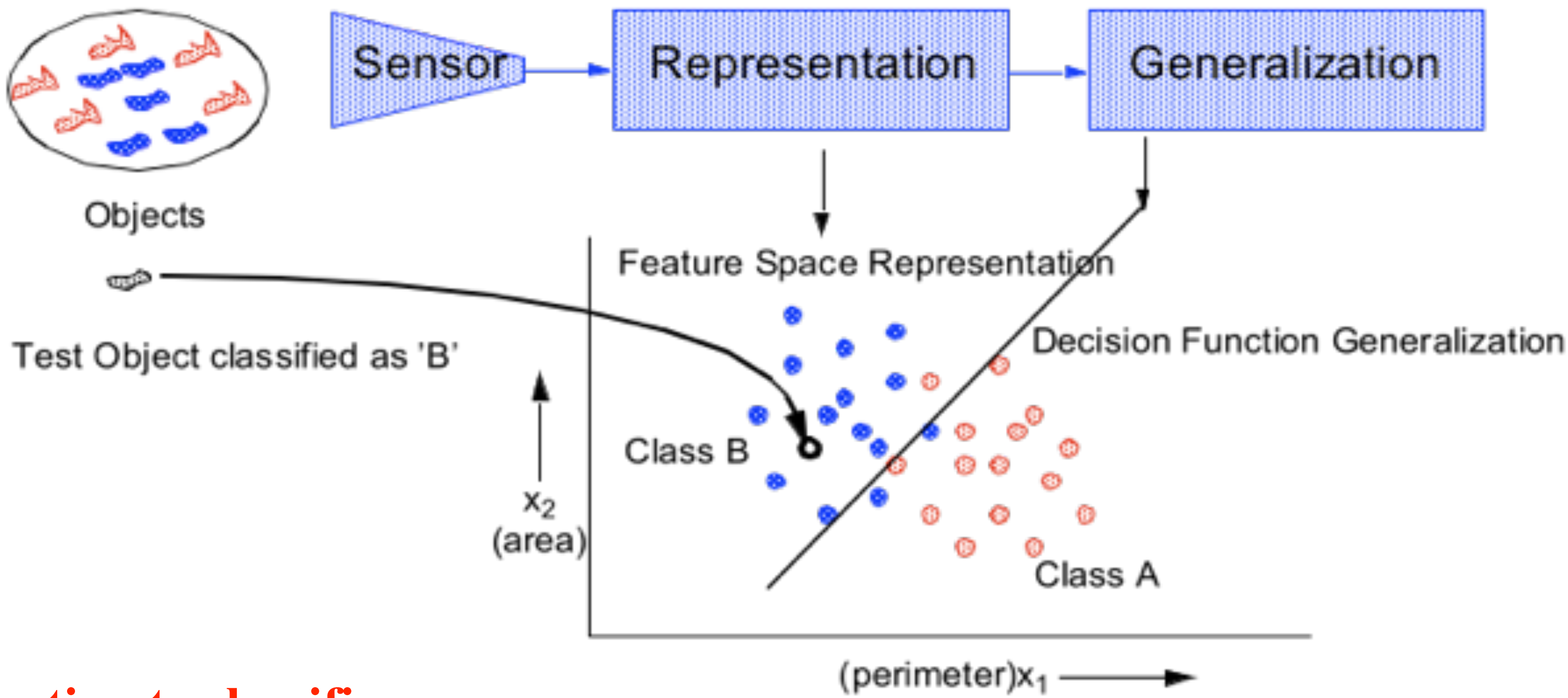
Classifier Evaluation

David Tax

PR Laboratory
D.M.J.Tax@tudelft.nl

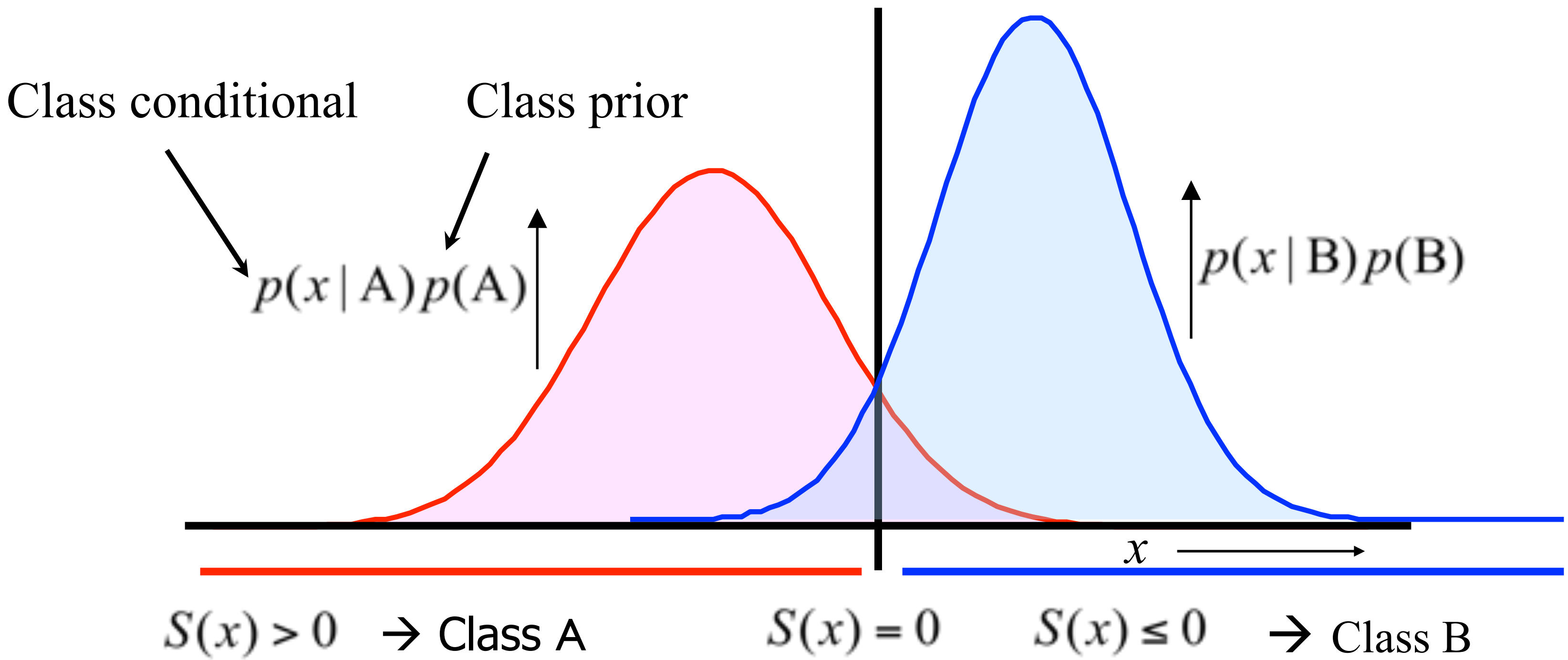
Classifier Evaluation



- **How to estimate classifier performance.**

- **Learning curves**
- **Feature curves**
- **Reject and ROC curves**

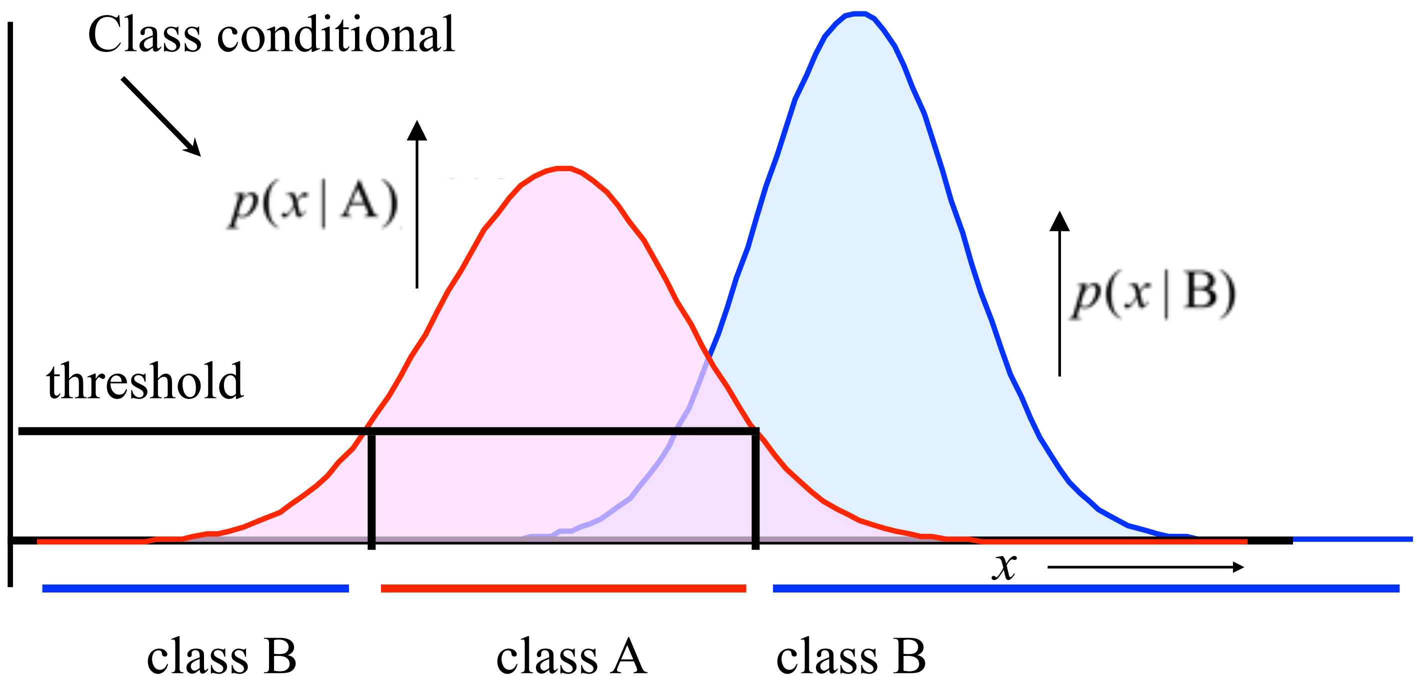
How do we make decision? Weighting



The soft outputs of the classes are multiplied by the class priors (chosen weights).

The samples are assigned to the class for which the resulting soft output is higher.

How do we make decision? Thresholding



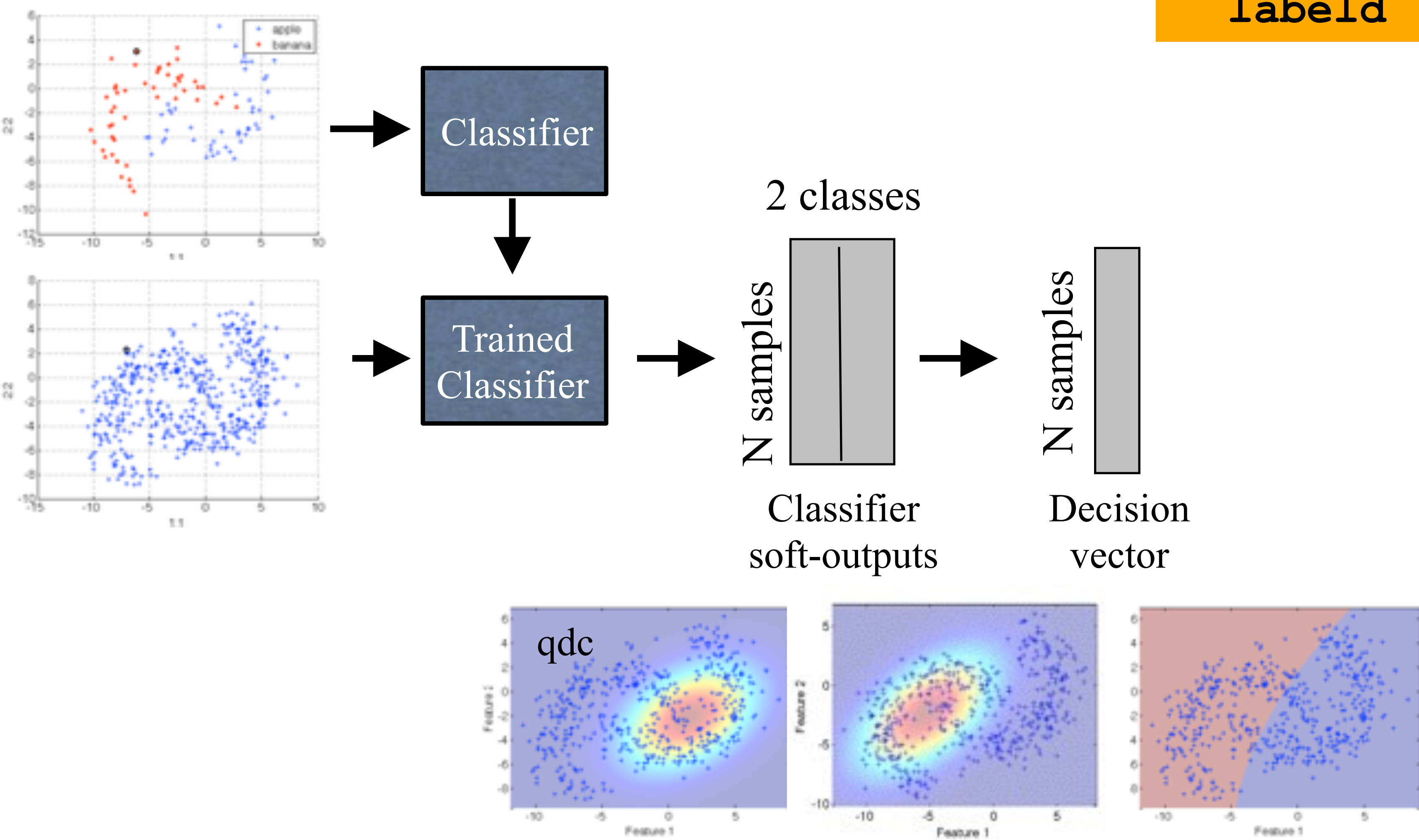
Only the class of interest A is considered.

The threshold value is chosen.

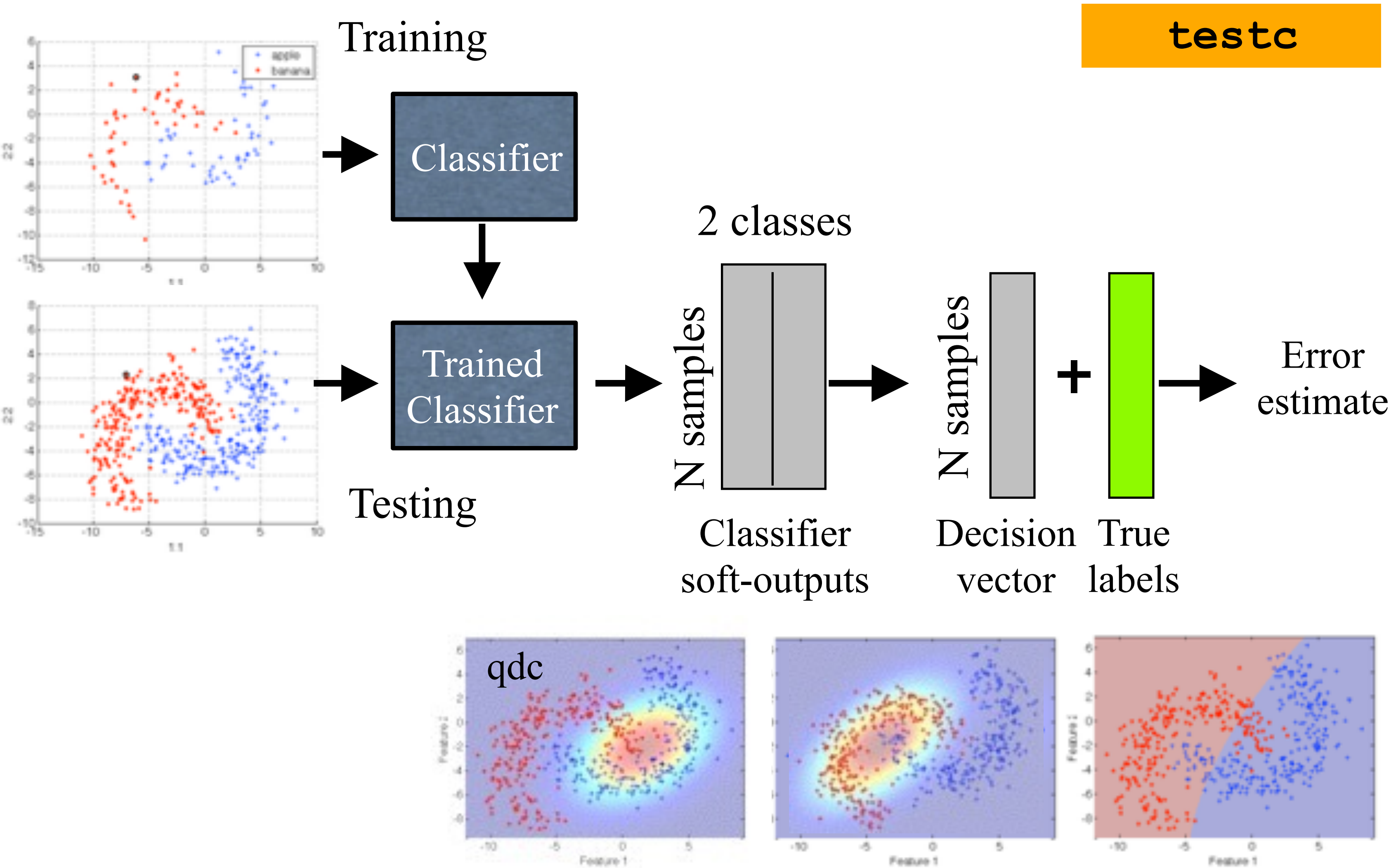
Samples above the threshold are labelled as A, below the threshold as B.

How do we make decision?

labeld

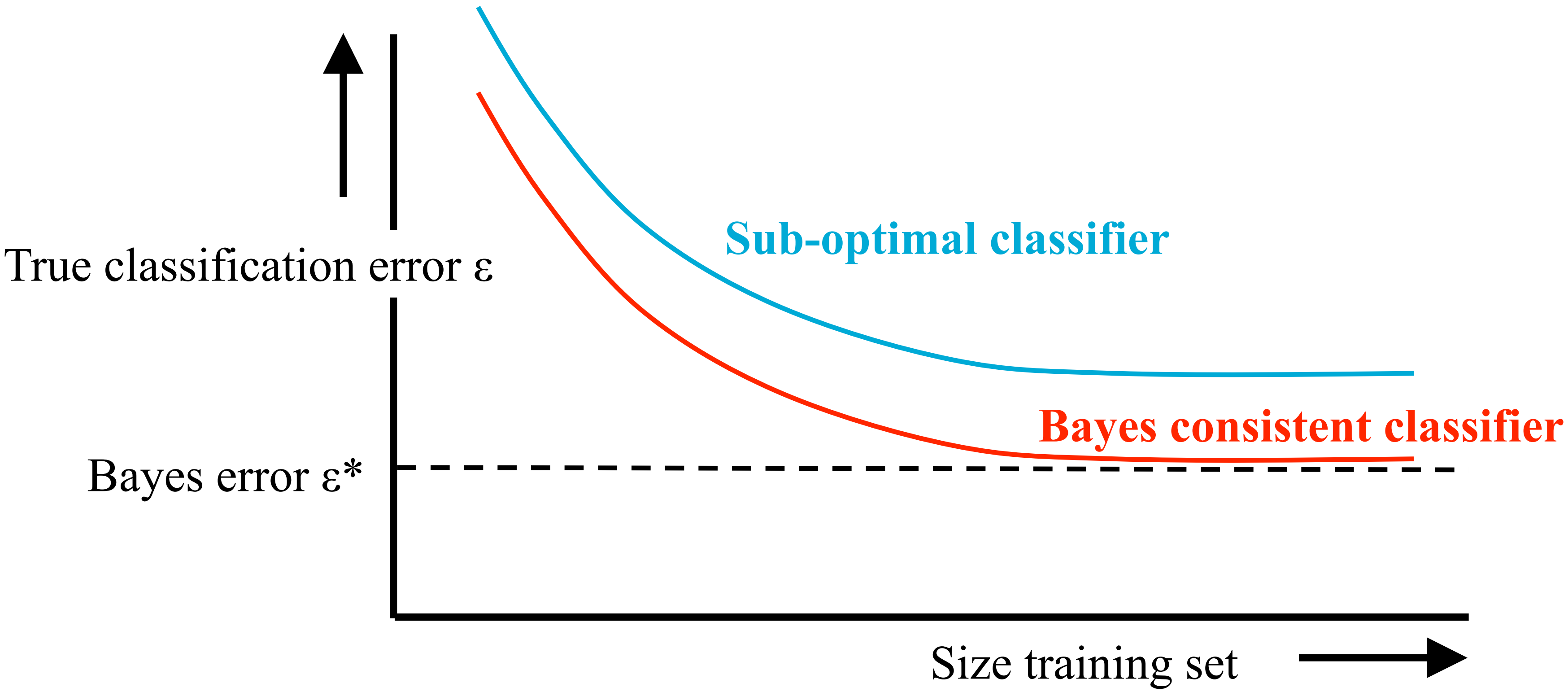


How do we make decision?



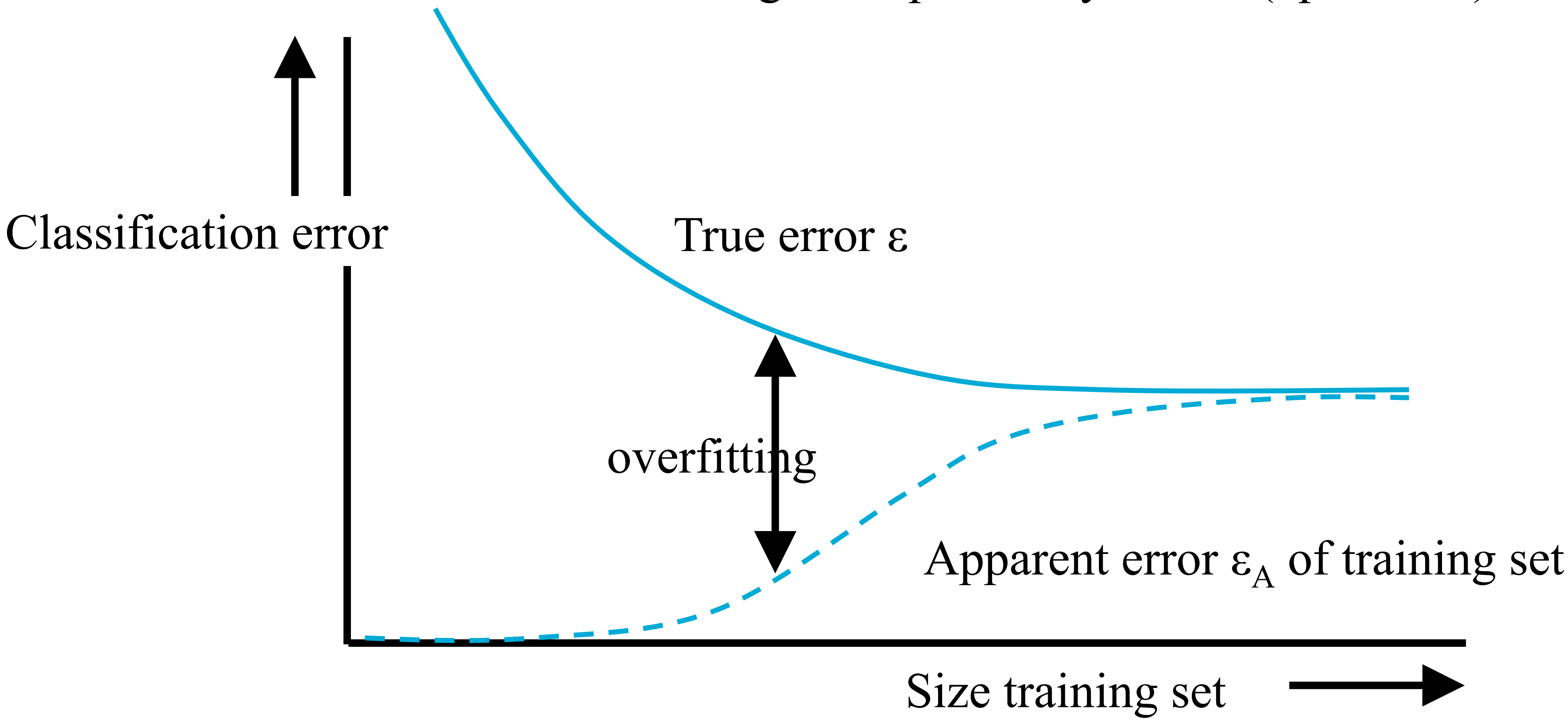
Learning Curve

cleval



The Apparent Classification Error

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

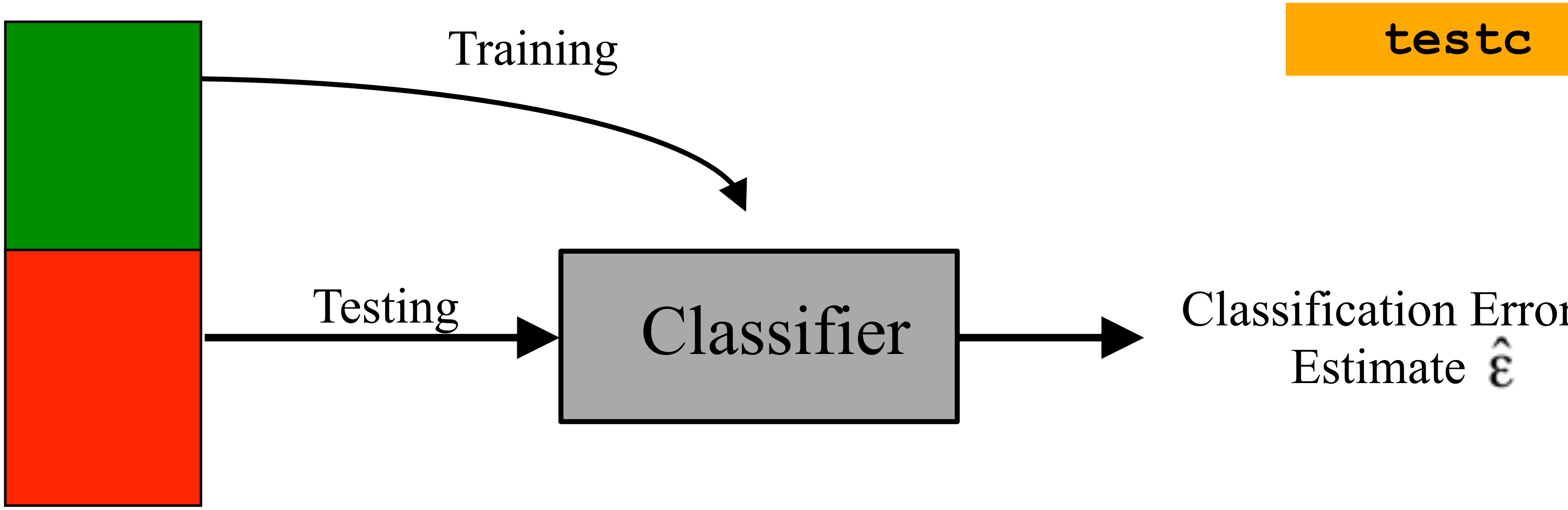


An independent test set is needed!

Error Estimation by Test Set

`gendat`

`testc`



Design Set

Other training set → other classifier

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

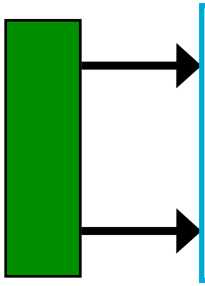
$$\sigma_{\hat{\varepsilon}}^2 = \mathrm{Var}(\hat{\varepsilon} \mid \text{test set size } N) = \frac{\varepsilon(1-\varepsilon)}{N} \qquad \sigma_{\hat{\varepsilon}} = \sqrt{\frac{\varepsilon(1-\varepsilon)}{N}}$$

N	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

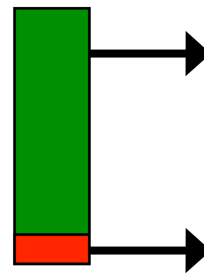
Training Set Size ←→ Test Set Size

gendat

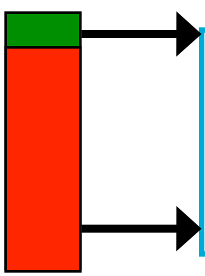
- Training set should be large for good classifiers.
- Test set should be large for a reliable, unbiased error estimate.
- In practice just a single design set is given



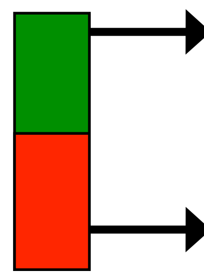
Same set for training and testing may give a good classifier, but will definitely yield an optimistically biased error estimate.



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



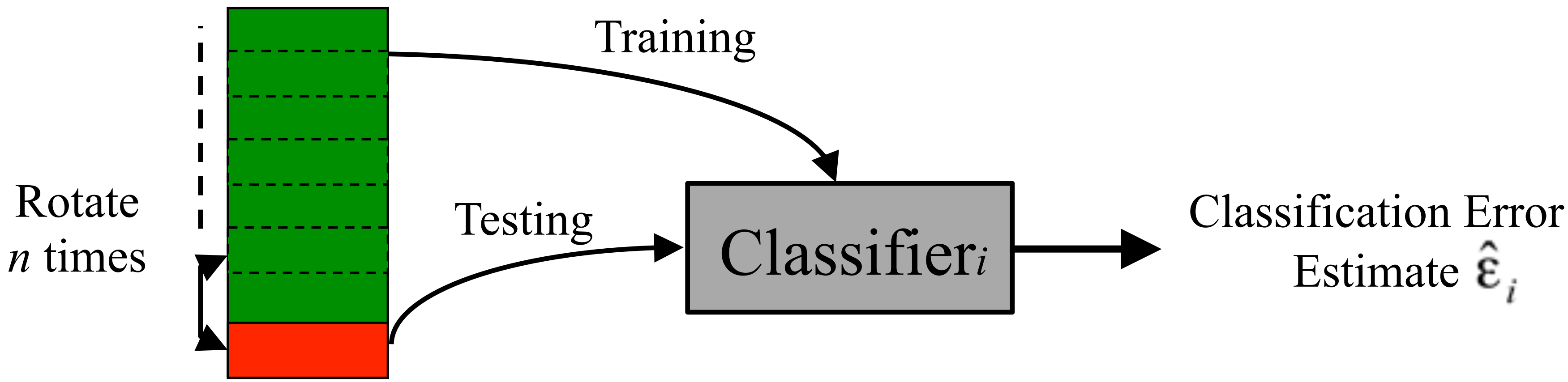
A large, independent test set yields an unbiased and reliable (small variance) error estimate for a badly trained classifier.



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

Cross-validation

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. (Default choice: n = 10)

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

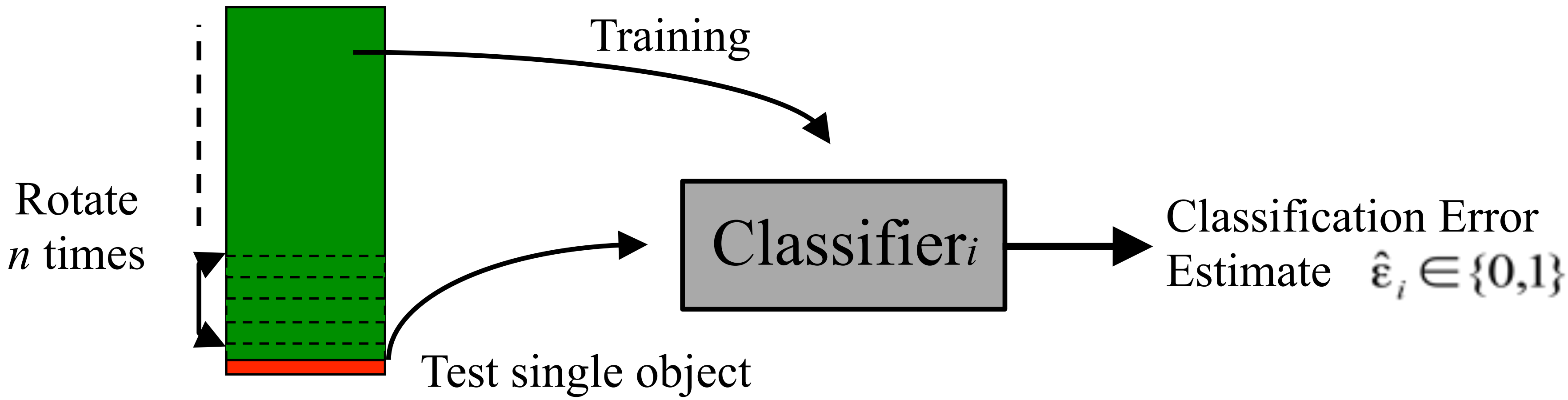
Final classifier: Trained on all objects → the best possible classifier.

Error estimate is slightly pessimistically biased.

Leave-one-out Procedure

`crossval`

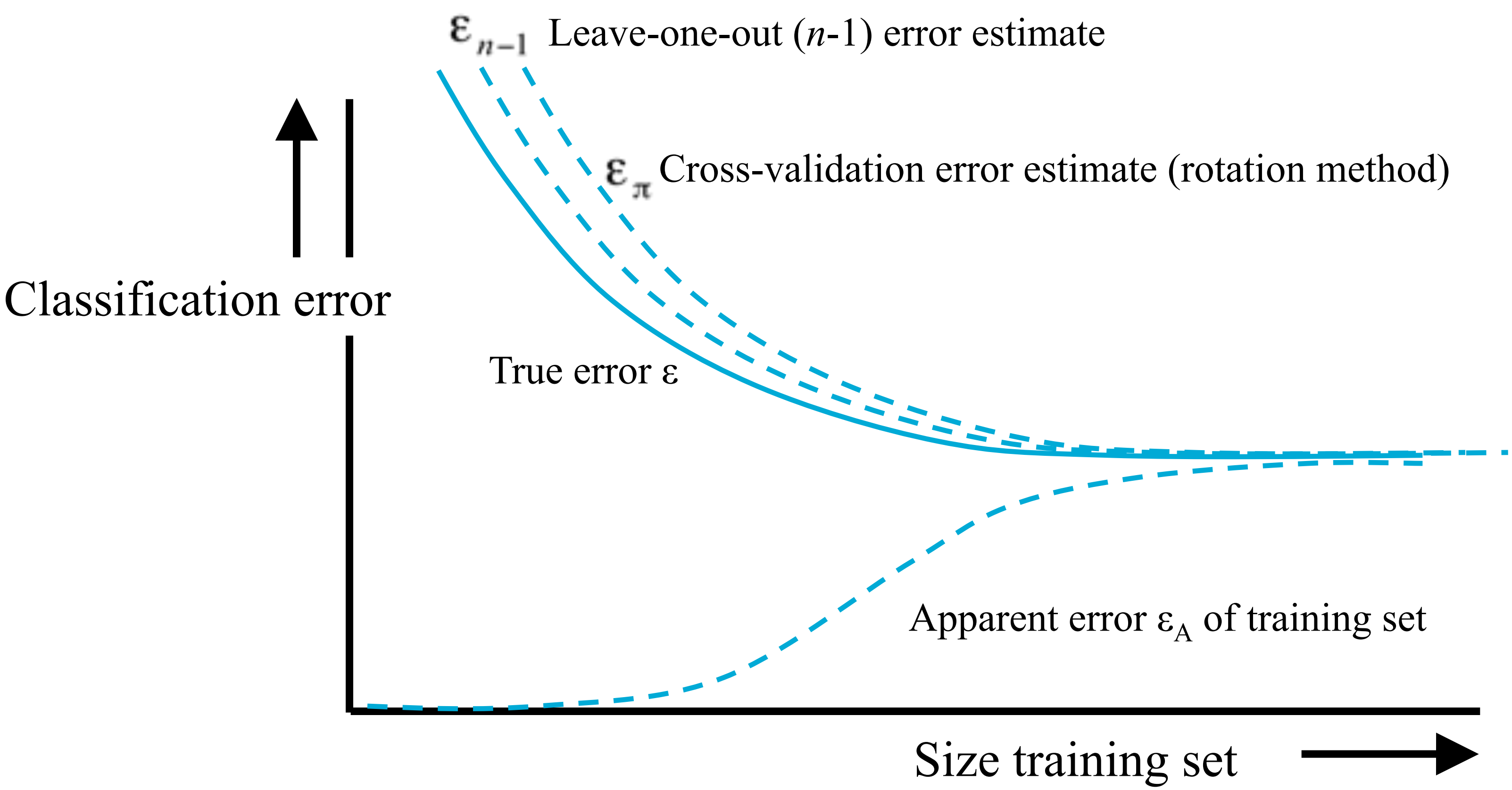
`testk`



Cross-validation 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).

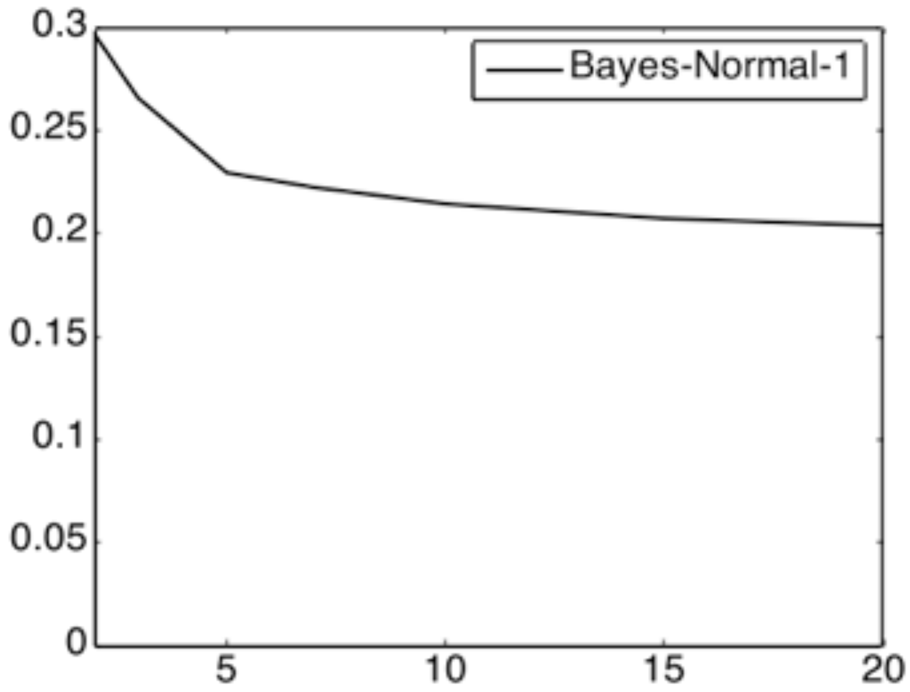
Expected Learning Curves by Estimated Errors

cleval



Averaged Learning Curve

cleval



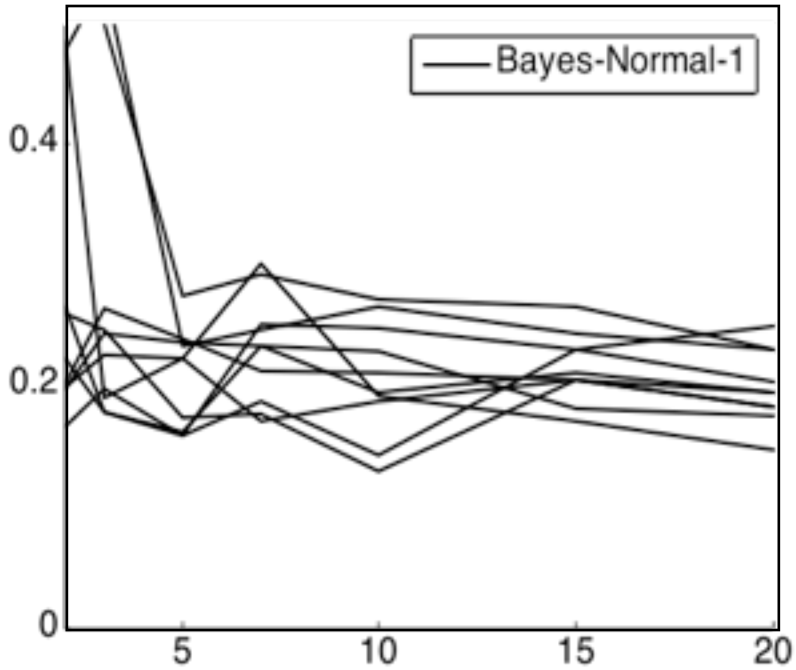
For obtaining 'theoretically expected' curves many repetitions are needed.

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

Repeated Learning Curves

cleval

plote

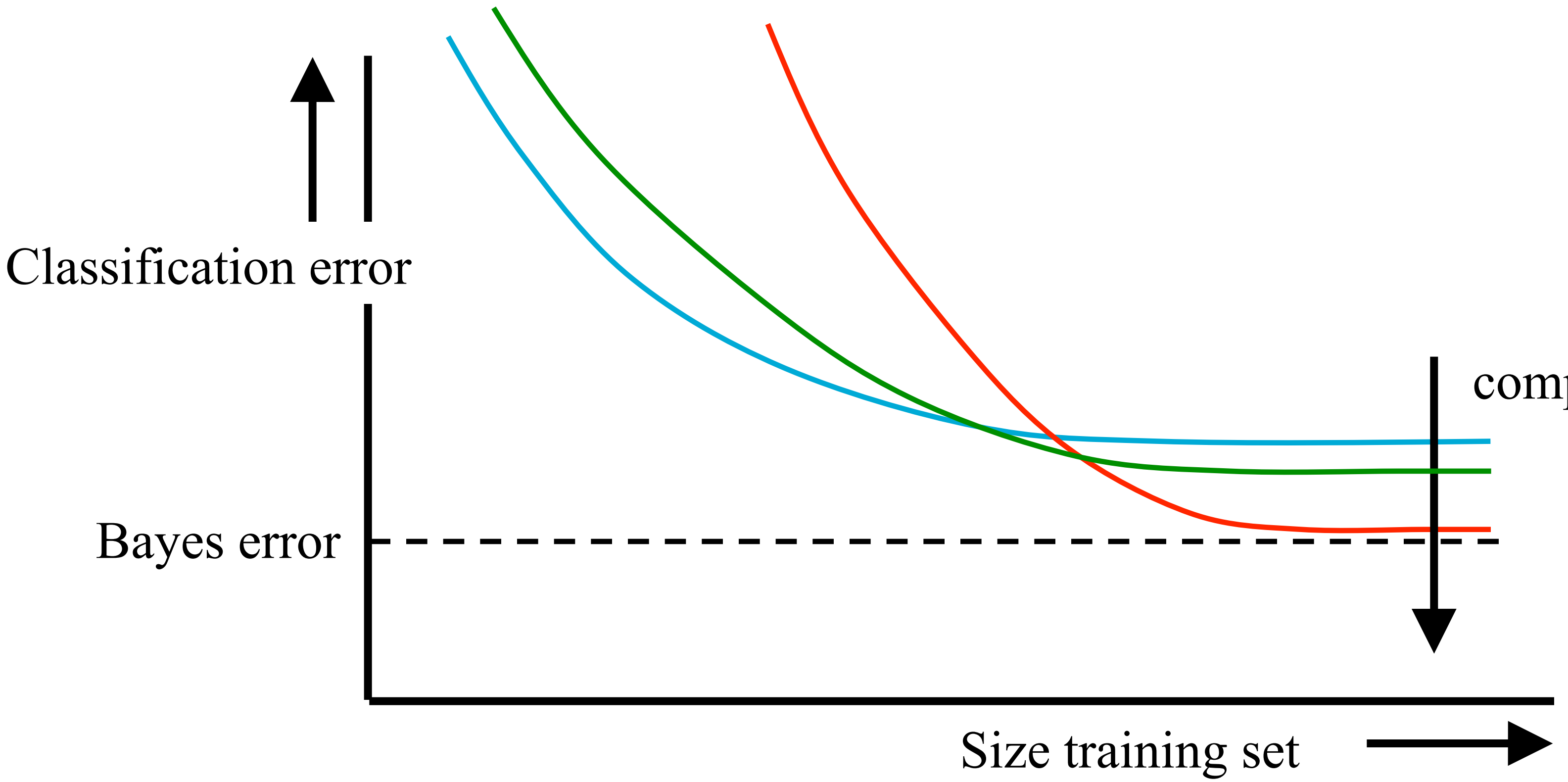


Small sample sizes have a very large variability.

```
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
```

Learning Curves for Different Classifier Complexity

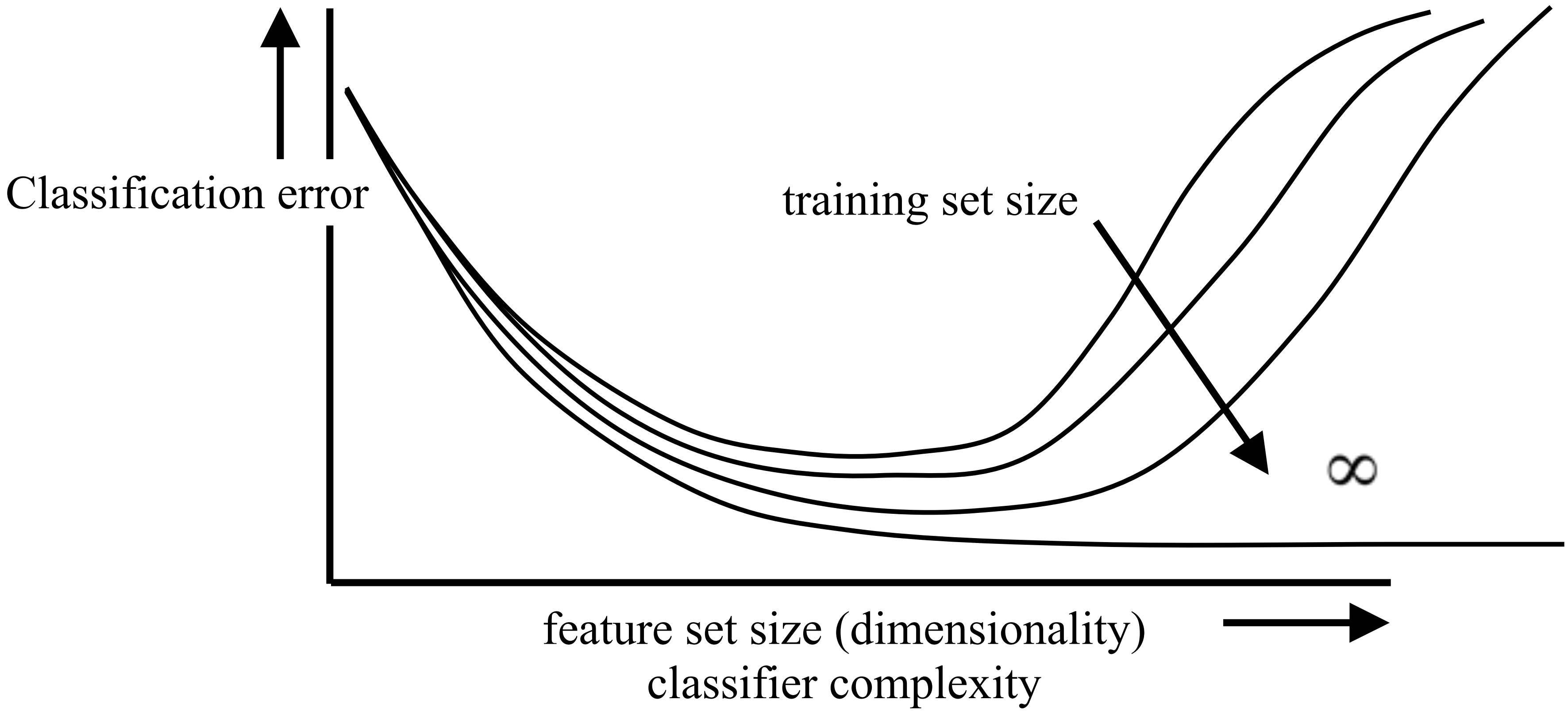
cleval



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

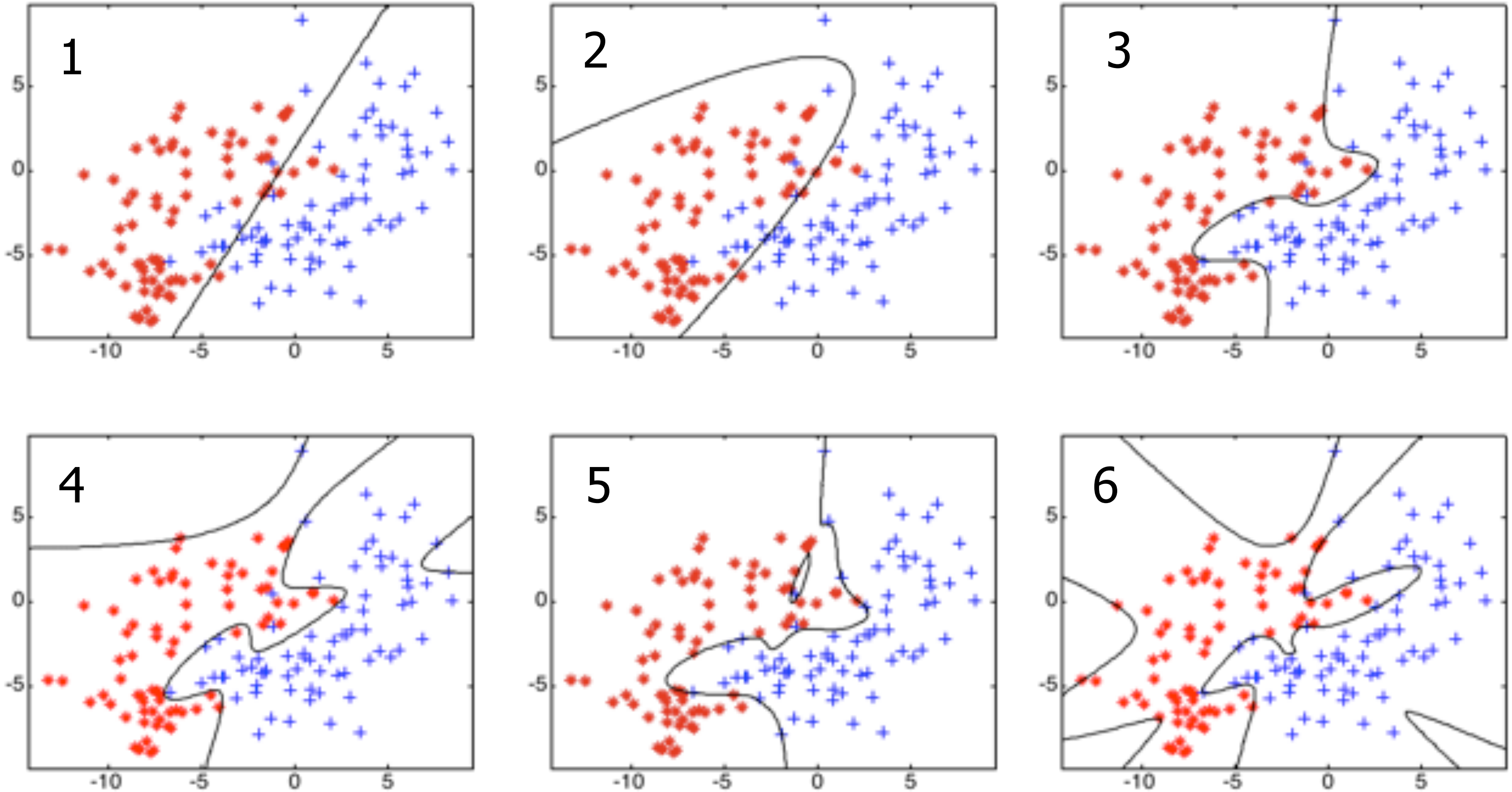
Peaking Phenomenon, Overtraining
Curse of Dimensionality, Rao's Paradox

`clevalf`



Example Overtraining, Polynomial Classifier

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

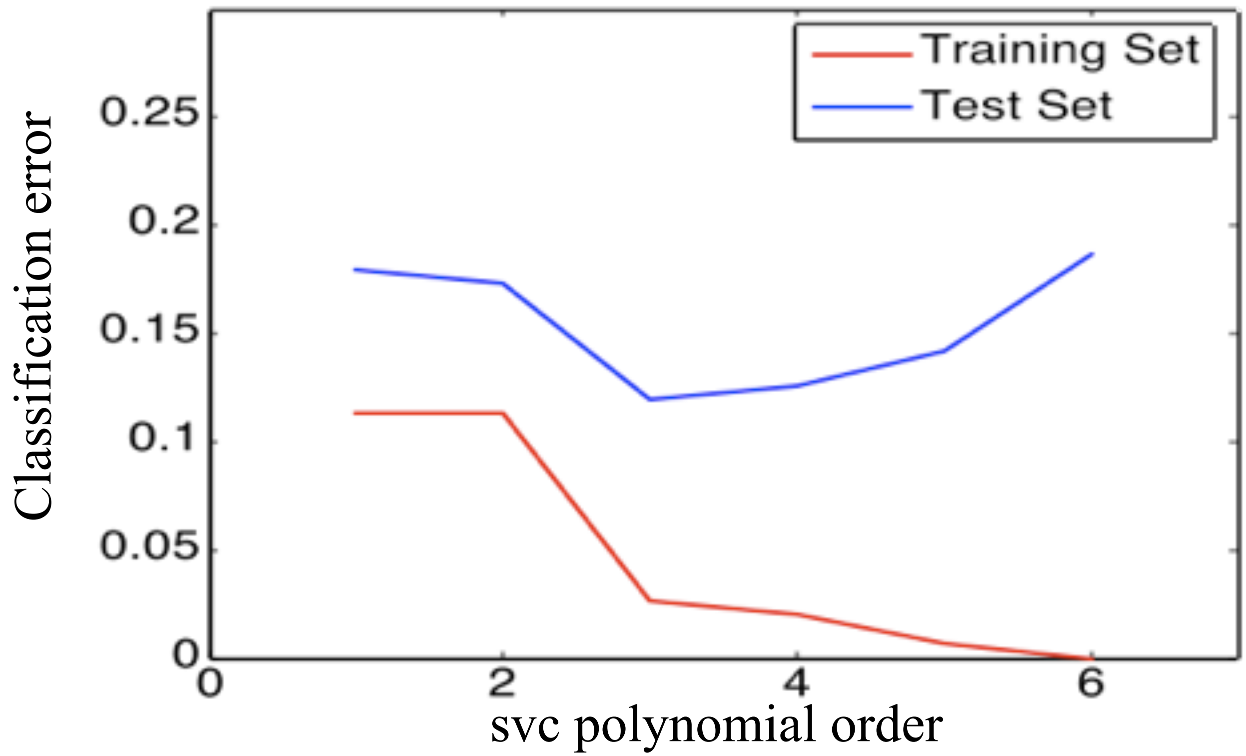


Example Overtraining (2)

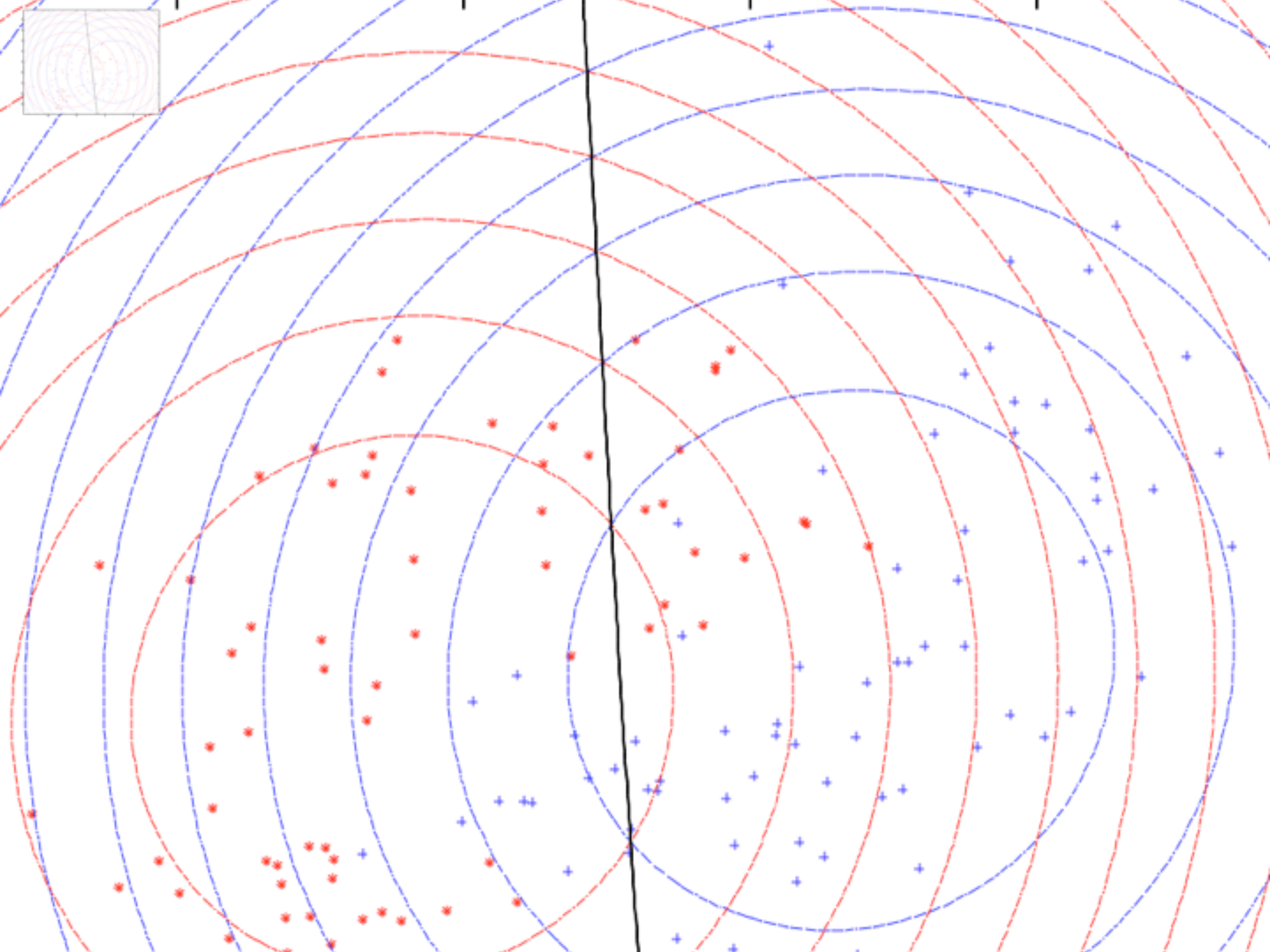
`gendat`

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

`testc`

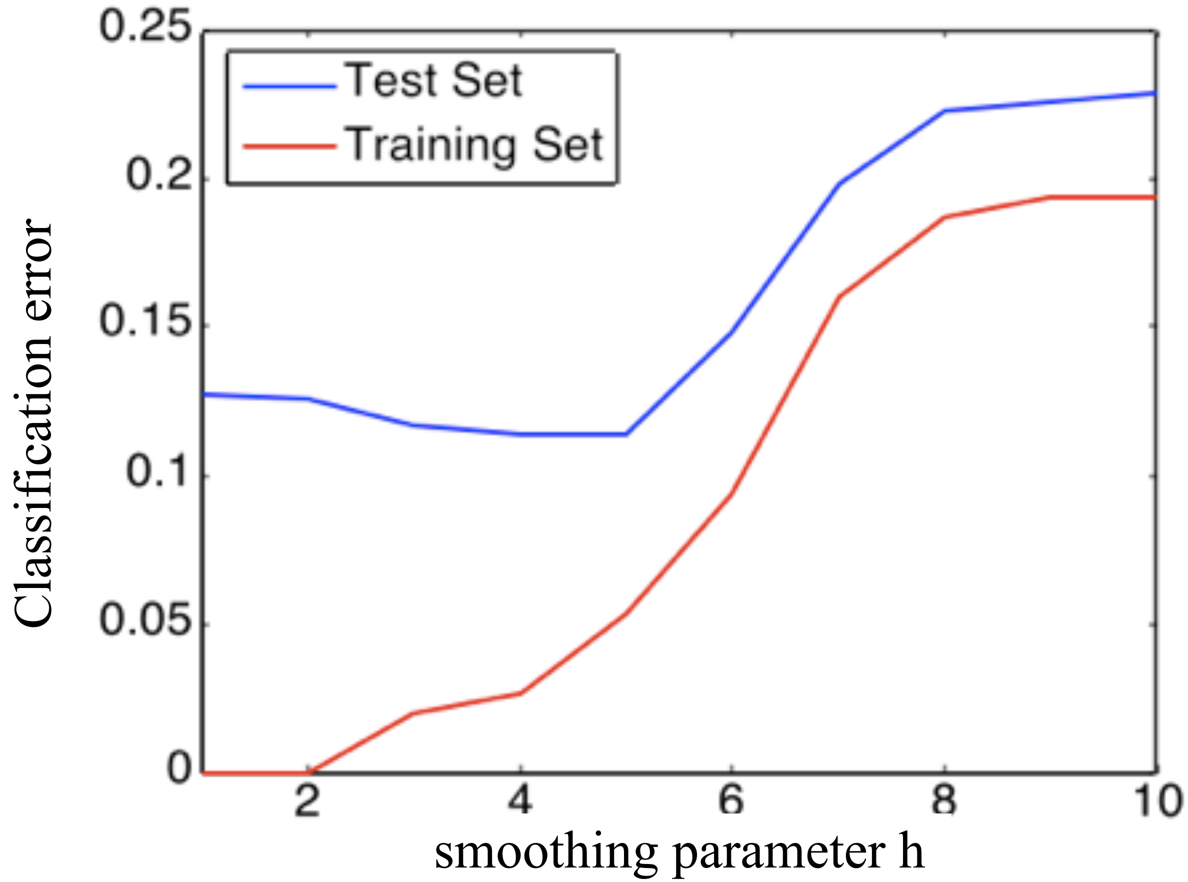


Complexity



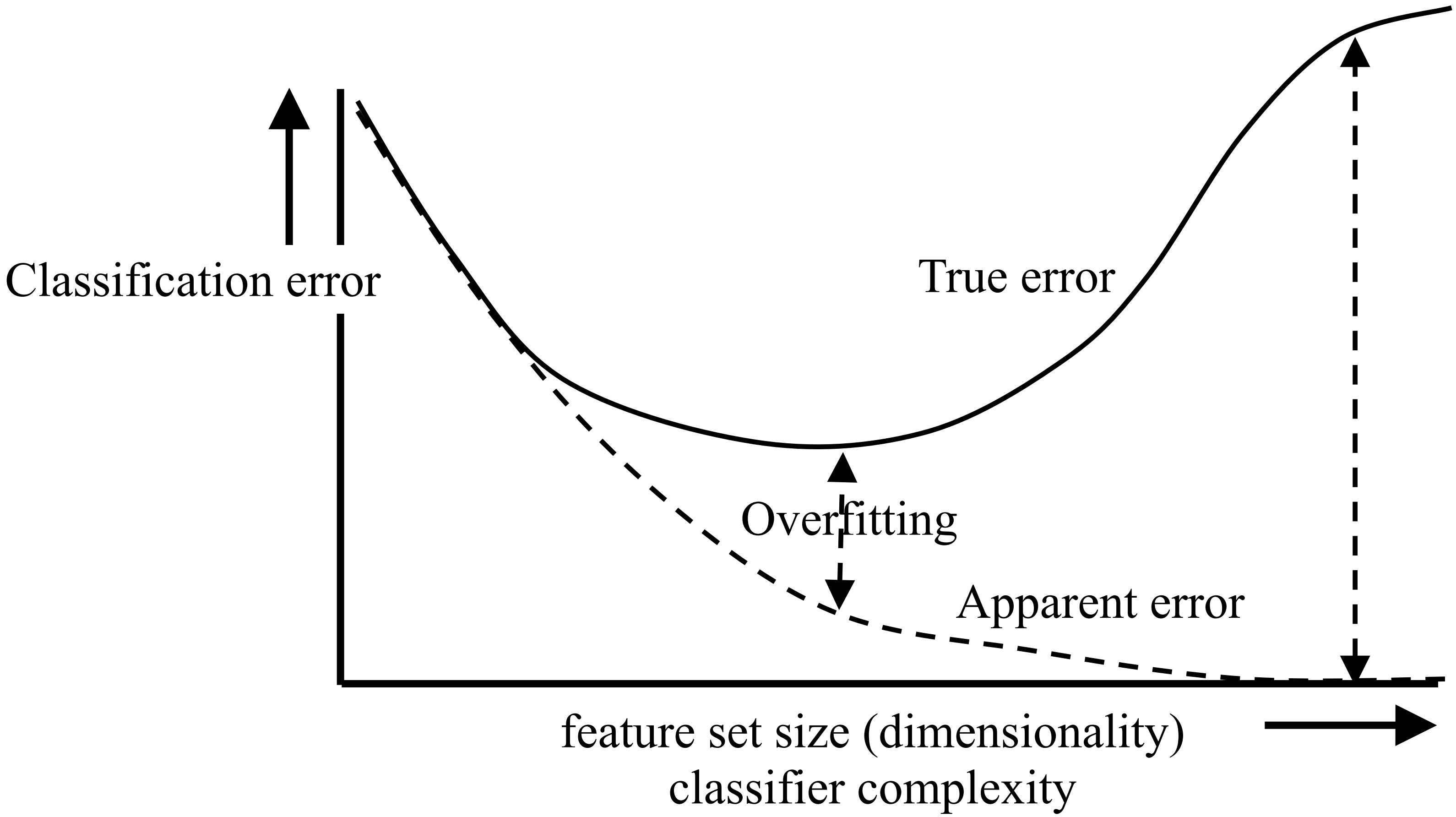
Example Overtraining (4)

`parzenc([],h)`

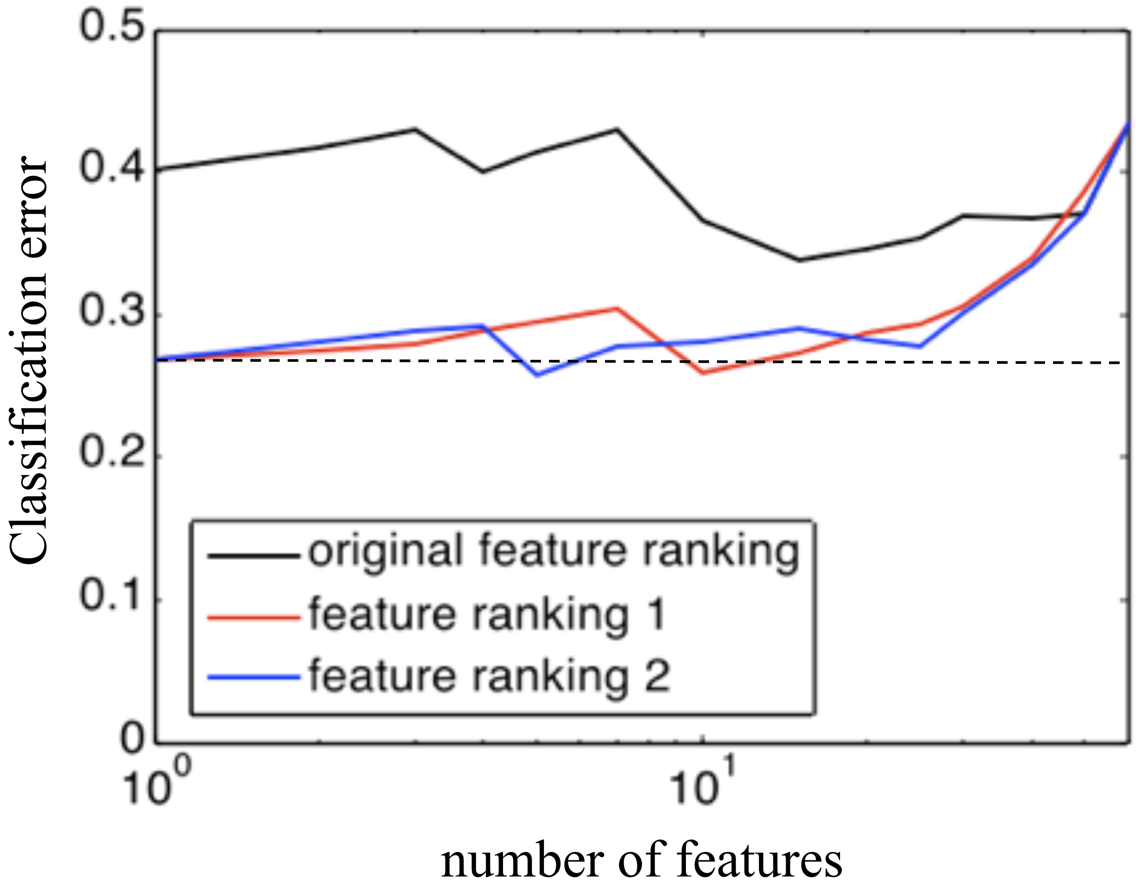


← Complexity

Overtraining ←→ Increasing Bias



Example Curse of Dimensionality



Fisher classifier for various feature rankings

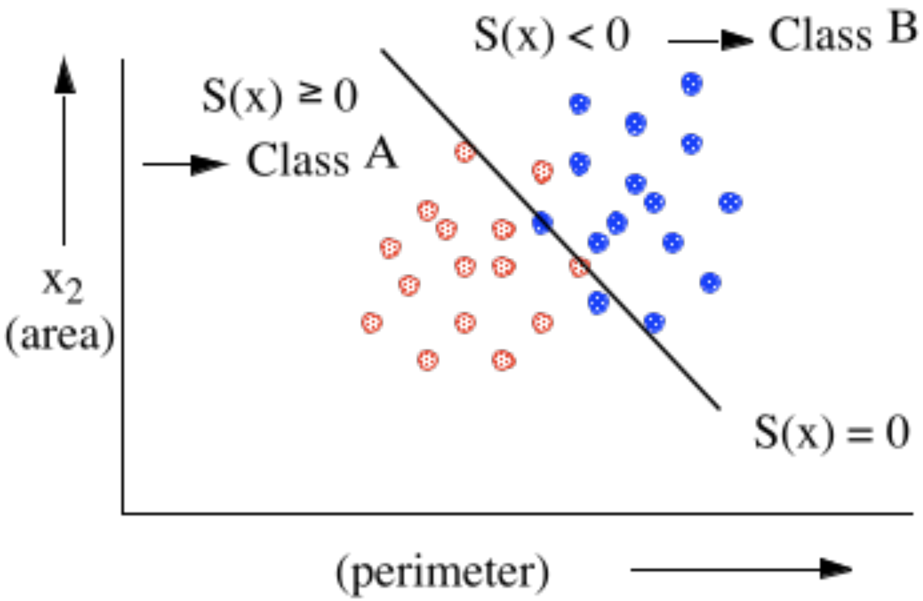
Confusion Matrix (1)

`confmat`

real labels   obtained labels

$$\Lambda = \begin{bmatrix} \lambda_1 \\ \dots \\ \lambda_N \end{bmatrix} \qquad L = \begin{bmatrix} l_1 \\ \dots \\ l_N \end{bmatrix}$$

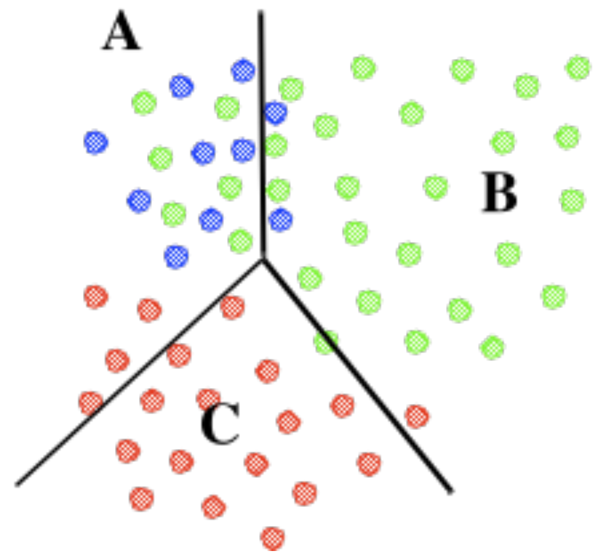
$$\lambda, l \in \{\pi_1, \dots, \pi_K\}$$



Confusion matrix: $$C = \begin{bmatrix} c_{11} & \dots & c_{1K} \\ \dots & \dots & \dots \\ c_{K1} & \dots & c_{KK} \end{bmatrix} \qquad c_{ij} = N \times \mathrm{Prod}(x \in \pi_j \mid \pi_i)$$

Confusion Matrix (2)

$$N_A = 10,\ N_B = 30,\ N_C = 20$$



testc

$$E = \frac{c_{12} + c_{13} + c_{21} + c_{23} + c_{31} + c_{32}}{N_A + N_B + N_C}$$

$$E = 14/60 = 0.2333$$

confmat

$C = \text{confmat}(\Lambda, L)$

Λ real labels

L obtained labels

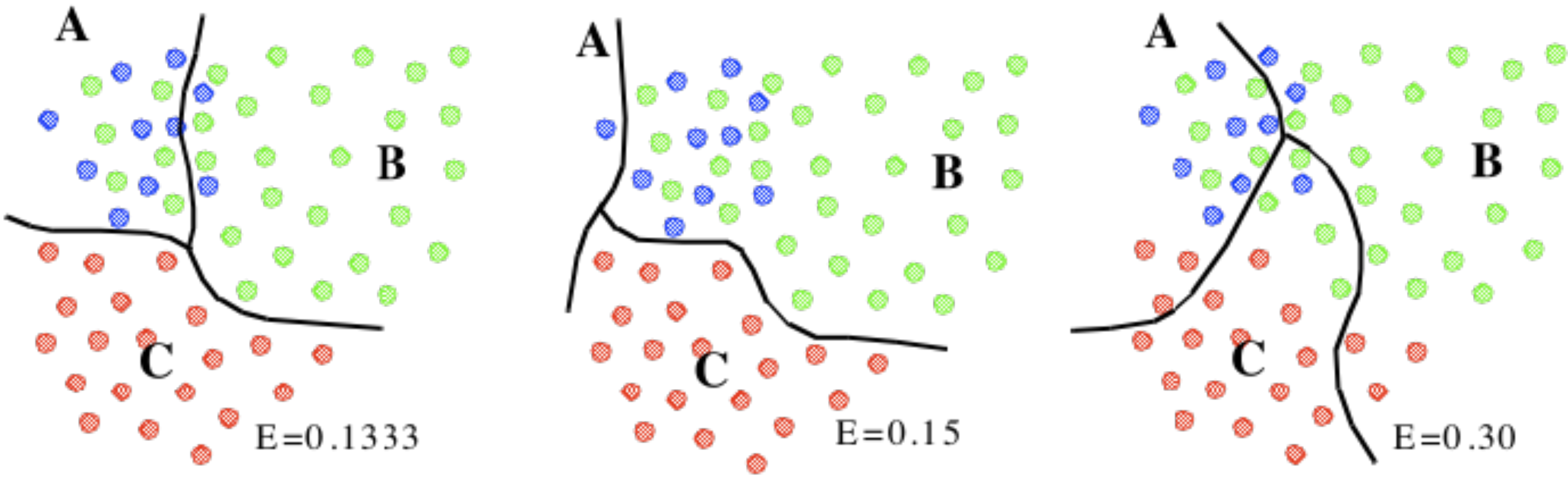
classified to

objects from	A	B	C
class A	8	2	0
class B	6	23	1
class C	4	1	15

0.20 error in class A
0.23 error in class B
0.25 error in class C

0.228 averaged error

Confusion Matrix (3)



objects from

classified to

	A	B	C	
class A	8	2	0	0.20
class B	6	24	0	0.20
class C	0	0	20	0.00
				0.133

classified to

	A	B	C	
class A	1	9	0	0.9
class B	0	30	0	0.0
class C	0	0	20	0.0
				0.30

classified to

	A	B	C	
class A	7	2	1	0.30
class B	5	21	4	0.30
class C	3	2	15	0.30
				0.30

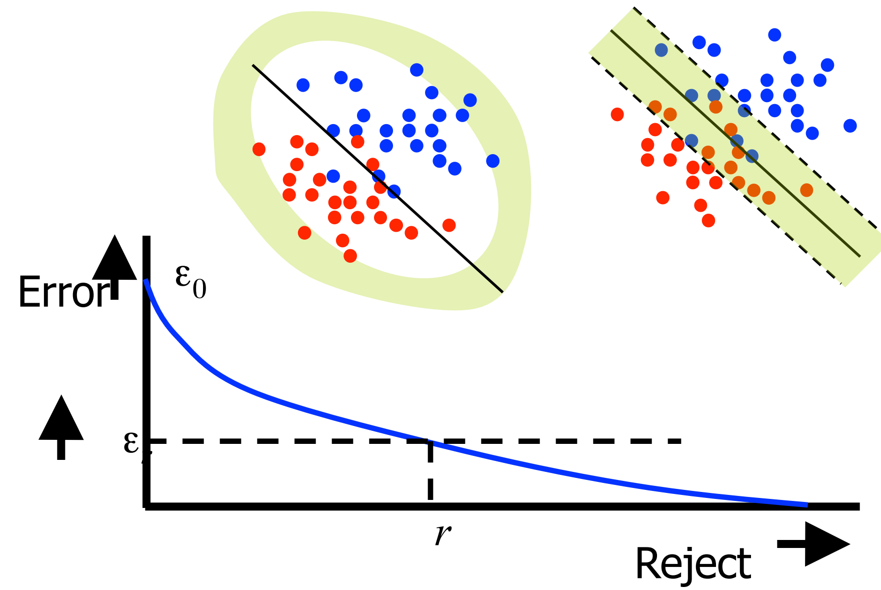
classification details are only observable in the confusion matrix!!

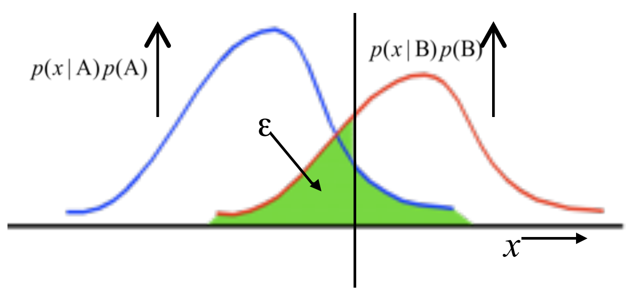
Conclusions on Error Estimations

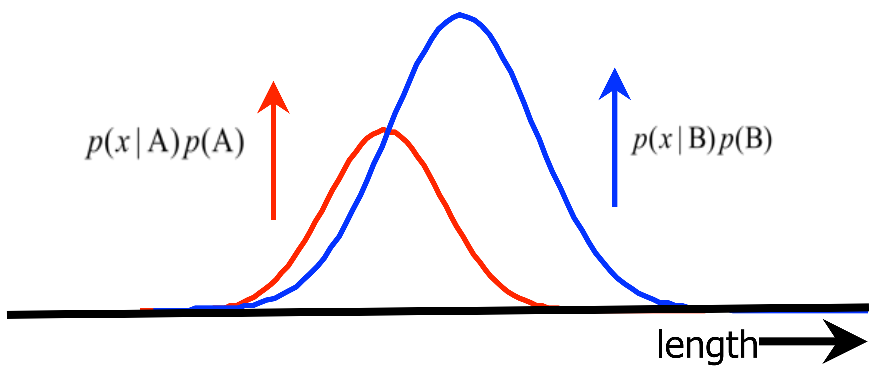
- Larger training sets yield better classifiers.
- Independent test sets are needed for obtaining unbiased error estimates.
- Larger test sets yield more accurate error estimates.
- Leave-one-out cross-validation seems to be an optimal compromise, but might be computationally infeasible.
- 10-fold cross-validation 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 simple classifiers or smaller feature sets are needed.
- Confusion matrices allow a detailed look at the per class classification

Reject and ROC Analysis

- Reject Types
- Reject Curves
- Performance Measures
- Varying Costs and Priors
- ROC Analysis



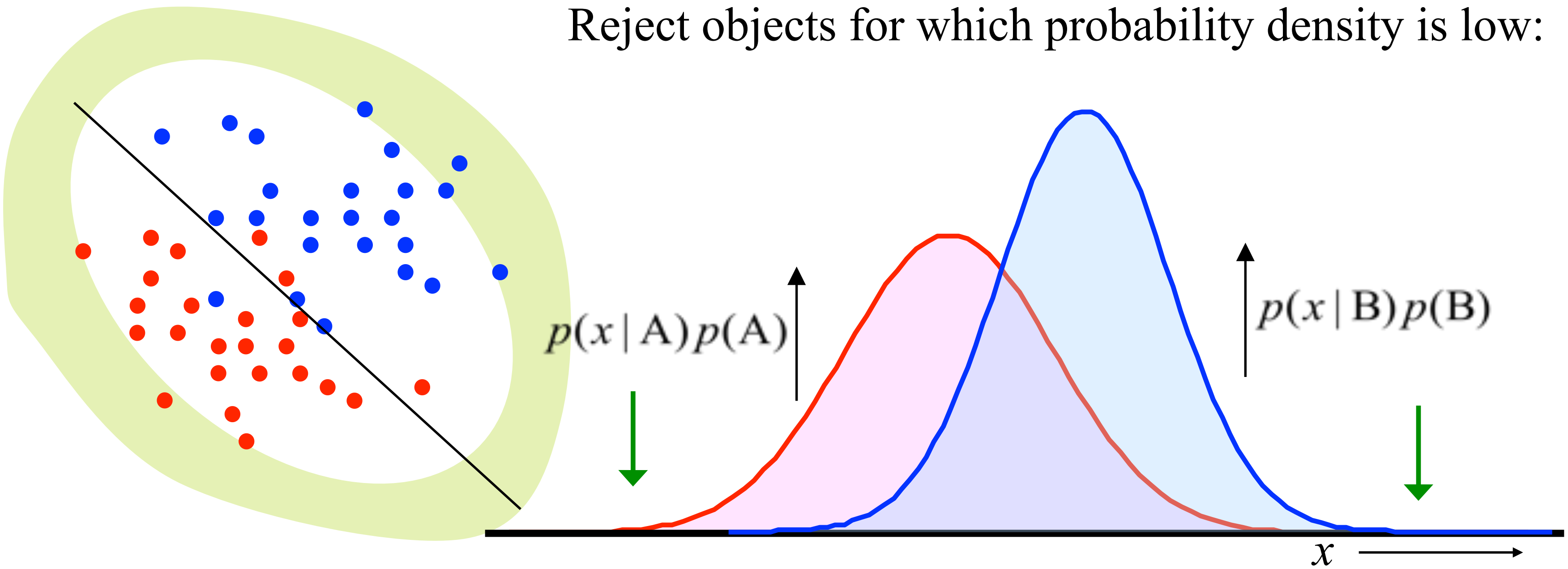




Outlier Reject

`rejectc`

Reject objects for which probability density is low:

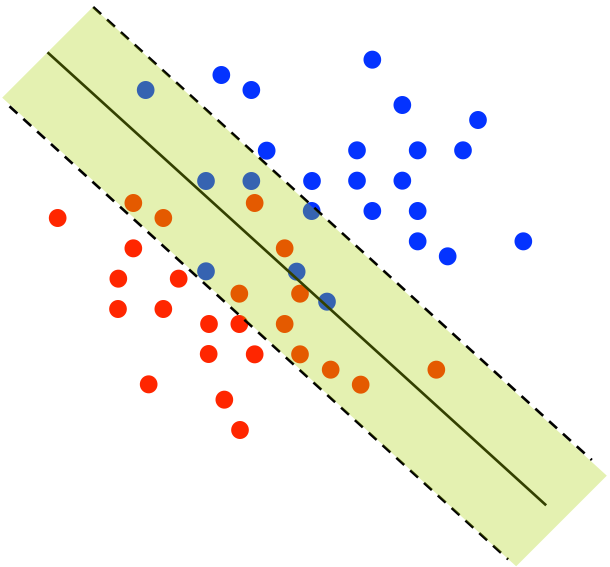


$$P(x) = P(x \mid \mathrm{A}) P(\mathrm{A}) + P(x \mid \mathrm{B}) P(\mathrm{B}) \approx 0$$

Note: in these area the posterior probabilities might be high! $\quad | S(x) | >> 0$

Ambiguity Reject

rejectc



Reject objects for which classification is unsure:
about equal posterior probabilities:

$$P(\mathrm{A} \mid x) \approx P(\mathrm{B} \mid x)$$

$$\frac{P(x|\mathrm{A})P(\mathrm{A})}{P(x)} \approx \frac{P(x|\mathrm{B})P(\mathrm{B})}{P(x)}$$

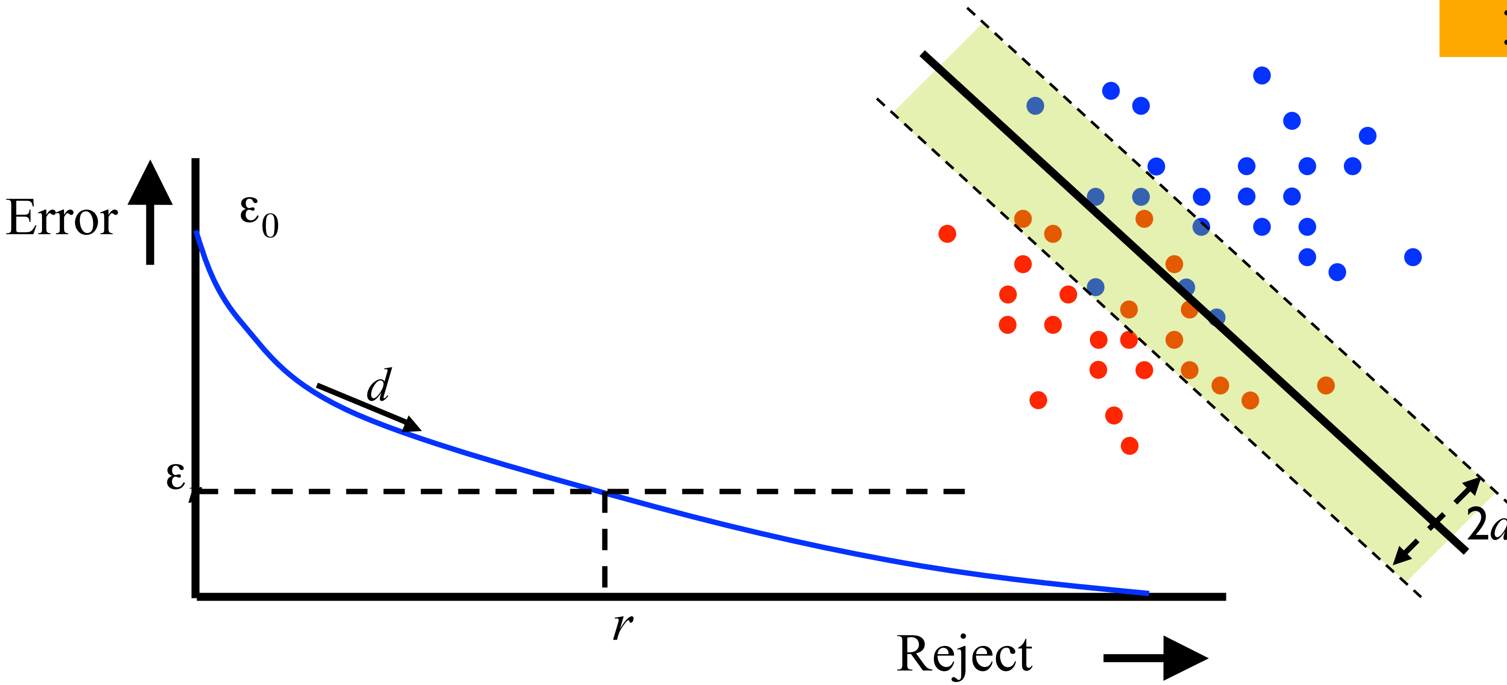
$$P(x \mid \mathrm{A})P(\mathrm{A}) - P(x \mid \mathrm{B})P(\mathrm{B}) \approx 0$$

$$S(x) \approx 0$$

Reject Curve

reject

plote



The classification error ε_0 can be reduced to ε_r by rejecting a fraction r of the objects.

How much to reject?

reject plote

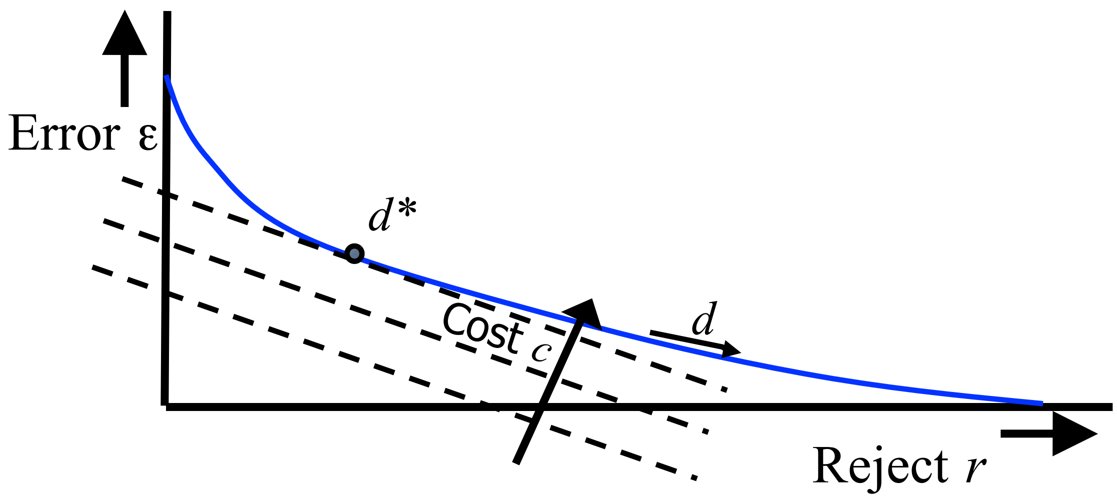
Compare the cost of a rejected object, c_r, with the cost of a classification error, c_ε:

$$c = c_r P(reject) + c_\varepsilon P(error)$$

$$c = c_r r + c_\varepsilon \varepsilon$$

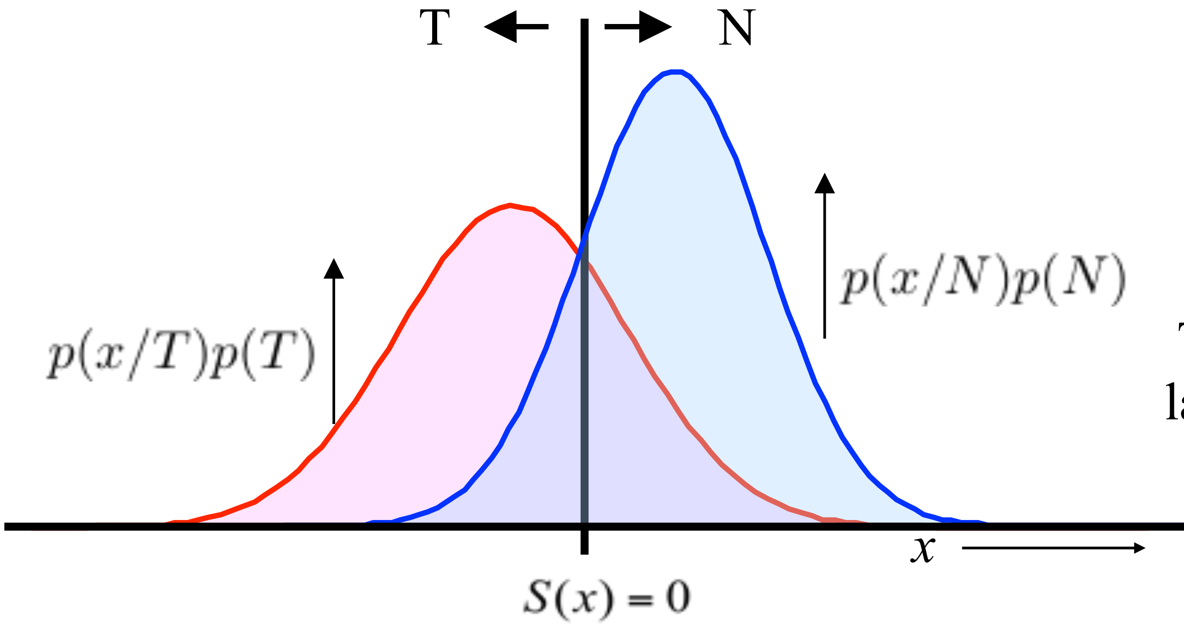
For given total cost c this is a linear function in the (r,ε) space.

Shift it until a possible operating point is reached.



Error / Performance Measures

Given a trained classifier and a test set:



		classified as T	classified as N	
True labels	T	TP	FN	N_T
	N	FP	TN	N_N

Error: $$\frac{FP + FN}{N}$$

Sensitivity: $$\frac{TP}{N_T} = TP_r$$

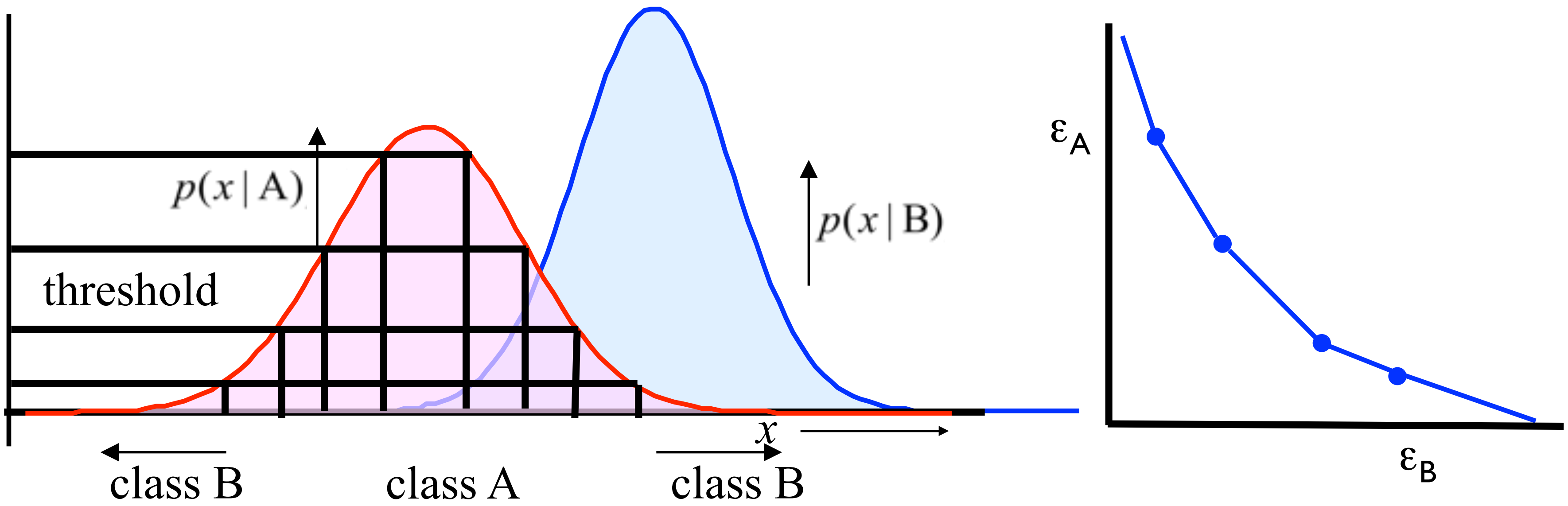
Specificity: $$\frac{TN}{FP + TN} = 1 - FP_r$$

False Discovery Rate (FDR): $$\frac{FP}{FP + TP}$$

Error / Performance Measures

- **Error**: probability of erroneous classifications
- **Performance**: 1 – error
- **Sensitivity** of a target class (e.g. diseased patients): performance for objects from that target class.
- **Specificity**: performance for all objects outside the target class.
- **Precision** of a target class: fraction of correct objects among all objects assigned to that class.
- **Recall**: fraction of correctly classified objects. This is identical to the performance. It is also identical to the sensitivity when related to a particular class.

ROC: thresholding

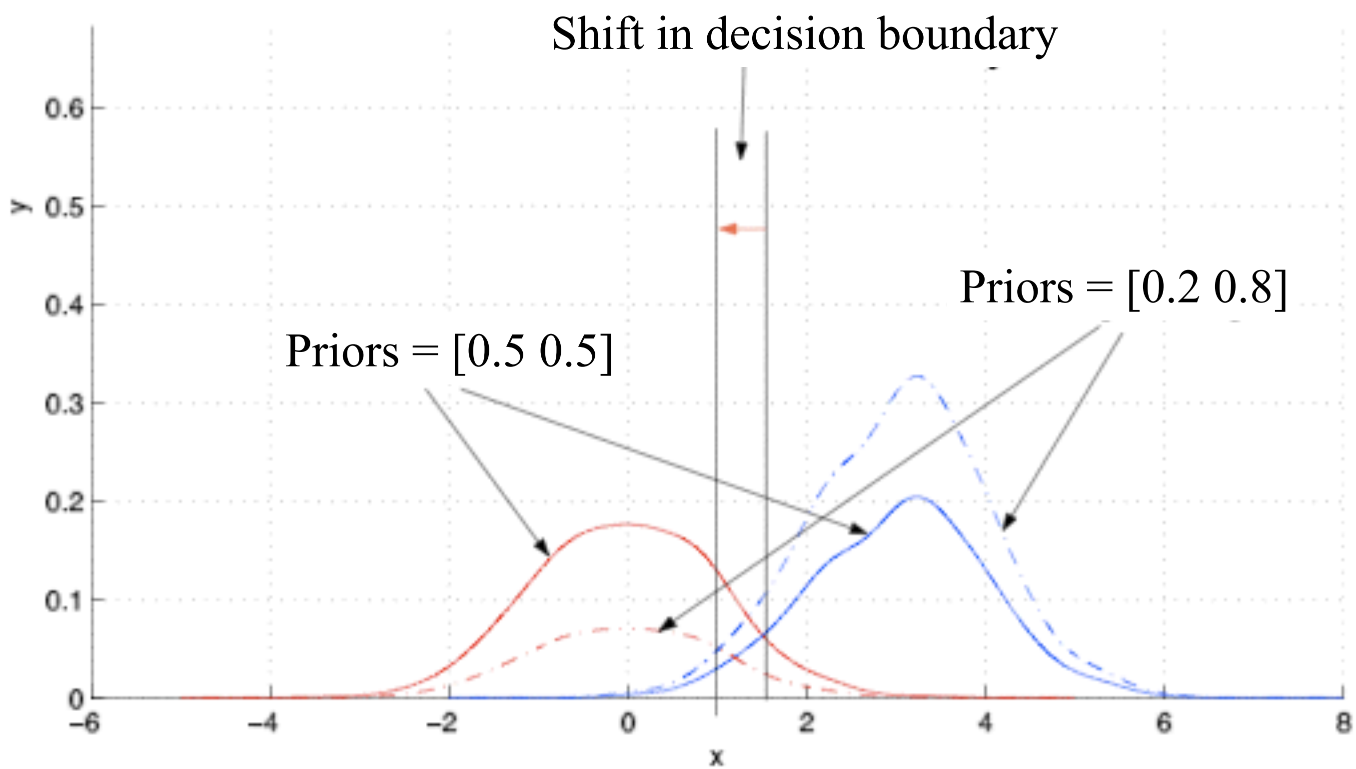


A is the class of interest.

A range of values for the threshold is chosen.

At each value the corresponding classifier is evaluated, and the error is computed.

ROC: weighting



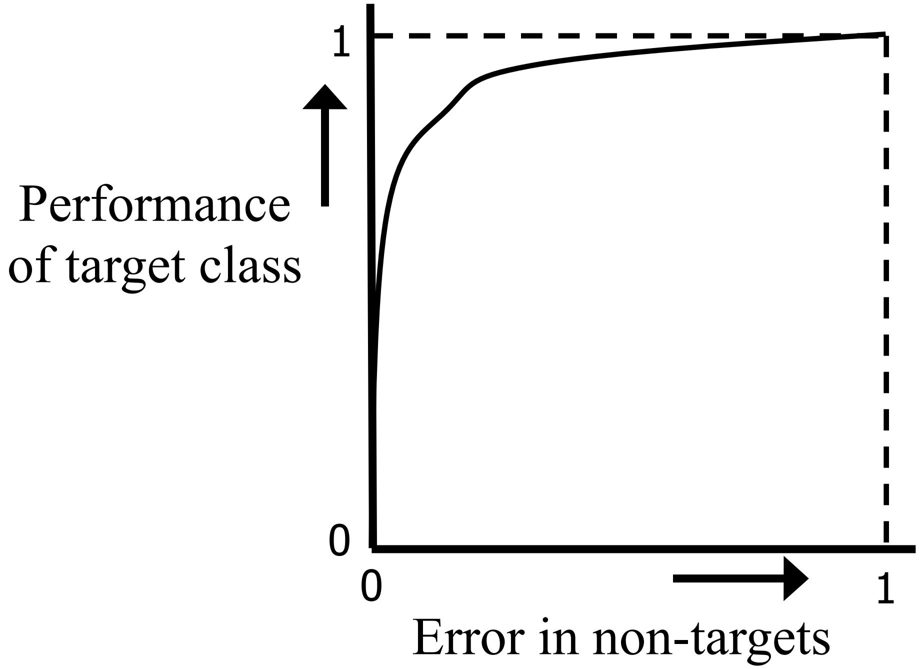
ROC is independent of class prior probabilities

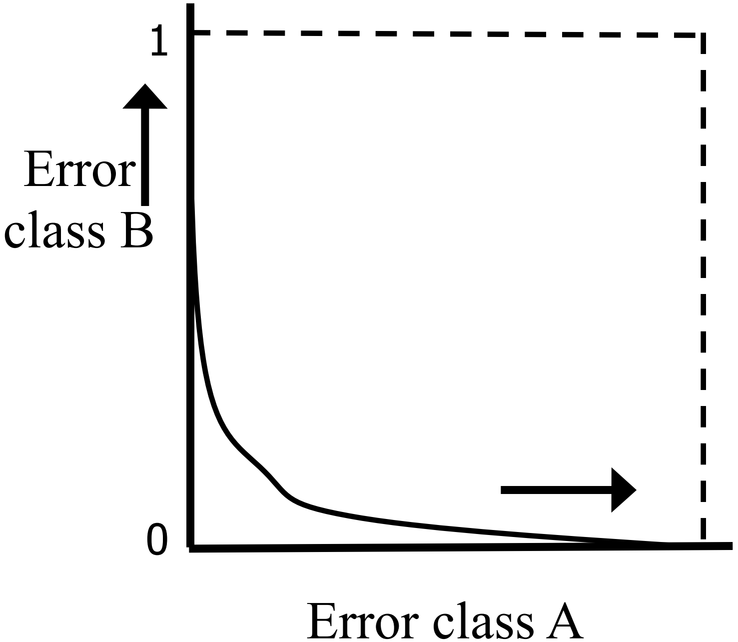
Change of prior (weight) is analogous to the shift of the decision boundary

ROC Analysis

roc | plote

ROC: Receiver-Operator Characteristic (from communication theory)

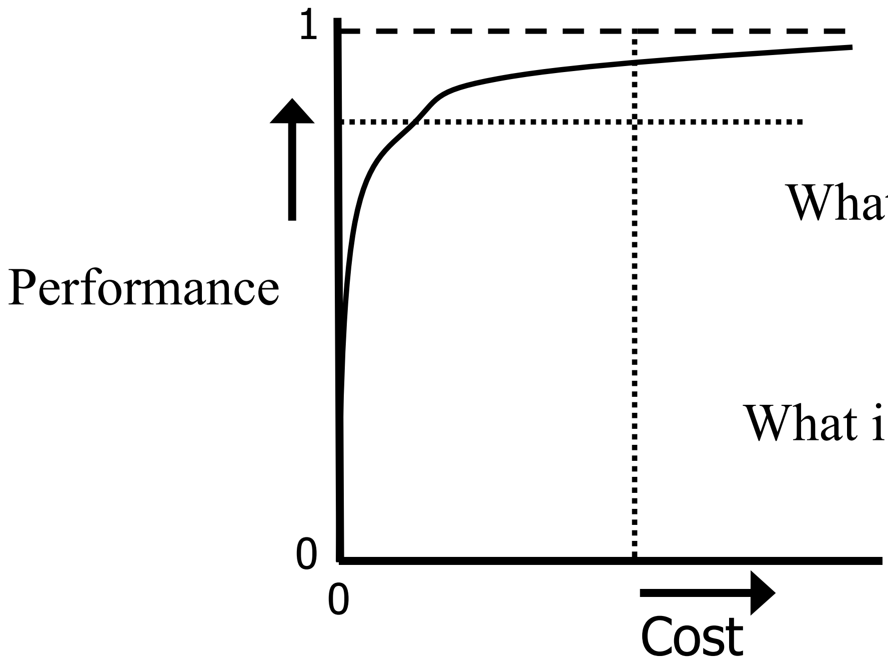




Medical diagnostics
Database retrieval

2-class pattern recognition

When are ROC Curves Useful? (1)



What is the cost of a preset performance?

What is the performance for a given cost?

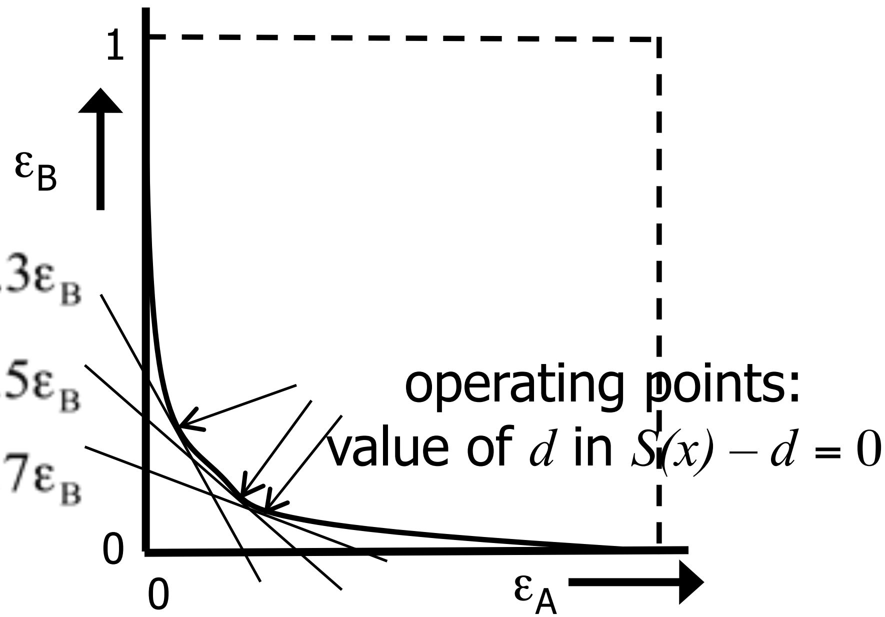
Trade-off between cost and performance.

When are ROC Curves Useful? (2)

$$\varepsilon_{total} = 0.7\varepsilon_A + 0.3\varepsilon_B$$

$$\varepsilon_{total} = 0.5\varepsilon_A + 0.5\varepsilon_B$$

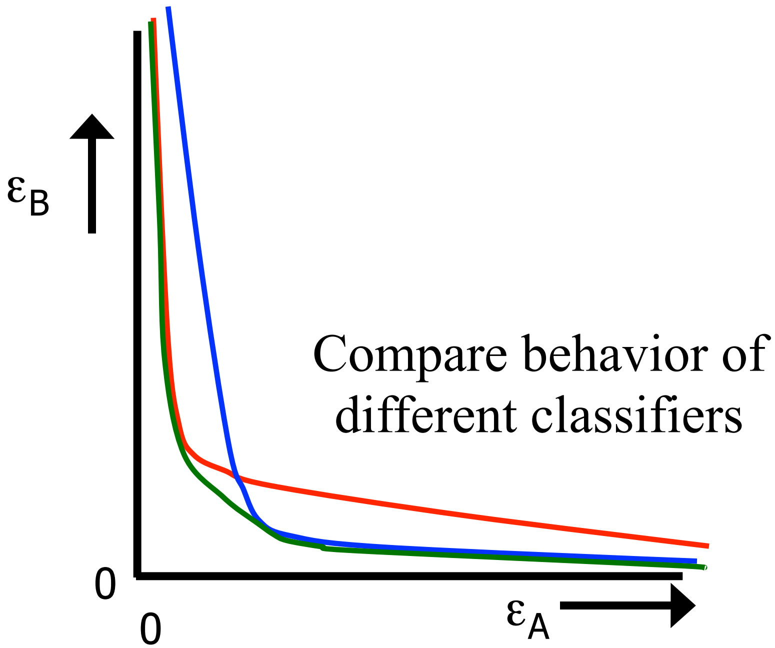
$$\varepsilon_{total} = 0.3\varepsilon_A + 0.7\varepsilon_B$$

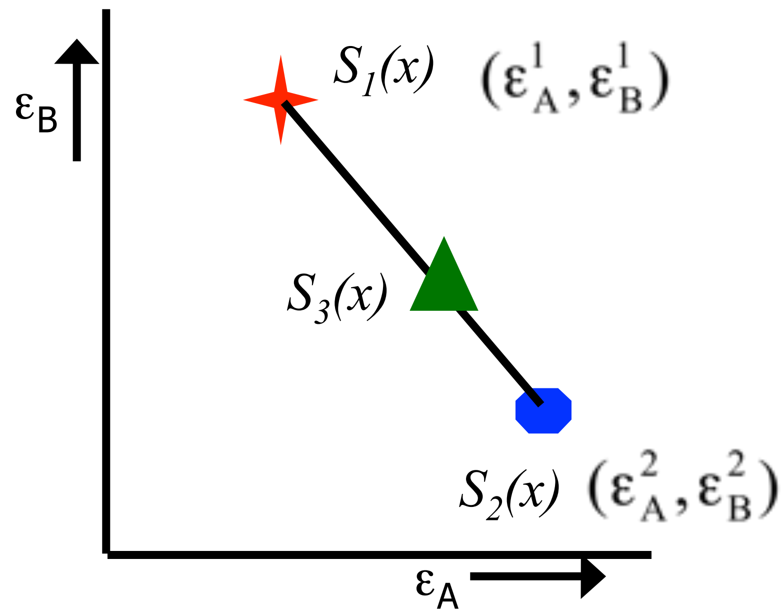


$$\varepsilon_{total} = P(\mathrm{A})\varepsilon_A + P(\mathrm{B})\varepsilon_B$$

Study of the effect of changing priors

When are ROC Curves Useful? (3)



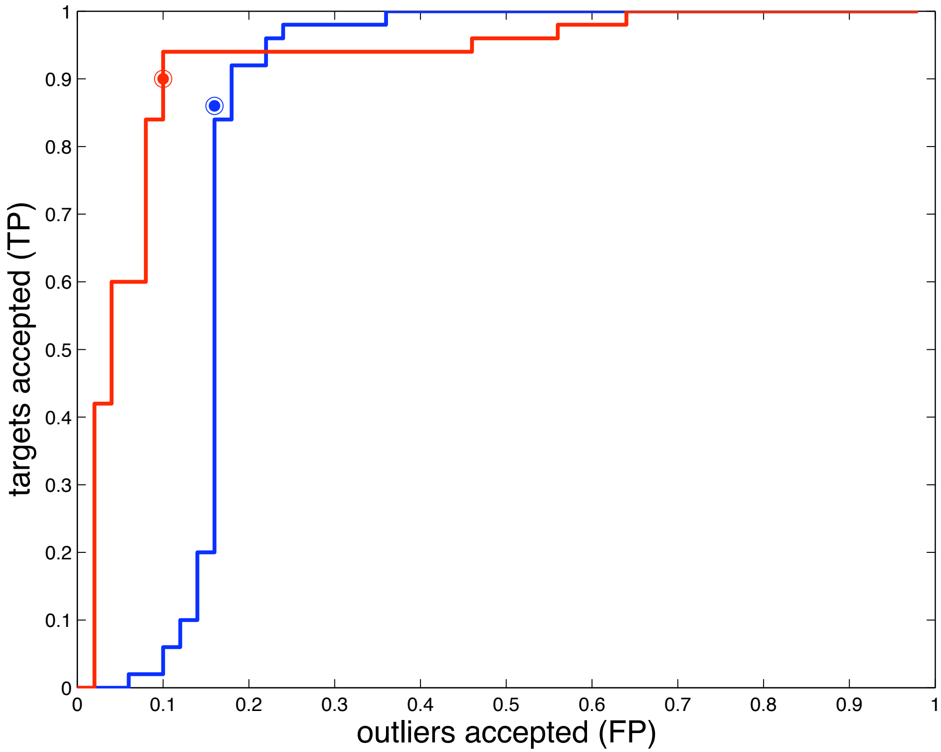


Combine Classifiers

Any 'virtual' classifier between $S_1(x)$ and $S_2(x)$ in the $(\varepsilon_A, \varepsilon_B)$ space can be realized by using at random α times $S_1(x)$ and $(1-\alpha)$ times $S_2(x)$.

$$\varepsilon_A = \alpha\varepsilon_A^1 + (1-\alpha)\varepsilon_A^2 \qquad \varepsilon_B = \alpha\varepsilon_B^1 + (1-\alpha)\varepsilon_B^2$$

Area under the ROC curve



- Comparing ROC curves is not so simple: for each threshold it is different
- An well-known overall measure is the AUC: Area under the ROC curve
- Integrate uniformly over all thresholds
- Value should be between 0.5 and 1
- Insensitive to class imbalance in the test set

Area under the ROC curve

- The ROC curve shows the true positive fraction as function of the false positive fraction for varying threshold
- Independent of class priors and misclassification costs

- The AUC is identical to the chance that a random '+'-class object is ranked higher than a random '-'-class object

$$A_z = Pr\left(f(\mathbf{x}_+) > f(\mathbf{x}_-)\right)$$

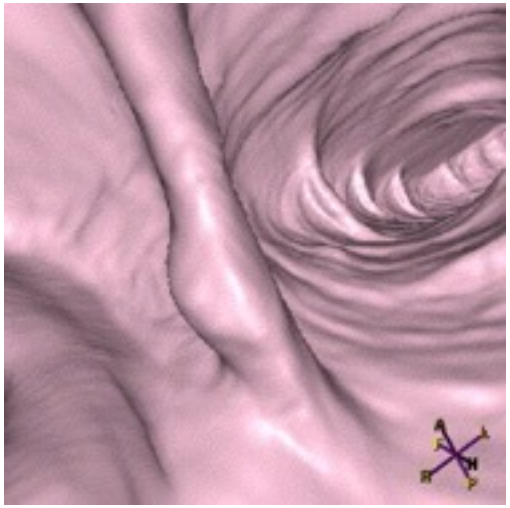
This can be estimated from a test set:

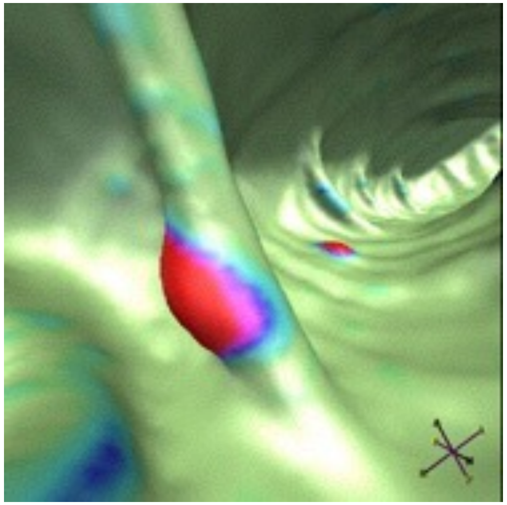
$$\hat{A}_z = \frac{1}{N^+ N^-} \sum_{k^+=1}^{N^+} \sum_{k^-=1}^{N^-} \mathcal{I}(f(\mathbf{x}_{k^+}) > f(\mathbf{x}_{k^-}))$$

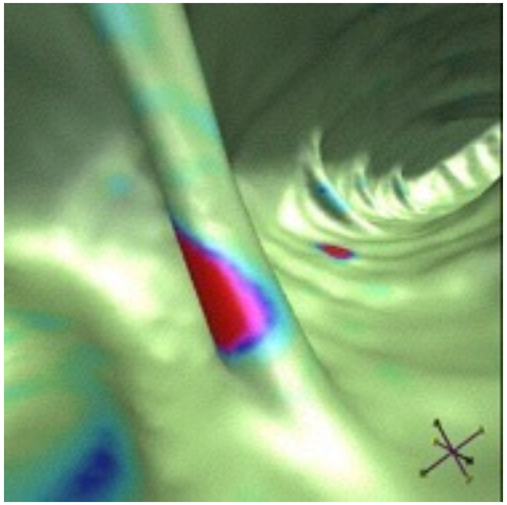
This is called the Wilcoxon-Mann-Whitney statistics

Example: Polyp Detection in CT Colonography

Borrowed from Vincent van Ravesteijn



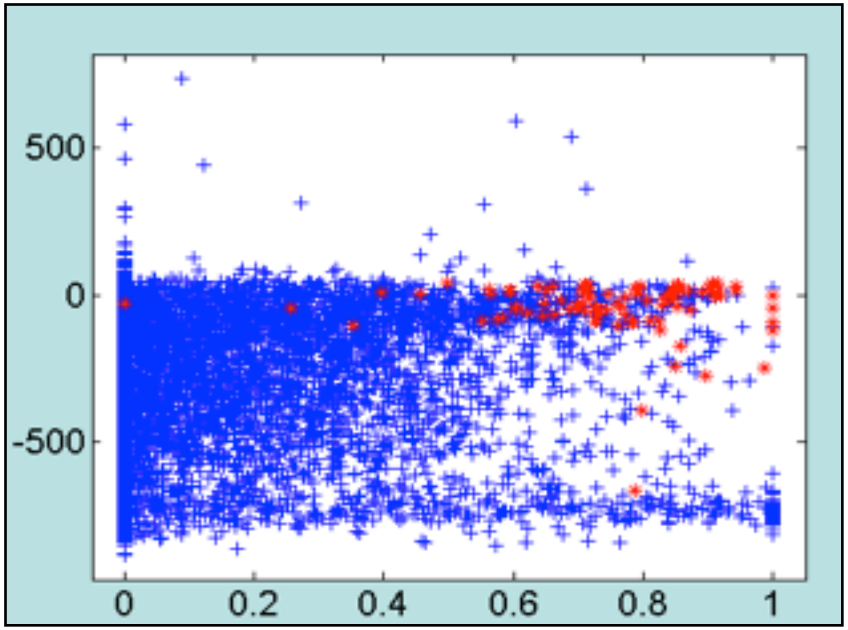


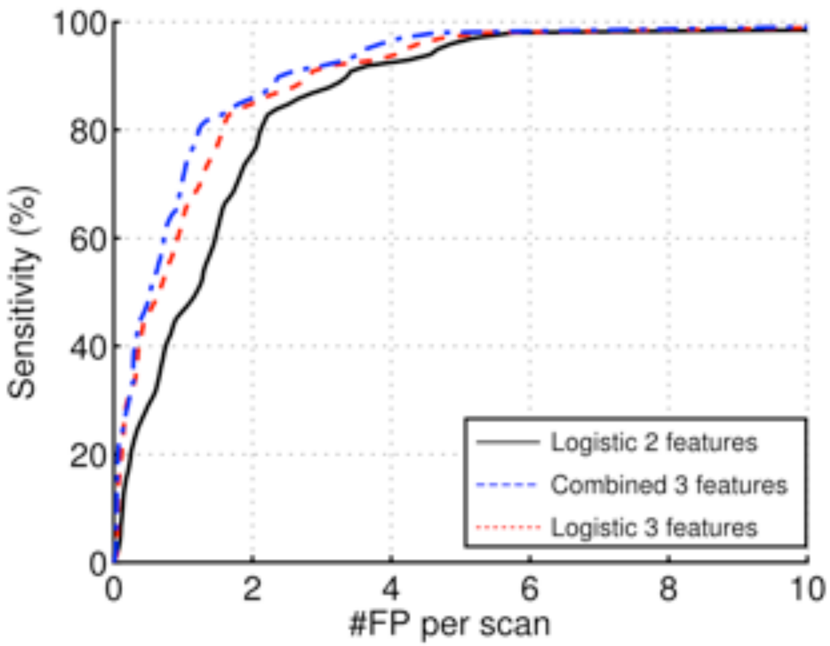


Polyp Detection

2D Feature Space

Just 0.6% of candidates are polyps





86 patients / 172 scans with
59 polyps ≥ 6 mm

Summary Reject and ROC

- Reject for solving ambiguity: reject objects close to the decision boundary → lower costs.
- Reject option for protection against outliers.
- ROC analysis for performance – cost trade-off.
- ROC analysis in case of unknown or varying priors.
- ROC analysis for comparing / combining classifiers.
- AUC to compare classifiers