versus `ldc` for classes with very different domains

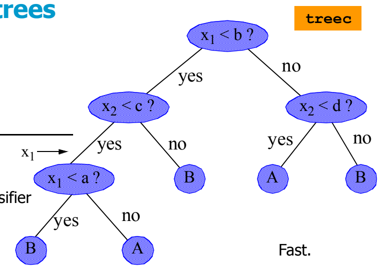
Linear (`svc`) versus nonlinear support vector classifier (`rbsvc`)

TU Delft

Decision trees

`treec`





Implementation of a piece-wise linear classifier

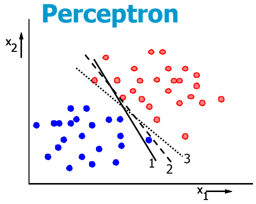
Fast.

Moderate performance

TU Delft

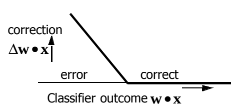
Perceptron

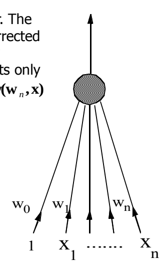
`perlc`



Linear classifier. The weights are corrected for erroneously classified objects only

$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta\mathbf{w}(\mathbf{w}_n, \mathbf{x})$





TU Delft

Neural network classifiers

1985-1995

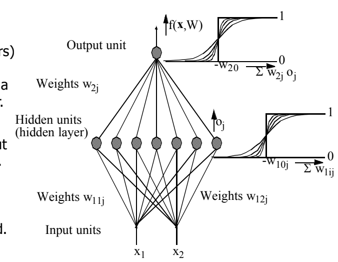
Number of layers with identical neurons (simple linear classifiers) with non-linear transitions in between (sigmoids). Results is a moderately non-linear classifier.

Trained object by object to minimize the MSE on the output compared with targets (labels).

Tricky training procedure.

Slow training and execution, unless special hardware is used.

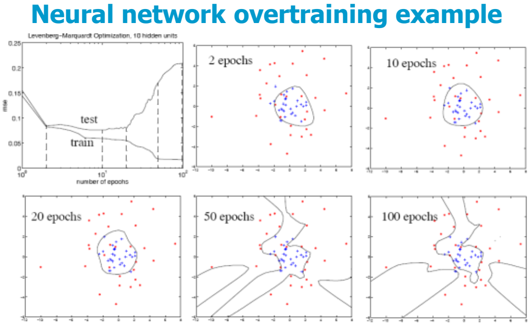
Good performance, danger of overtraining.



`rnnc` `neurc` `bpxnc` `lmnc` `rbnc`

TU Delft

Neural network overtraining example



TU Delft

Classifier outputs

What are the possible outcomes of y = classifier(x)?

- Label, $y_1 \in$ {'apple','banana'}.
- $y_2 \in \{0,1\}$ as crisp numeric labels
- $y_3 \in [0,1]$ for soft labels (confidences)
- $y_4 \in [0,\infty)$ for distances to a class
- $y_5 \in (-\infty,+\infty)$ for distances to a classifier

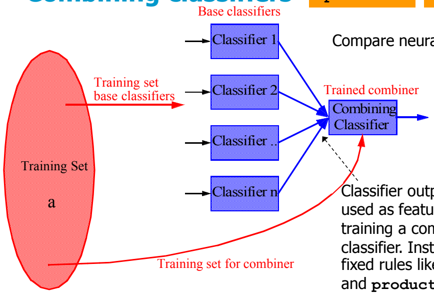
Conversions are often made, e.g.:

```
y2 = (y1 == 'apple')
y2 = round(y3)
y3 = sigm(y5)
y5 = invsigm(y3)
```


TU Delft

Combining classifiers

parallel stacked



Compare neural networks

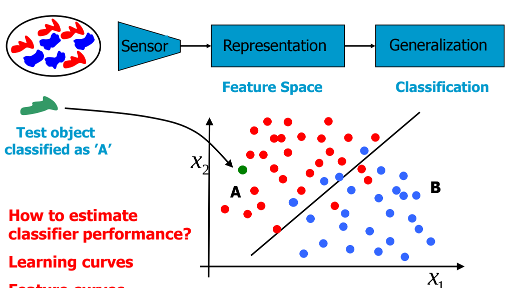
Classifier outputs may be used as features for training a combining classifier. Instead often fixed rules like **max**, **sum** and **product** are used.

TUDelft

Evaluation

TUDelft

Classifier evaluation



How to estimate classifier performance?

Learning curves

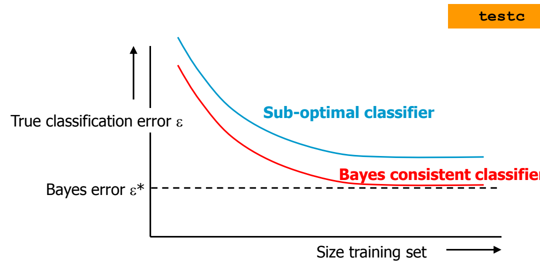
Feature curves

TUDelft

Learning Curve

cleval

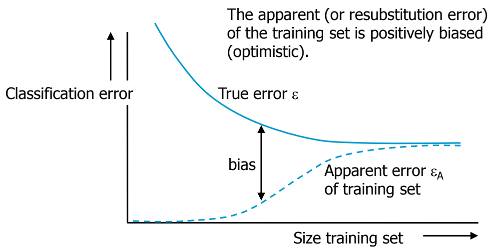
testc



TUDelft

The Apparent Classification Error

The apparent (or resubstitution error) of the training set is positively biased (optimistic).



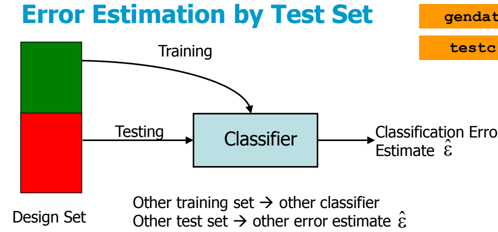
An independent test set is needed!

TUDelft

Error Estimation by Test Set

gendat

testc



Other training set → other classifier

Other test set → other error estimate $\hat{\varepsilon}$

$$\sigma_{\hat{\varepsilon}}^2 = \mathrm{Var}(\hat{\varepsilon}\,|\,\text{test set size } N) = \frac{\varepsilon(1-\varepsilon)}{N} \qquad \sigma_{\hat{\varepsilon}} = \sqrt{\frac{\varepsilon(1-\varepsilon)}{N}}$$

N \ ε	0.01	0.03	0.1
10	0.031	0.054	0.095
100	0.010	0.017	0.003
1000	0.003	0.005	0.009

TUDelft

Training Set Size ←→ Test Set Size

- Training set should be large for good classifiers.
- Test set should be large for a reliable, unbiased error estimate.
- In practice often just a single design set is given

`gendat`

Same set for training and testing may give a good classifier, but will definitely yield an optimistically biased error estimate.

A small, independent test set yields an unbiased, but unreliable (large variance) error estimate for a well trained classifier.

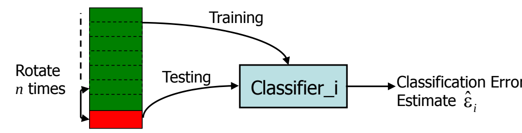
A large, independent test set yields an unbiased and reliable (small variance) error estimate for a badly trained classifier.

50-50 is an often used standard solution for the sizes for training and testing. Is it really good?
There are better alternatives.

TU Delft

Crossvalidation

`crossval`



Size test set $1/n$ of design set.

Size training set is $(n - 1)/n$ of design set.

Train and test n test times. Average errors. (Good choice: $n = 10$)

All objects are tested ones → most reliable test result that is possible.

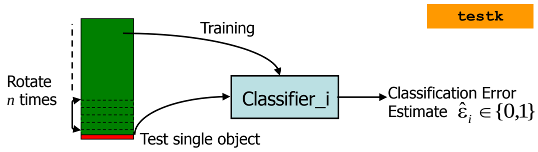
Final classifier: trained by all objects → best possible classifier.

Error estimate is slightly pessimistically biased.

TU Delft

Leave-one-out Procedure

`crossval`
`testk`

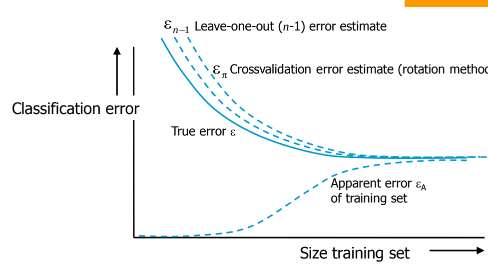


Crossvalidation in which n is total number of objects.
One object tested at a time.
n classifiers to be computed.
In general unfeasible for large n.
Doable for k-NN classifier (needs no training).

TU Delft

Expected Learning Curves by Estimated Errors

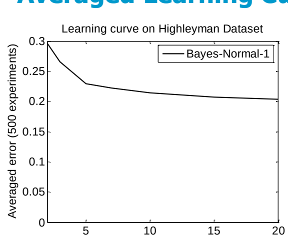
`cleval`



TU Delft

Averaged Learning Curve

`cleval`



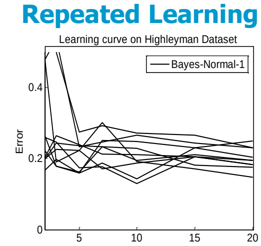
For obtaining 'theoretically expected' curves many repetitions are needed.

```
a = gendath([200 200]);
e = cleval(a,ldc,[2,3,5,7,10,15,20],500);
plote(e);
```


TU Delft

Repeated Learning Curves

`cleval`
`plote`



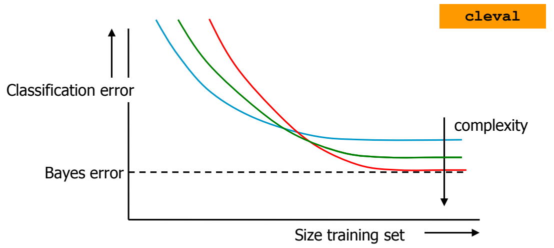
Small sample sizes have a very large variability.

```
a = gendath([200 200]);
for j=1:10
  e = cleval(a,ldc,[2,3,5,7,10,15,20],1);
  hold on; plote(e);
end
```


TU Delft

Learning Curves for Different Classifier Complexity

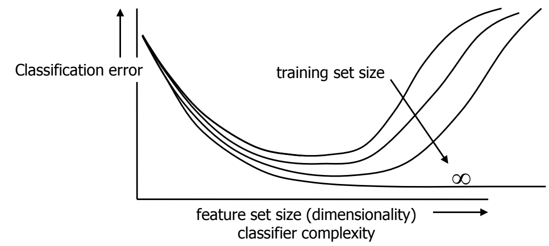
`cleval`



More complex classifiers are better in case of large training sets
and worse in case of small training sets

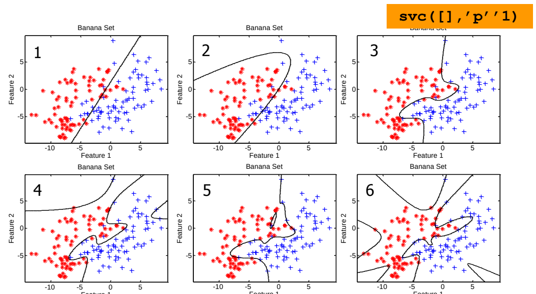
Peaking Phenomenon, Overtraining
Curse of Dimensionality, Rao's Paradox

`clevalf`



Example Overtraining, Polynomial Classifier

`svc([],'p',1)`

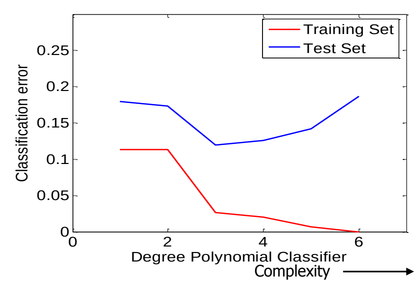


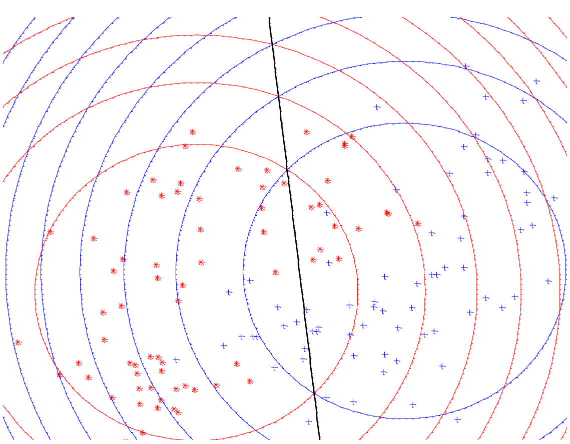
Example Overtraining (2)

`gendat`

`svc([],'p',1)`

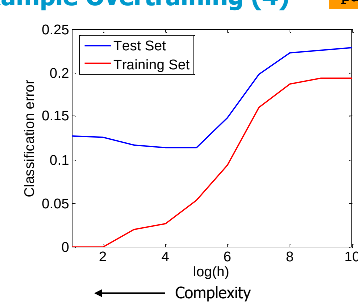
`testc`



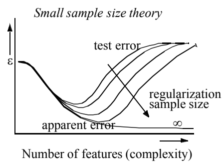


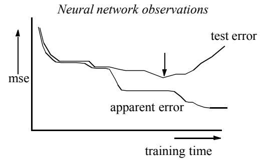
Example Overtraining (4)

`parzenc([],h)`

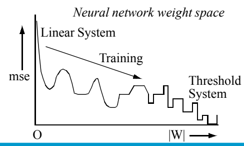


Neural Network Understanding



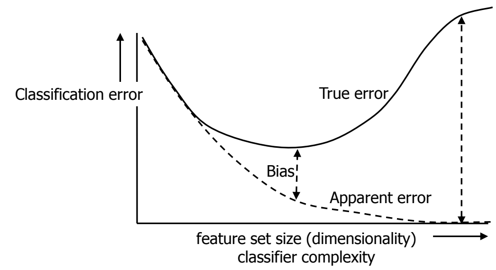


Conclusion: the effective complexity of a neural network increases during training.



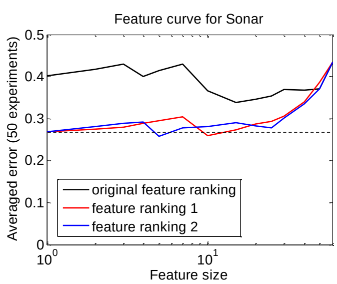
TUDelft

Overtraining ←→ Increasing Bias



TUDelft

Example Curse of Dimensionality



Fisher classifier for
Various feature rankings

TUDelft

Conclusions on Evaluation

- Larger training sets yield better classifiers.
- Independent test sets are needed for obtaining unbiased error estimates.
- Larger test sets yield more accurate error estimates.
- Leave-one-out crossvalidation seems to be an optimal compromise, but might be computationally infeasible.
- 10-fold cross-validation is a good practice.
- More complex classifiers need larger training sets to avoid overtraining.
- This holds in particular for larger feature sizes, due to the curse of dimensionality.
- For too small training sets, more imple classifiers or smaller feature sets are needed.

TUDelft