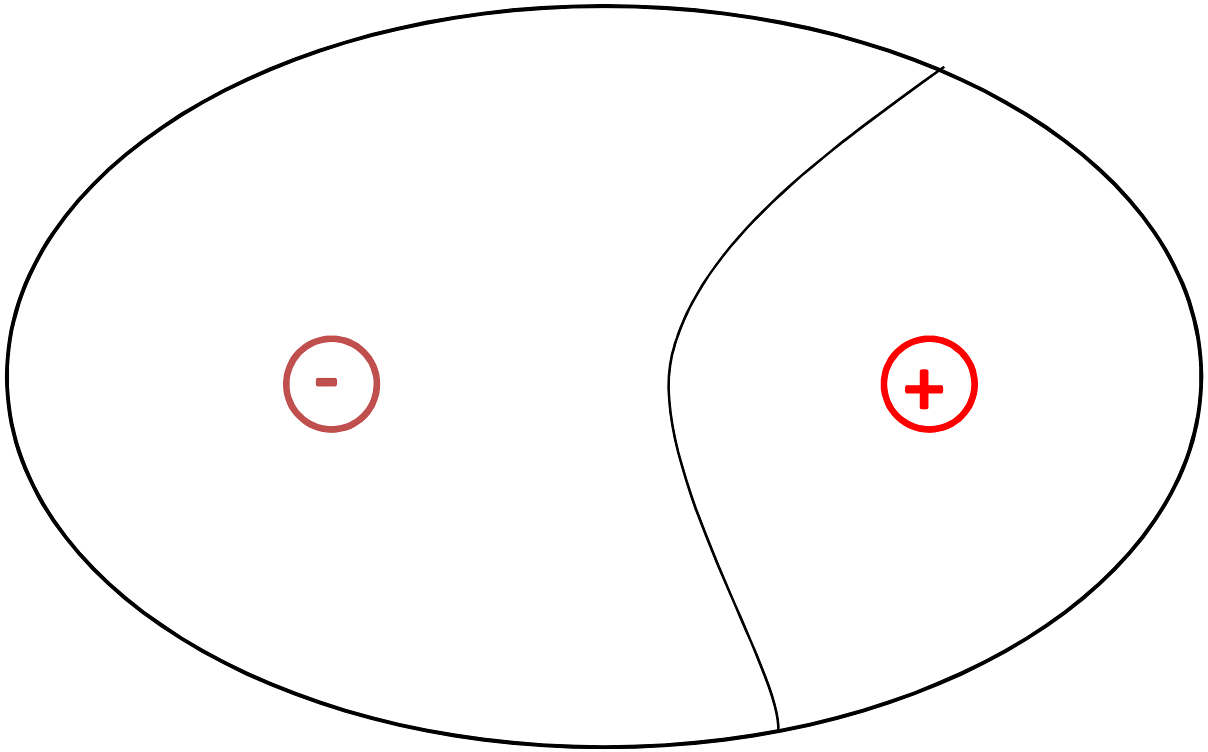
Supervised Learning 2

Blaise Hanczar (812-E)

Outline

- Performance estimation
- Classification with reject option
- Multi-class classification

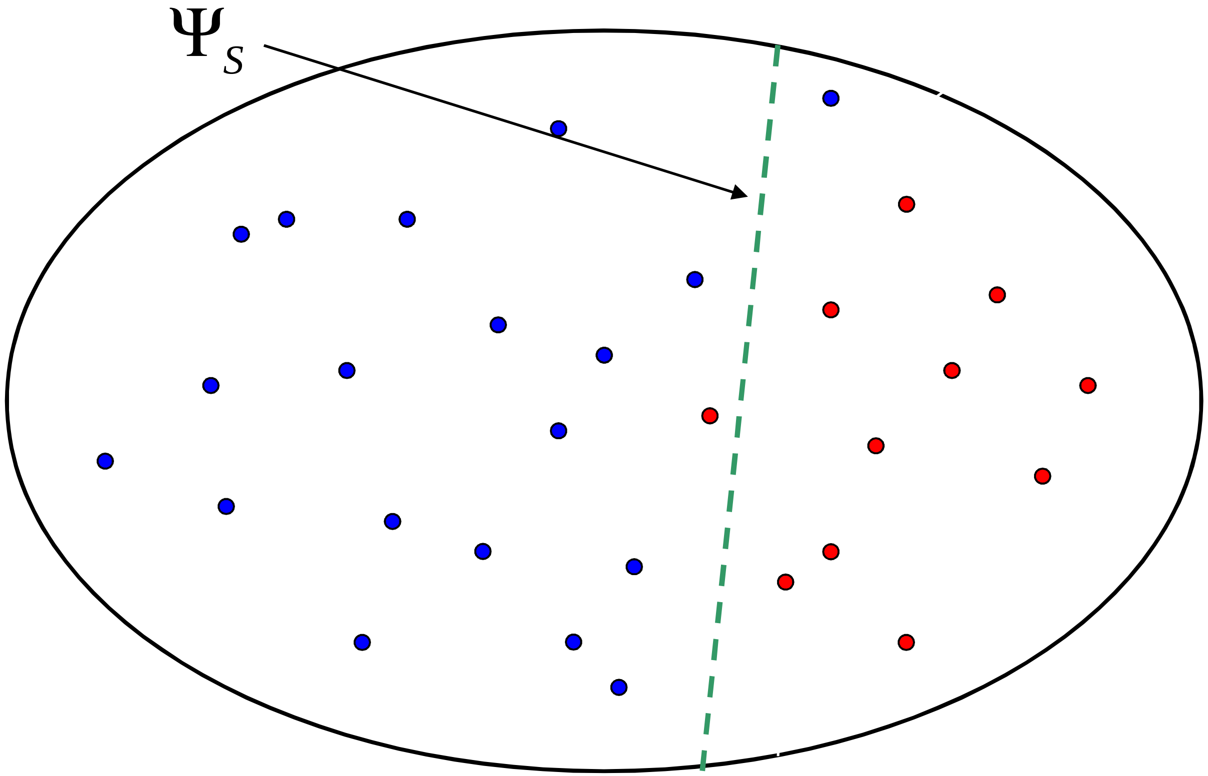
Estimation d'erreur



Distribution des classes

- **Construction d'un classeur**

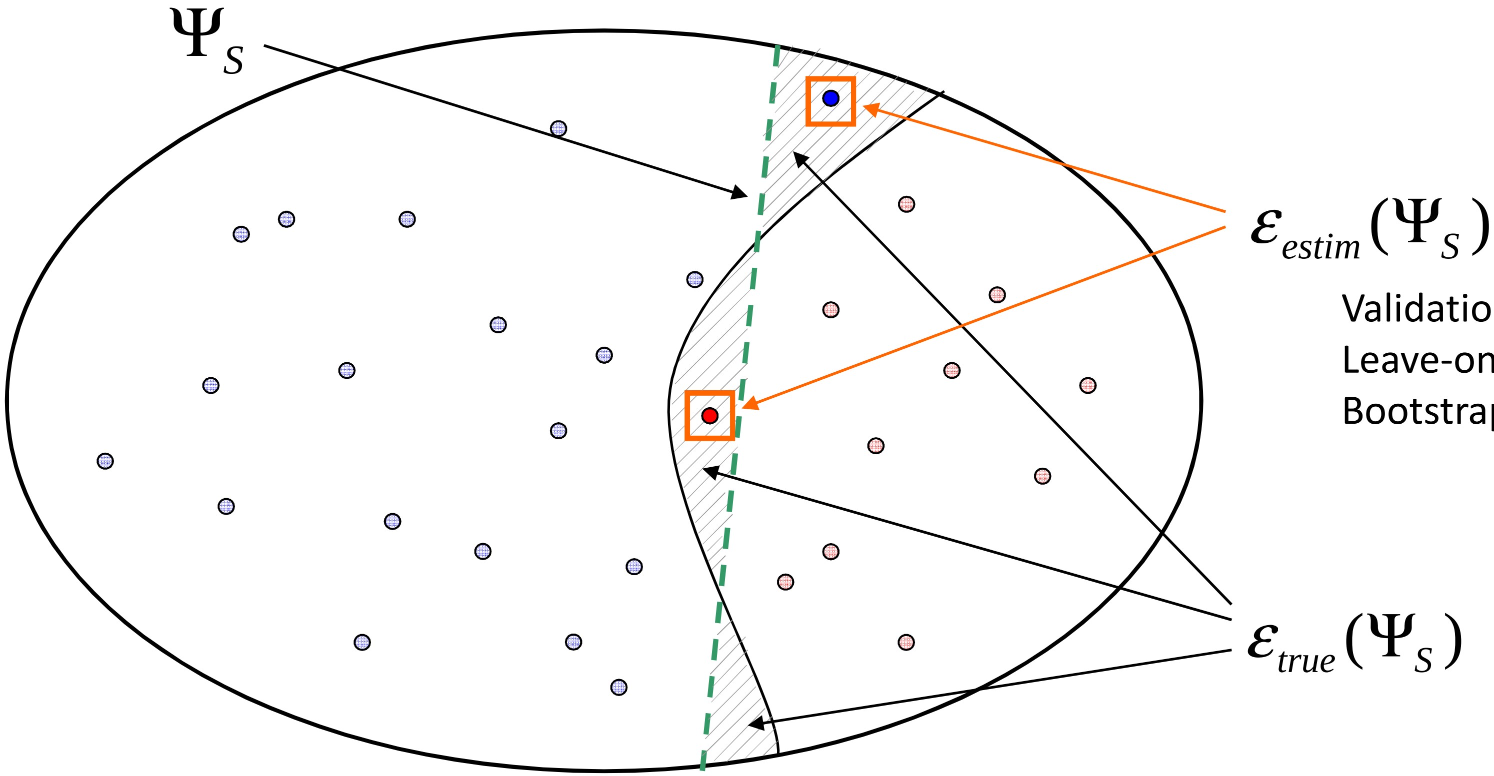
Estimation d'erreur



- Base d'apprentissage $S=\{X_i,Y_i\}$ de N exemples
- Construction d'un classeur Ψ_S

➢ **Calculer le taux d'erreur**

Estimation d'erreur



Validation croisée
Leave-one-out
Bootstrap

ε_{estim} est il une estimation fiable de ε_{true} dans les problèmes à grande dimension et peu d'exemples?

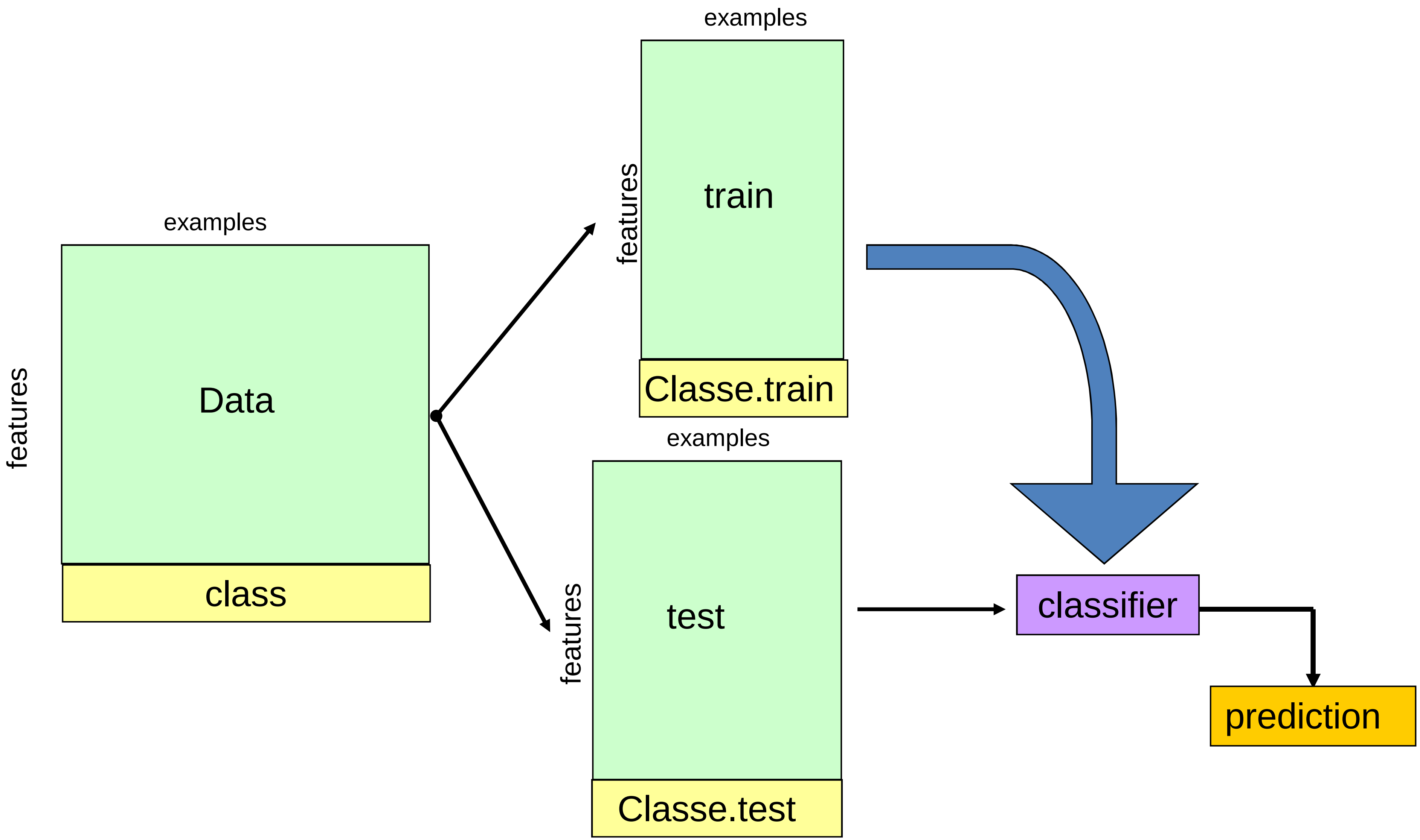
[Braga-Neto, 06]

Training and test datatset

- Training datatset
  - The classes are known
  - Use for model learning

- Test datatset
  - The classes are unknown
  - Not use in the model learning
  - Use to estimate the performance of the classifier

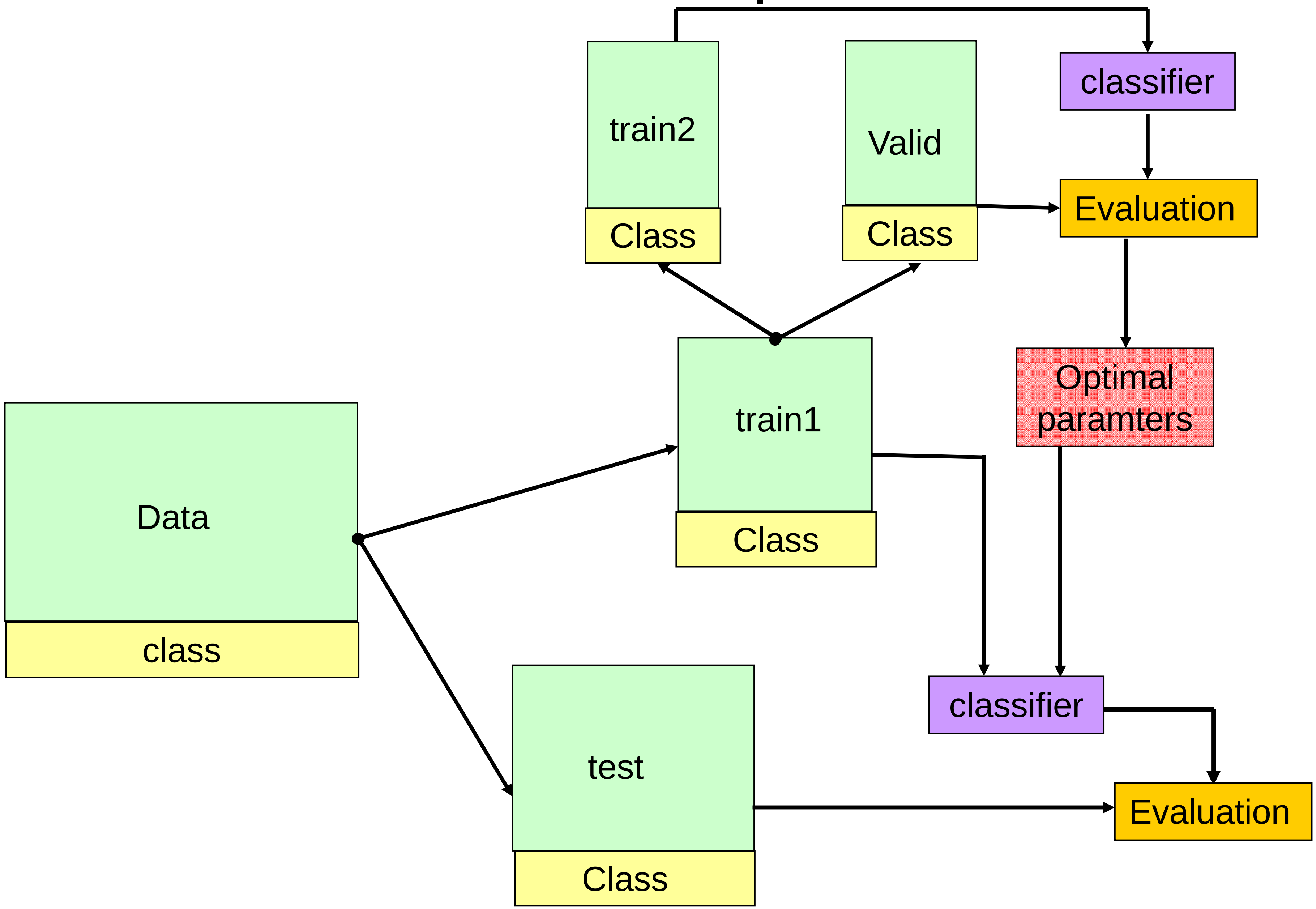
Data decomposition



Data decompositon

- Training set: 50%
  - Learning the model
- Validation set: 25%
  - Fitting the paramaters
- Test set: 25%
  - Classifier evaluation

Data decomposition



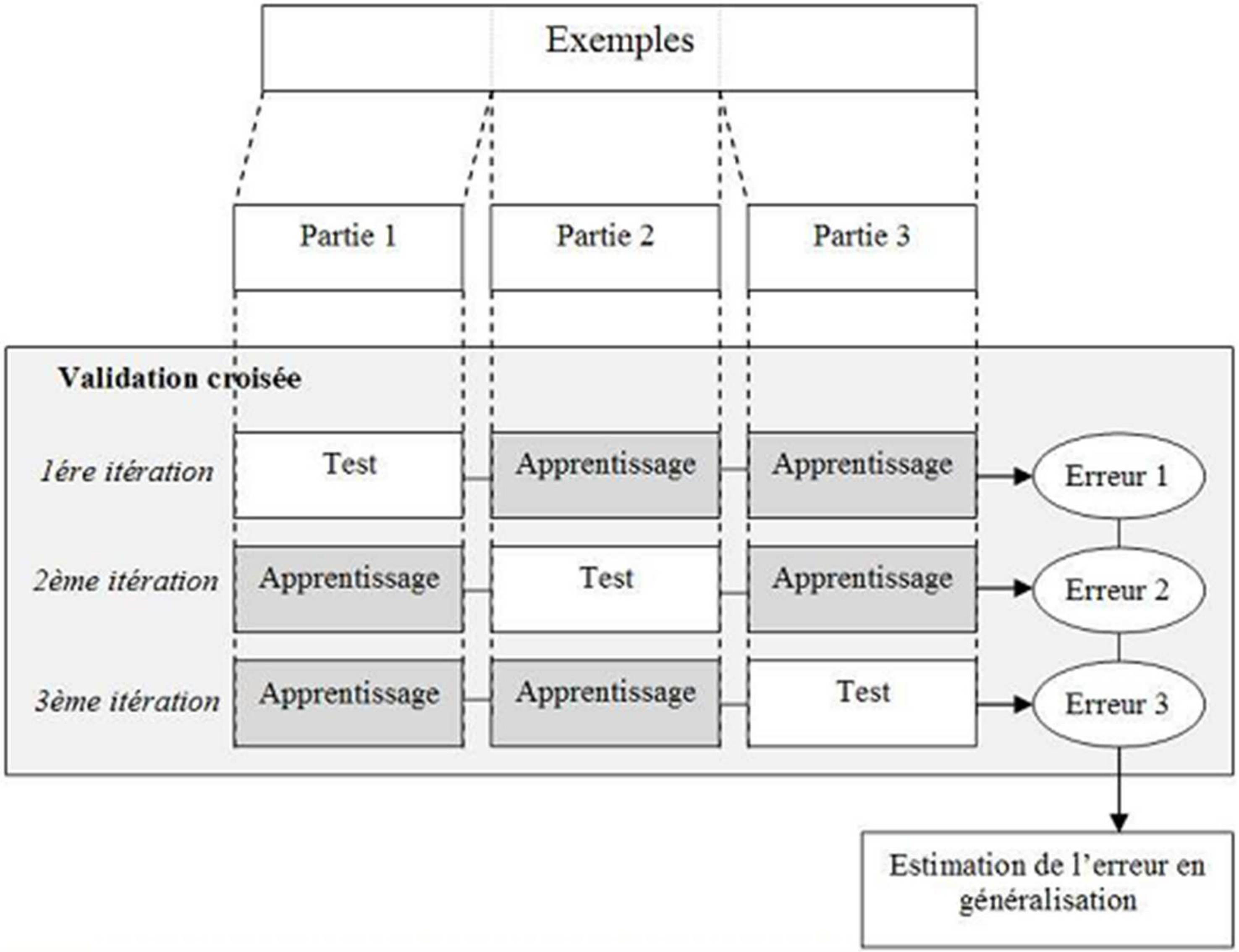
Cross validation

- Divide the example set into K distinct sets of the same size (ex: K=10)
- For i = 1:K
    - Training set: K-1 sets

      All sets except the i-th set
    - Test set: i-th set
    - Err(i) = compute the error
- Error = $1/K \sum_i err(i)$

Case K=N: leave-one-out

Cross Validation



Bootstrap

For = 1:B

- Generate a bootstrap sample of examples
- Training set: examples in the bootstrap sample
- Test set: examples not in the bootstrap sample
- Err(i) = compute the error

Error_bootstrap = $1/K \sum_i err(i)$

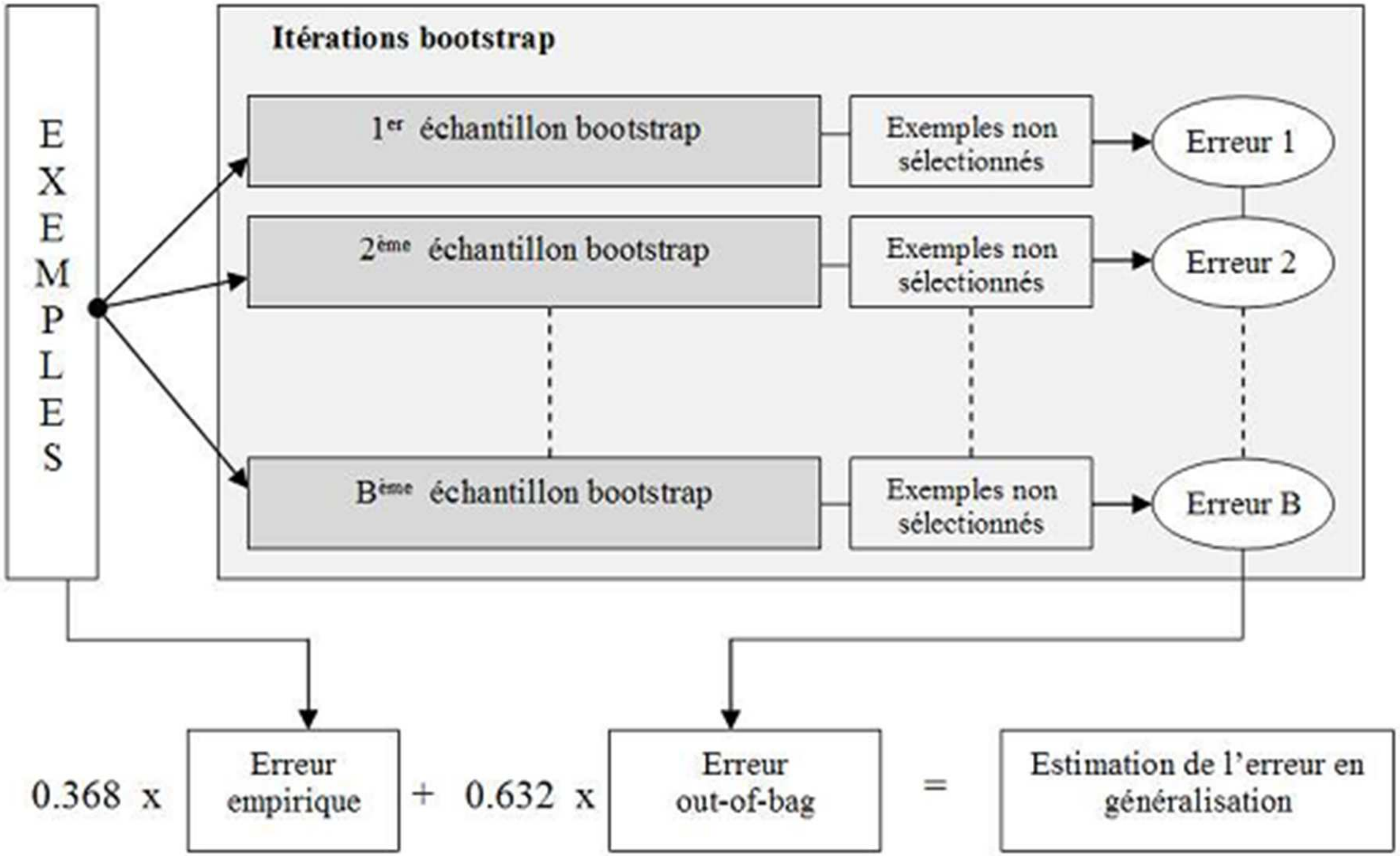
Boostrap 632:

0.368 X training error + 0.632 bootstrap error

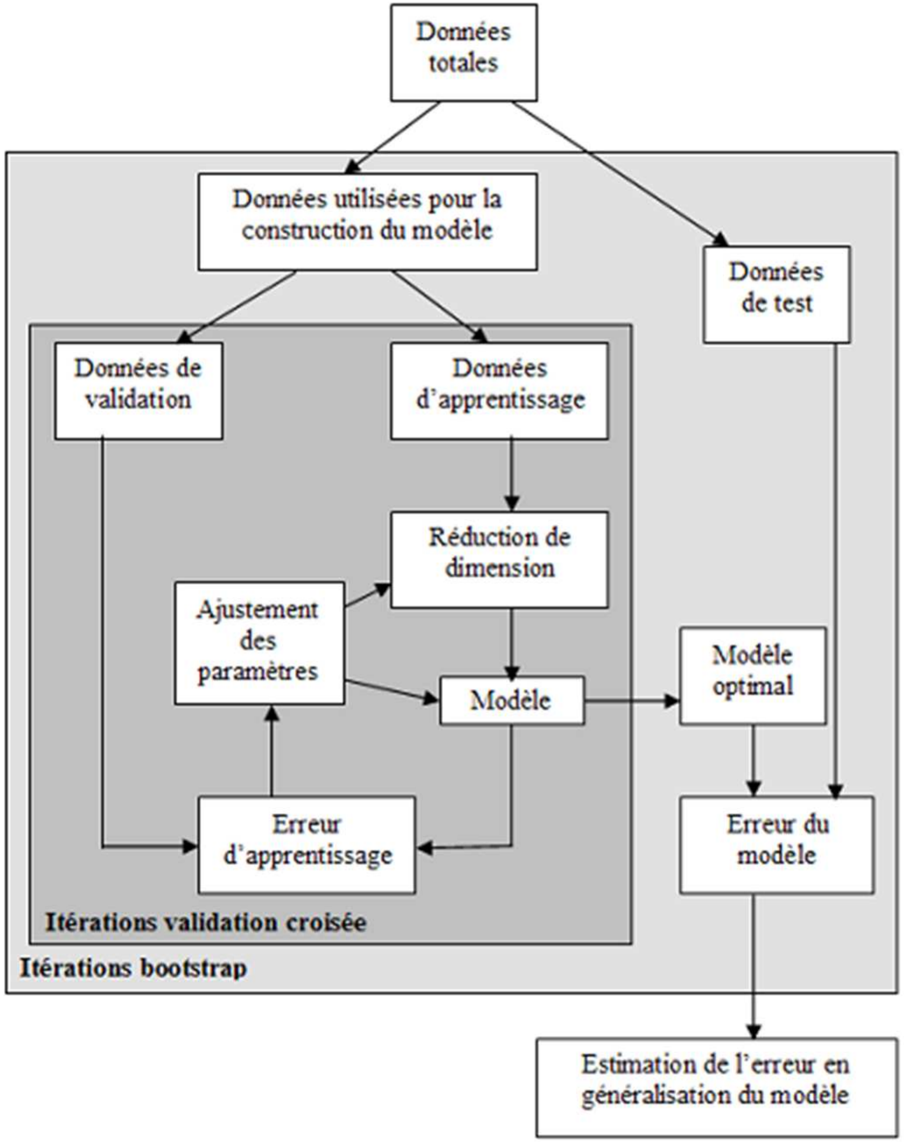
Echantillion Bootstrap

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	7	8	5	6	4	2	7	1
Training set 3	3	6	2	7	5	6	2	2
Training set 4	4	5	1	4	6	4	3	8

Bootstrap 632

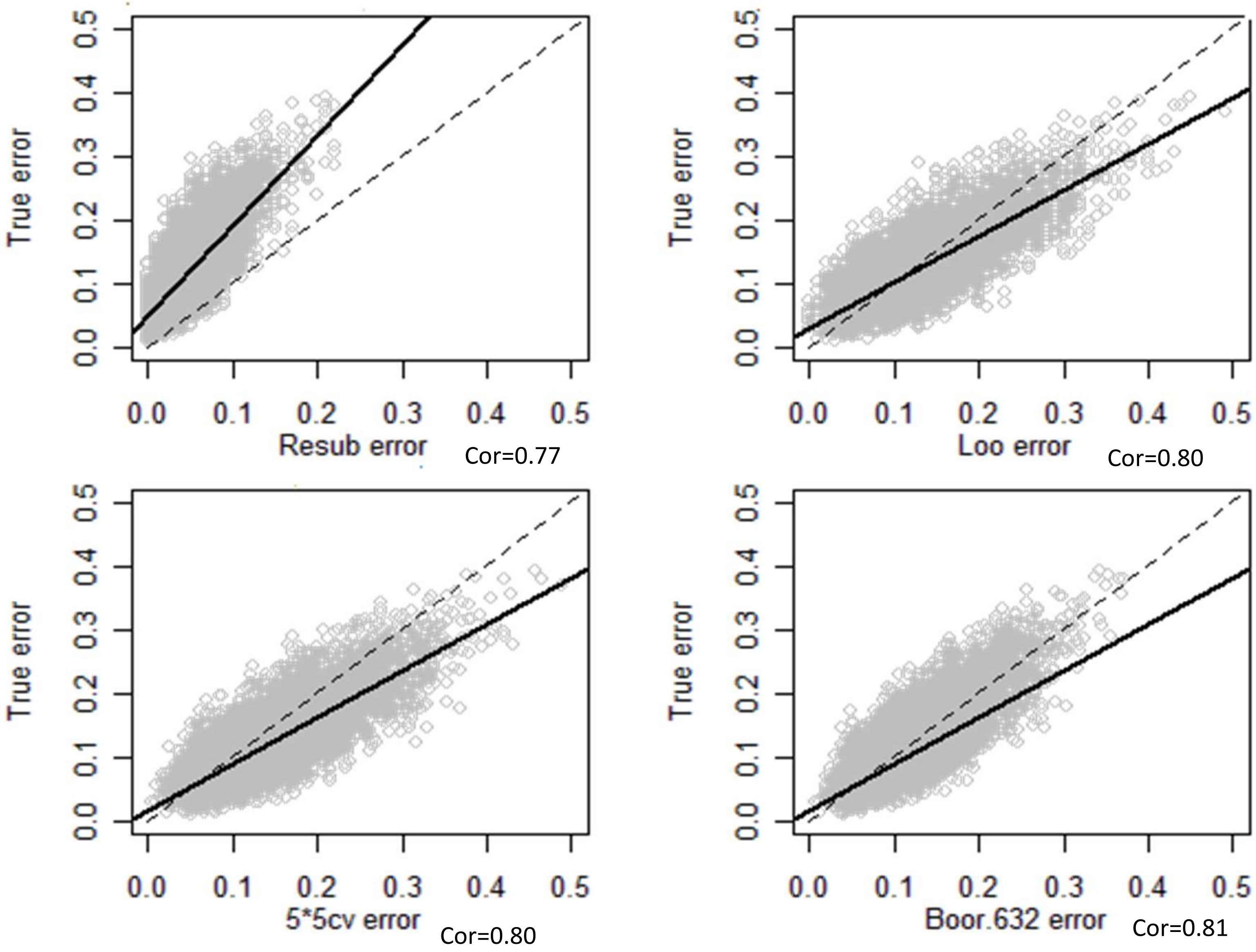


Example of estimation design

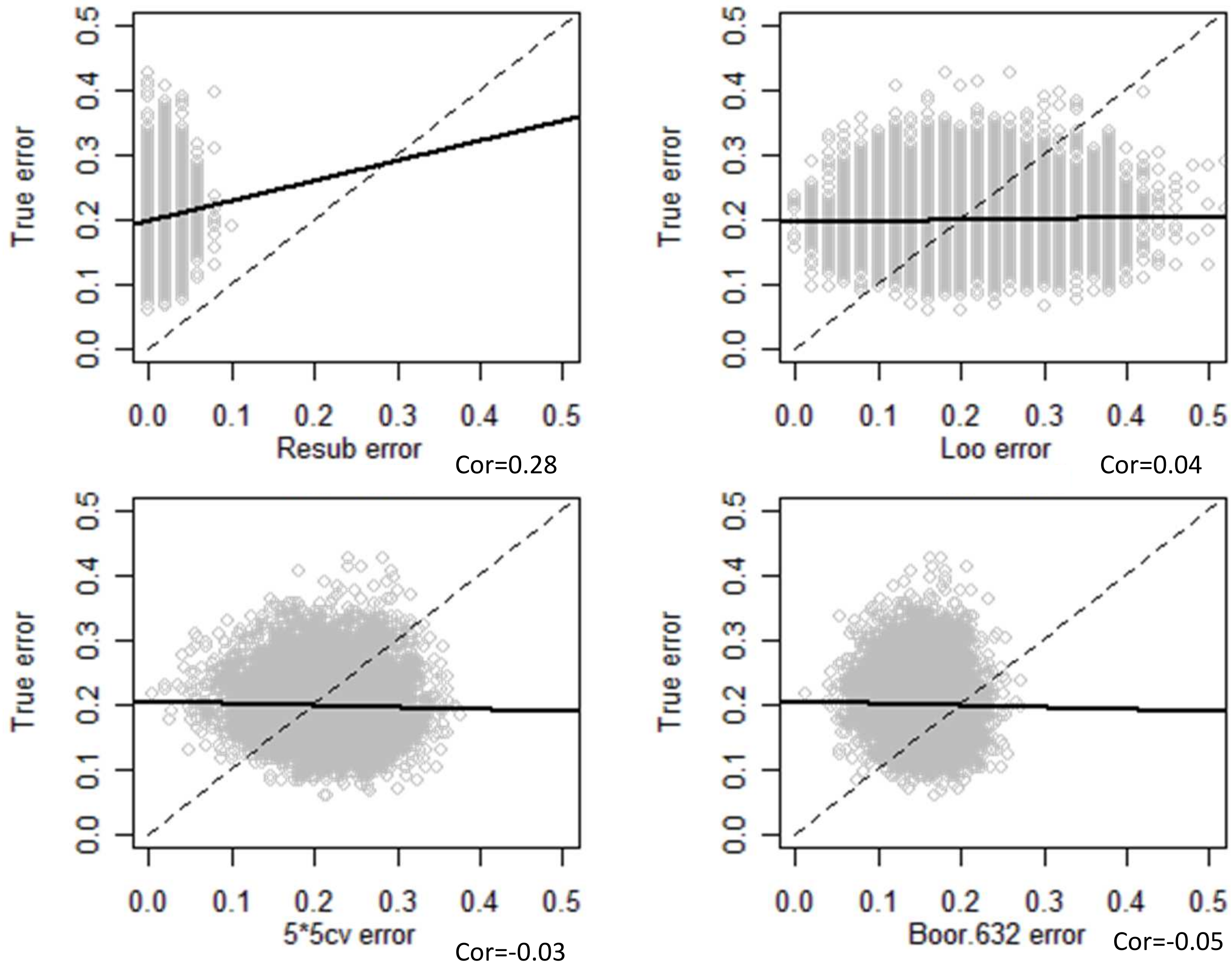


Error estimation (high sample size)

N=1000



Error estimation (small sample size)



N=50

Confiance Interval

- The estimated error is a statitics, we can compute a confidence interval.

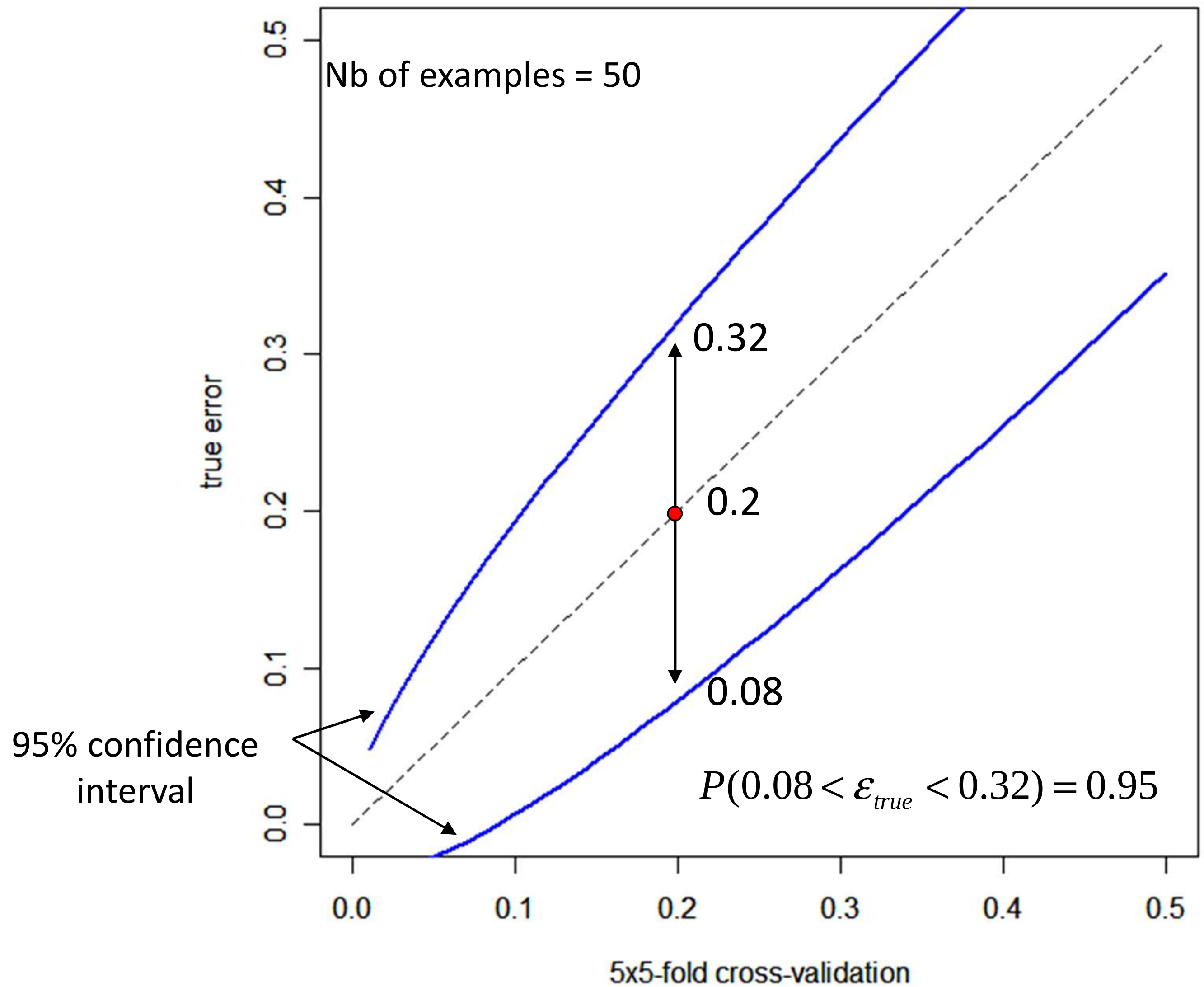
- We need to compute the variance of the error estimator
  - Consider that the erreur estimation follows a binomial distribution

$$Var_{\varepsilon} = \frac{\varepsilon(1-\varepsilon)}{N})$$

- The 95% confidence interval is given by

$$\varepsilon_{true} \in \left[\varepsilon - 1.96\sqrt{Var_{\varepsilon}}; \varepsilon + 1.96\sqrt{Var_{\varepsilon}}\right]$$

Confidence intervals

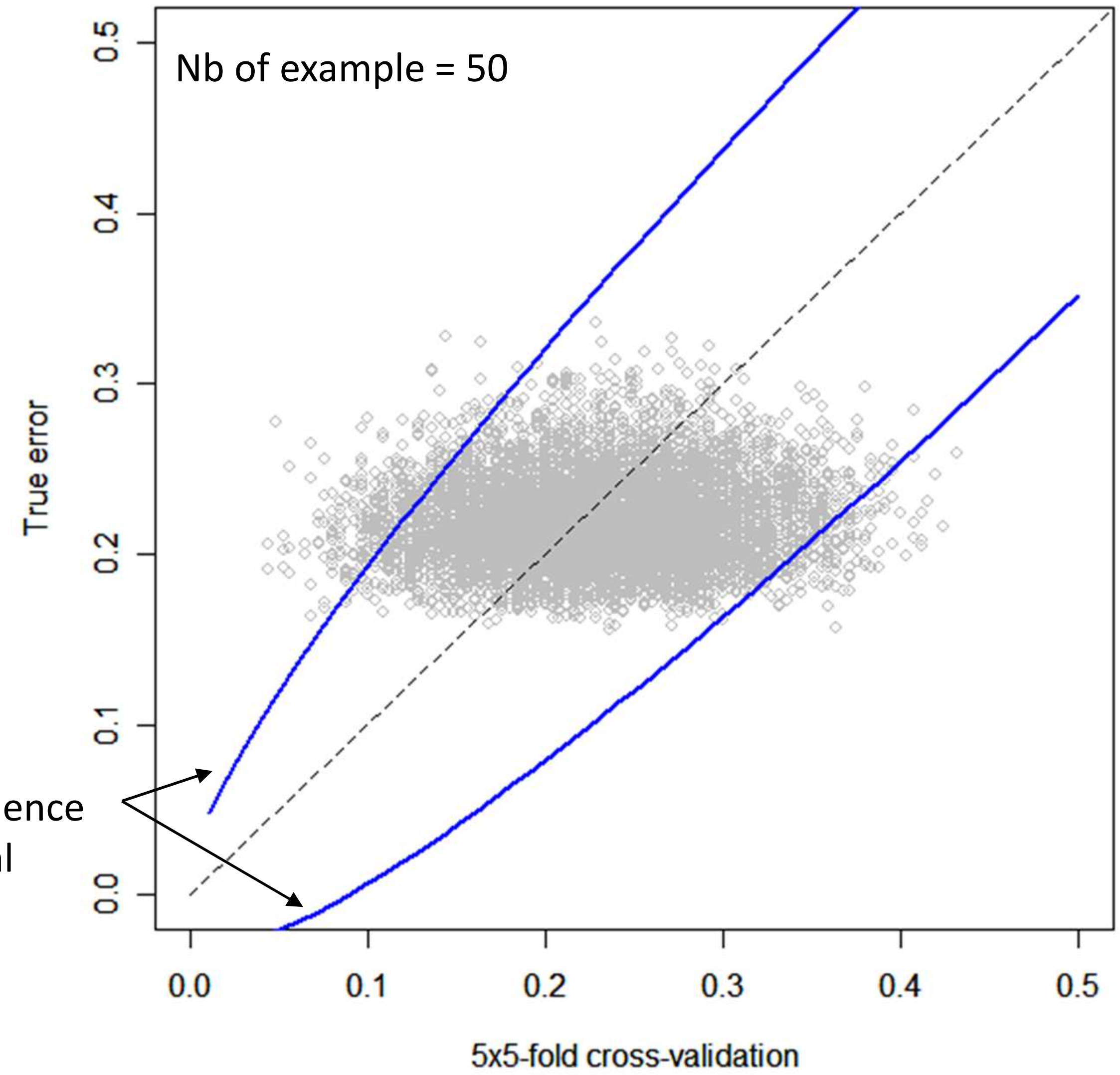


$P(0.08 < \varepsilon_{true} < 0.32) = 0.95$

Error $\in [0.08, 0.32]$

$$\varepsilon_{true} \rightarrow N(\varepsilon_{estim}, \frac{\varepsilon(1-\varepsilon)}{N})$$

Confidence intervals



95% confidence interval

95.85% of the points are in the confidence interval

Principle of reject option

Pretty good performance of classification may be obtain with regular classifier… but sometimes it is not good enough for practical use (ex: medical application)

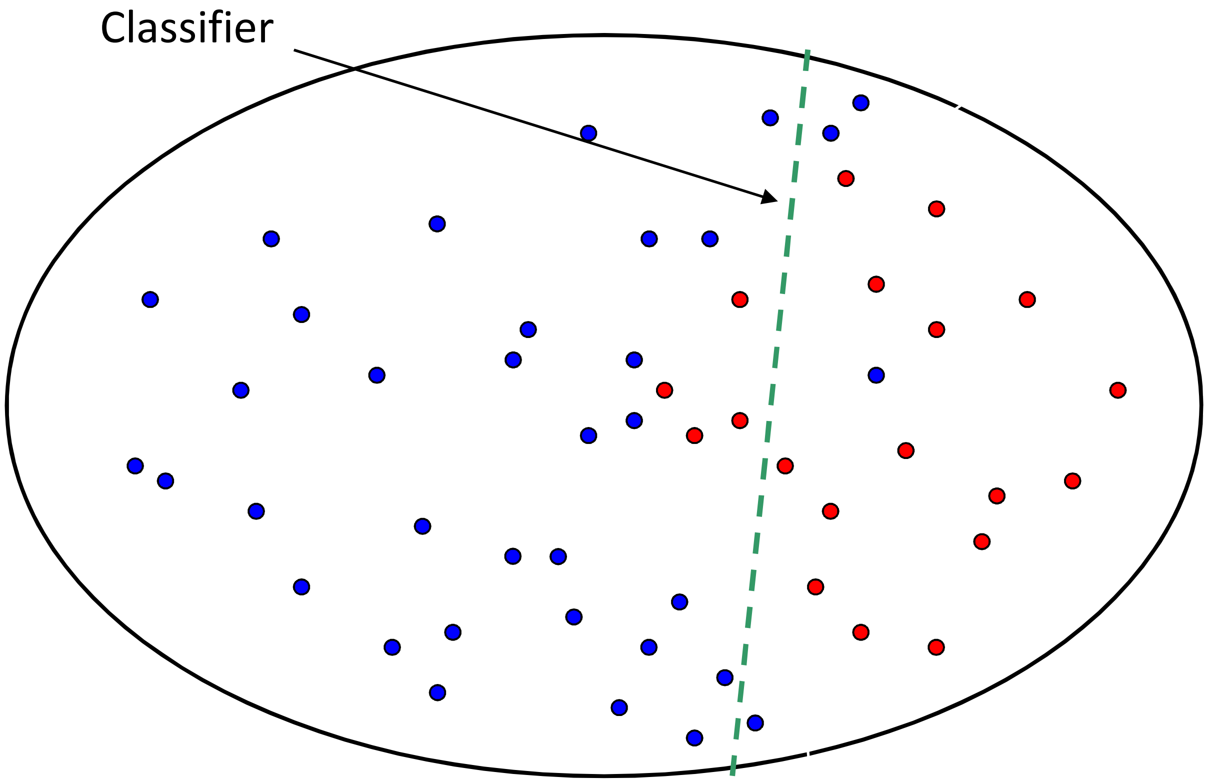
Idea: Make a prediction only when we are very confident

To predict 50% of the examples at 95% is better than predict 100% examples at 80%

The power to say “I don’t know”

➢ Add a reject option to the classifiers

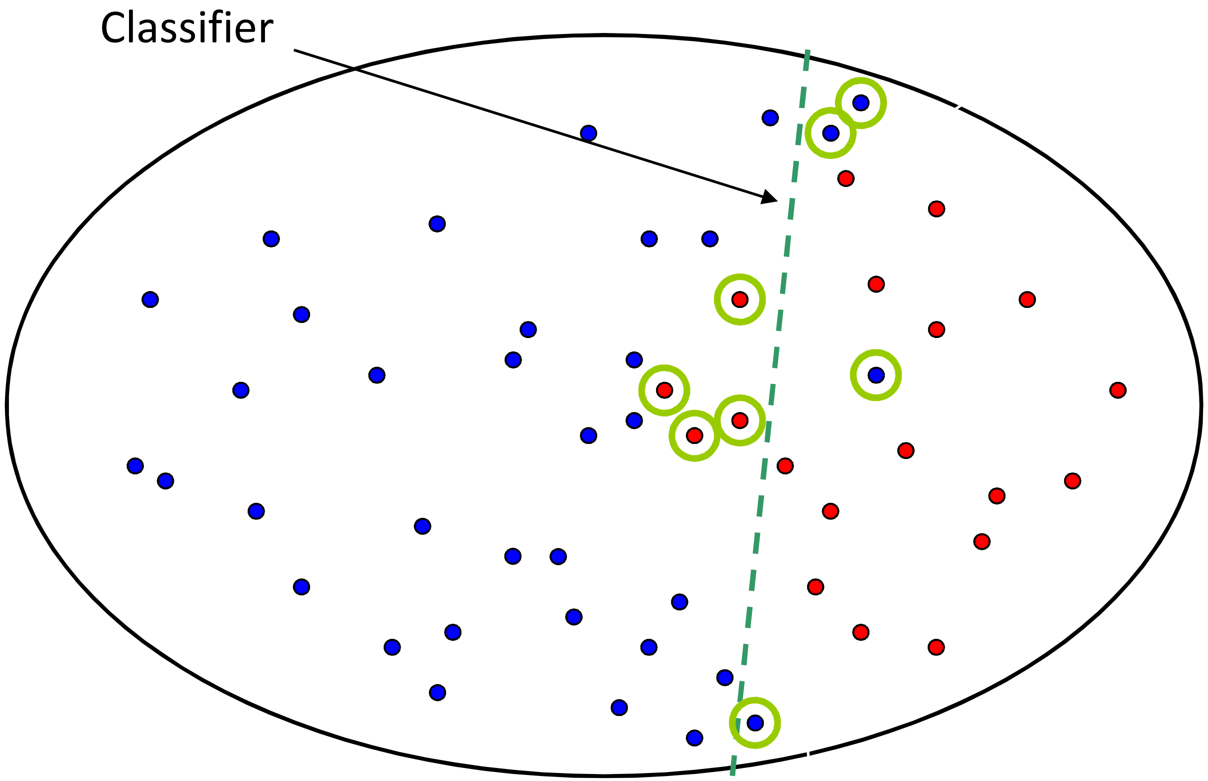
Principle



Training set of N examples

Construction of the classic classifier

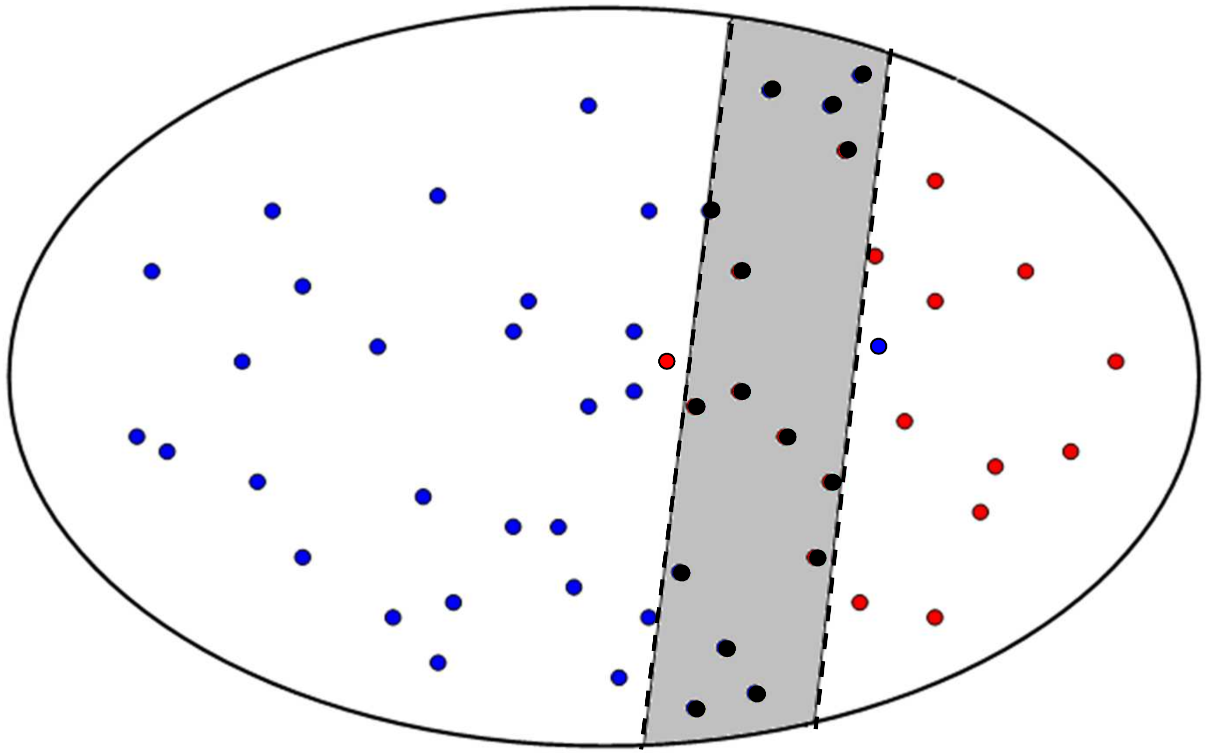
Principle



The errors are close to the seprator

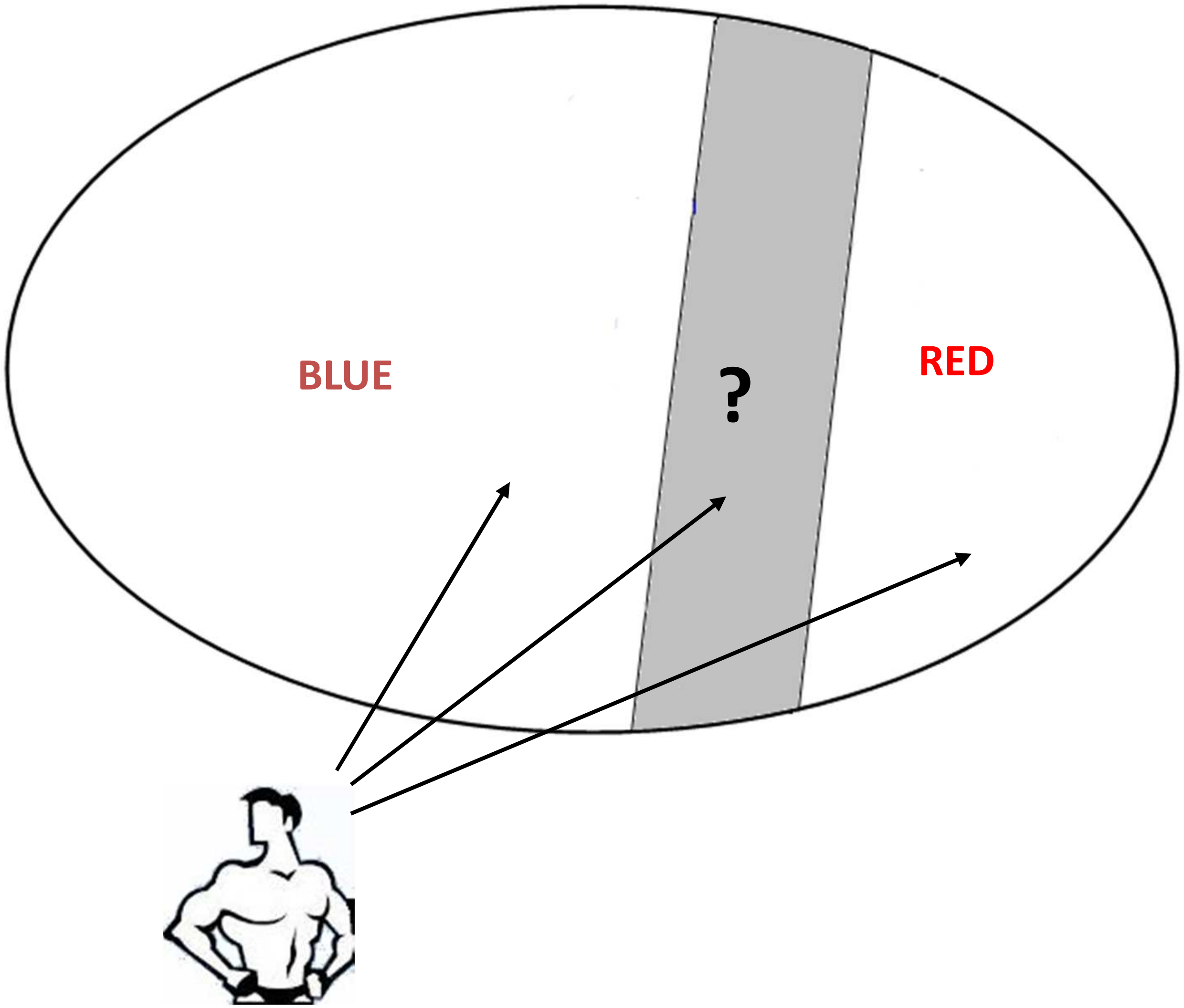
➢ **The farhrest from the separator, the lest number of errors**

Principle

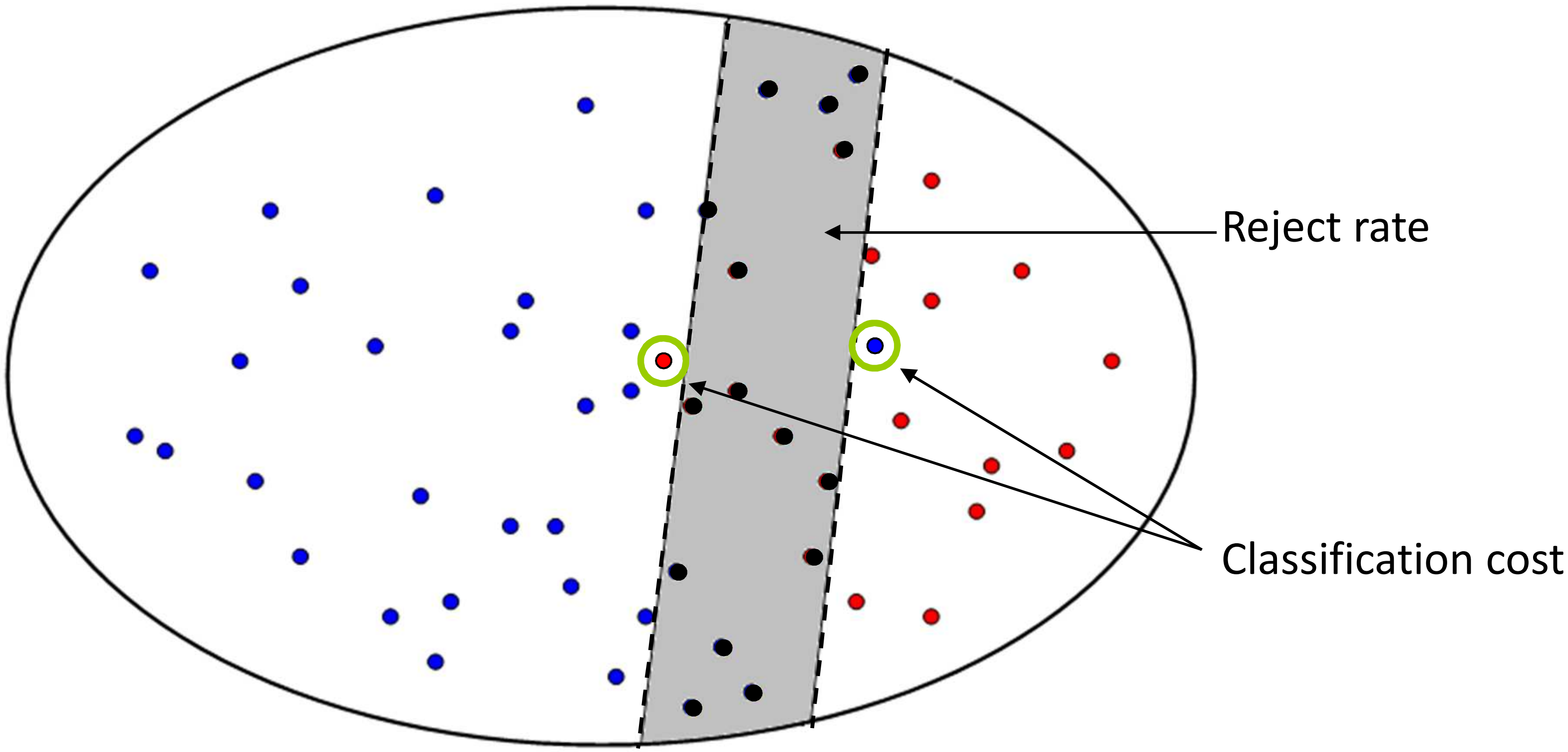


The examples close to the seperator are rejected.

Principle



Performance of classifier with reject option



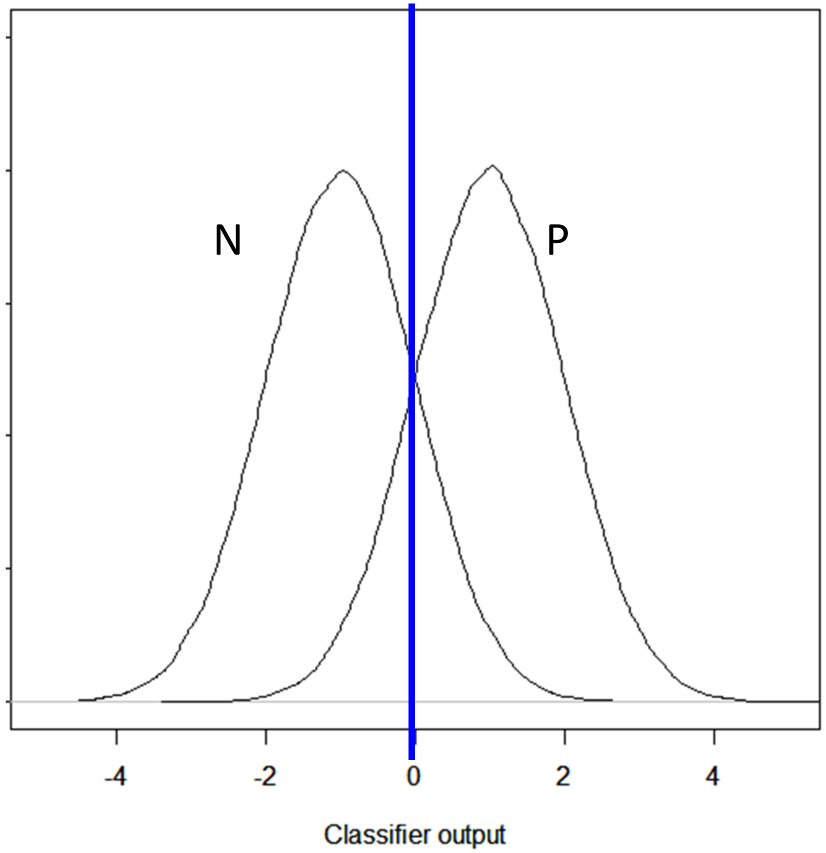
Classic Classifier

Classification cost

Classifier with rejet option

Classification cost
rejection rate

Classifier with reject option



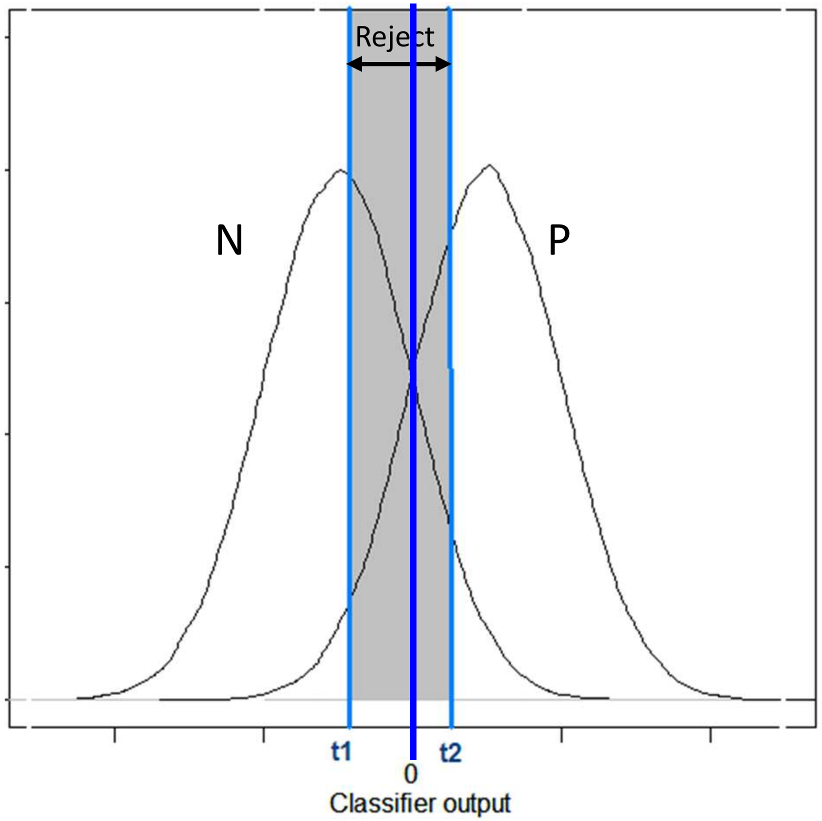
Regular classifier :

$$f : x \mapsto \Re$$

$$\begin{cases} f(x) > 0 \rightarrow P \\ f(x) \leq 0 \rightarrow N \end{cases}$$

[Chow,70][Fumera,00]

Classifier with reject option



$$f : x \mapsto \Re$$

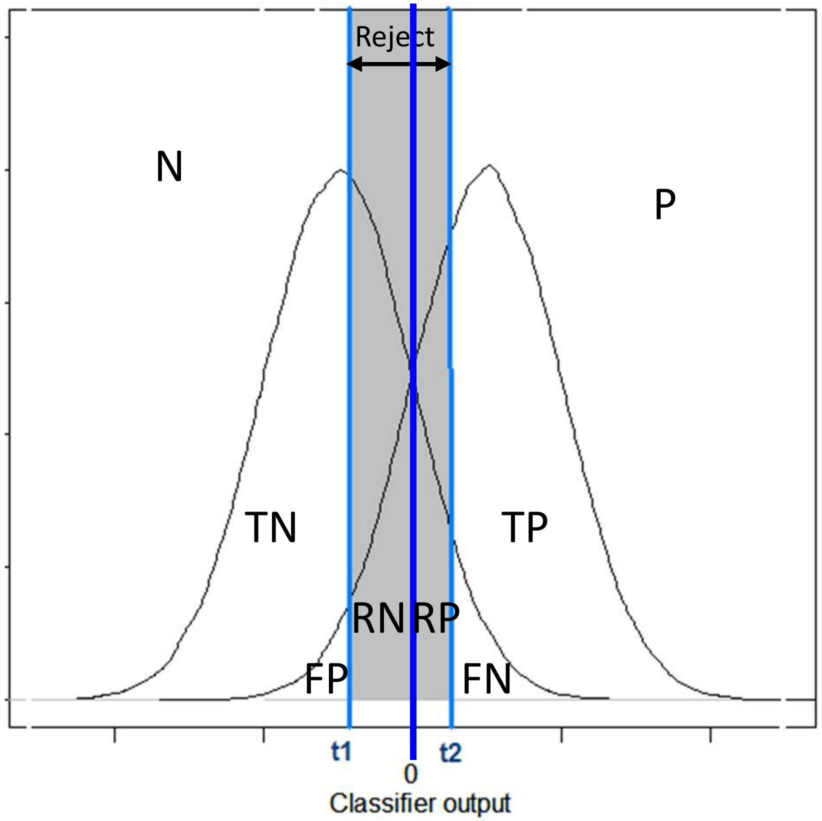
$$\begin{cases} f(x) \geq t_2 \rightarrow P \\ f(x) \leq t_1 \rightarrow N \\ t_2 < f(x) < t_1 \rightarrow R \end{cases}$$

Classifier with reject option :

- Classification model
- Threshold t_1
- Threshold t_2

[Chow,70][Fumera,00]

Classifier with reject option



	True class Positive	True class Negative
Predict Positive	True Positive TP	False Positve FP
Predict Negative	False Negative FN	True Negative TN
Reject	Reject Positive RP	Reject Negative RN
Total	P	N

Classification cost

Classification cost:

$$cost = \pi_P C_{TP}\frac{TP}{P} + \pi_P C_{FN}\frac{FN}{P} + \pi_P C_{RP}\frac{RP}{P} + \pi_N C_{TN}\frac{TN}{N} + \pi_N C_{FP}\frac{FP}{N} + \pi_N C_{RN}\frac{RN}{N}$$

	True class Positive	True class Negative
Predict Positive	C_{TP}	C_{FP}
Predict Negative	C_{FN}	C_{TN}
Reject	C_{RP}	C_{RN}

Classification cost

Classification cost:

$$cost = \pi_P C_{TP}\frac{TP}{P} + \pi_P C_{FN}\frac{FN}{P} + \pi_P C_{RP}\frac{RP}{P} +$$
$$\pi_N C_{TN}\frac{TN}{N} + \pi_N C_{FP}\frac{FP}{N} + \pi_N C_{RN}\frac{RN}{N}$$

Assume that C_{TN}=C_{TP}=0

$$cost = \pi_P C_{FN}\frac{FN}{P} + \pi_P C_{RP}\frac{RP}{P} + \pi_N C_{FP}\frac{FP}{N} + \pi_N C_{RN}\frac{RN}{N}$$

	True class Positive	True class Negative
Predict Positive	0	C_{FP}
Predict Negative	C_{FN}	0
Reject	C_{RP}	C_{RN}

Classification cost

Classification cost:

$$cost = \pi_P C_{TP} \frac{TP}{P} + \pi_P C_{FN} \frac{FN}{P} + \pi_P C_{RP} \frac{RP}{P} + \pi_N C_{TN} \frac{TN}{N} + \pi_N C_{FP} \frac{FP}{N} + \pi_N C_{RN} \frac{RN}{N}$$

Assume that C_{FN}=C_{FP}=0

$$cost = \pi_P C_{FN} \frac{FN}{P} + \pi_P C_{RP} \frac{RP}{P} + \pi_N C_{FP} \frac{FP}{N} + \pi_N C_{RN} \frac{RN}{N}$$

Assume that C_{FN}=1, $Q = \frac{C_{FP}}{C_{FN}}$ π_P=π_N=1/2

C_{RN}=C_{RP}=C_R

$$cost = \frac{1}{2}\left(\frac{FN}{P} + Q\frac{FP}{N} + C_R\left(\frac{RP}{P} + \frac{RN}{N}\right)\right)$$

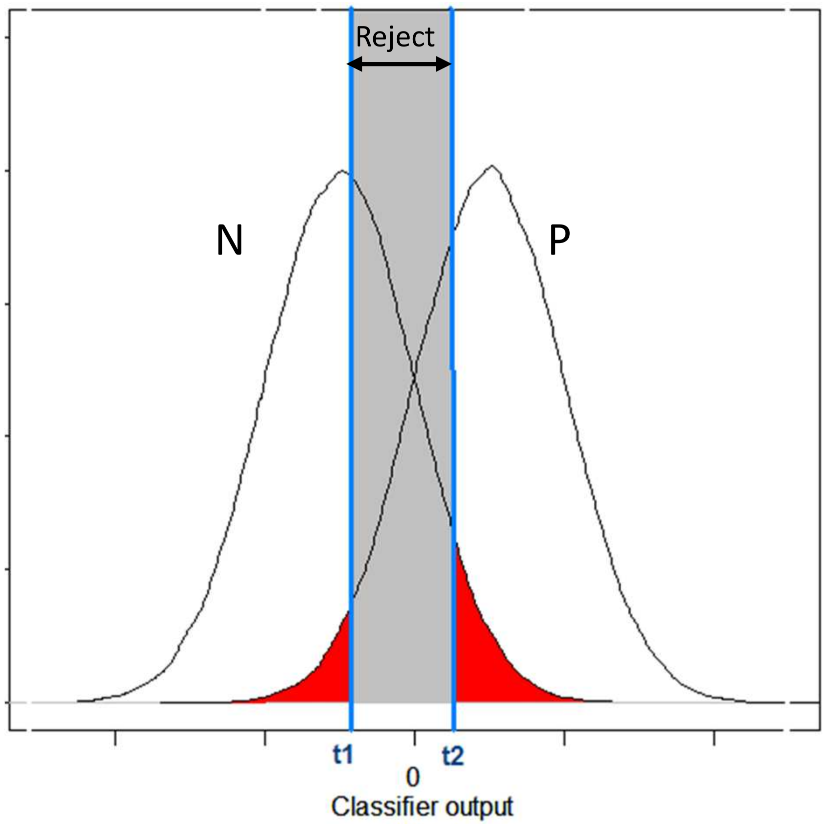
$$cost = \frac{1}{2}\left(\frac{FN}{P} + Q\frac{FP}{N} + C_R\left(\frac{R}{P+N}\right)\right)$$

Si Q=1

$$cost = error.rate + C_R\ reject.rate$$

	True class Positive	True class Negative
Predict Positive	0	Q
Predict Negative	1	0
Reject	C_R	C_R

Performance of a classifier with reject option



$$\boxed{p(reject)} + p(accept) = 1$$

$$p(x \in P;\ f(x) = P; accept)$$
$$p(x \in N;\ f(x) = N; accept)$$

$$p(good; accept)$$

$$p(x \in P;\ f(x) = P; accept)$$
$$p(x \in N;\ f(x) = N; accept)$$

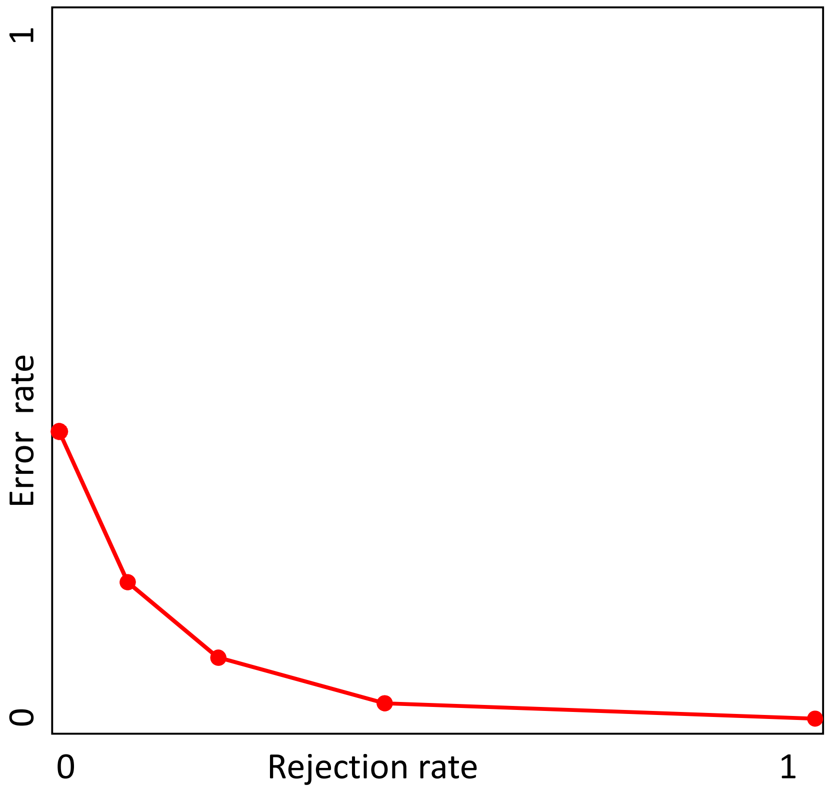
$$p(error; accept)$$

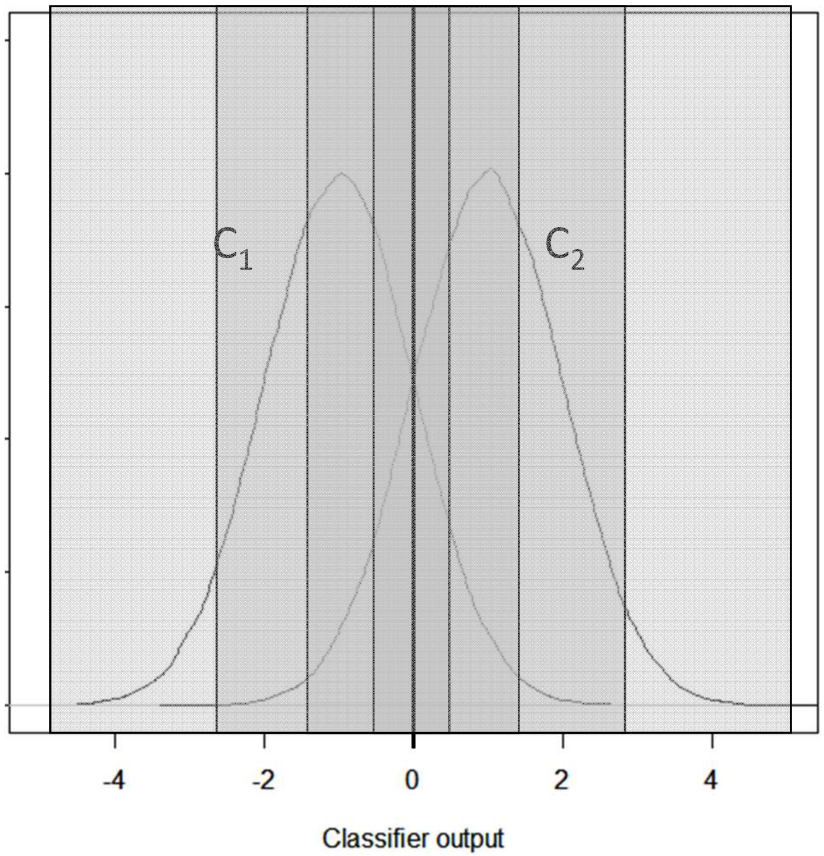
$$p(error; accept) + p(good; accept) + p(reject) = 1$$

$$p(error \mid accept)$$
$$p(good \mid accept)$$

$$\boxed{p(error \mid accept)} + p(good \mid accept) = 1$$

Error-reject Trade-off

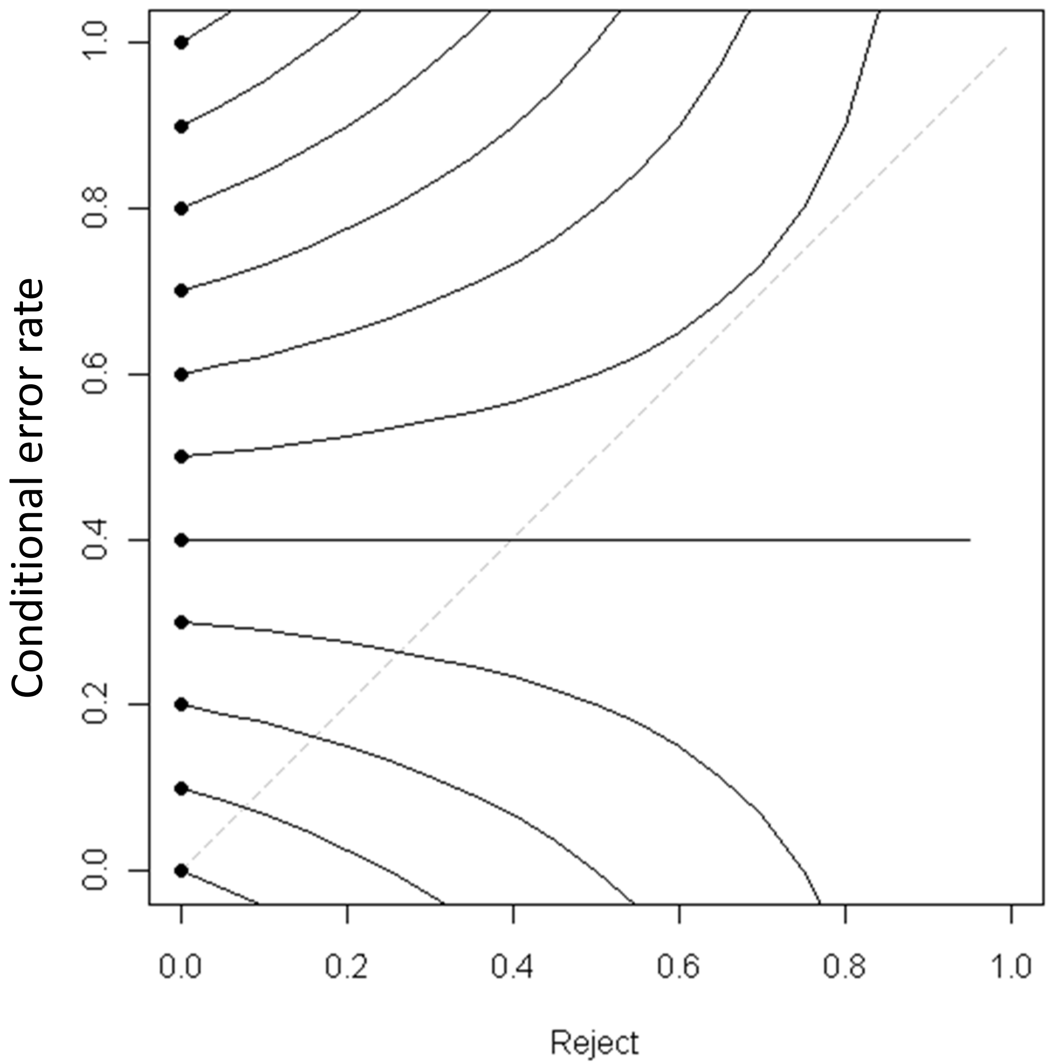




Accuracy increase with the rejection rate

What is the best trade-off between error and rejection rate ?

Iso-cost line



C_R=0.4

Define the reject option

- What is the best trade-off between error and rejection rate ?
- How to choice the decision threshold that define the reject option ?
  - Cost optimization
  - ROC analysis
  - Performance constraints

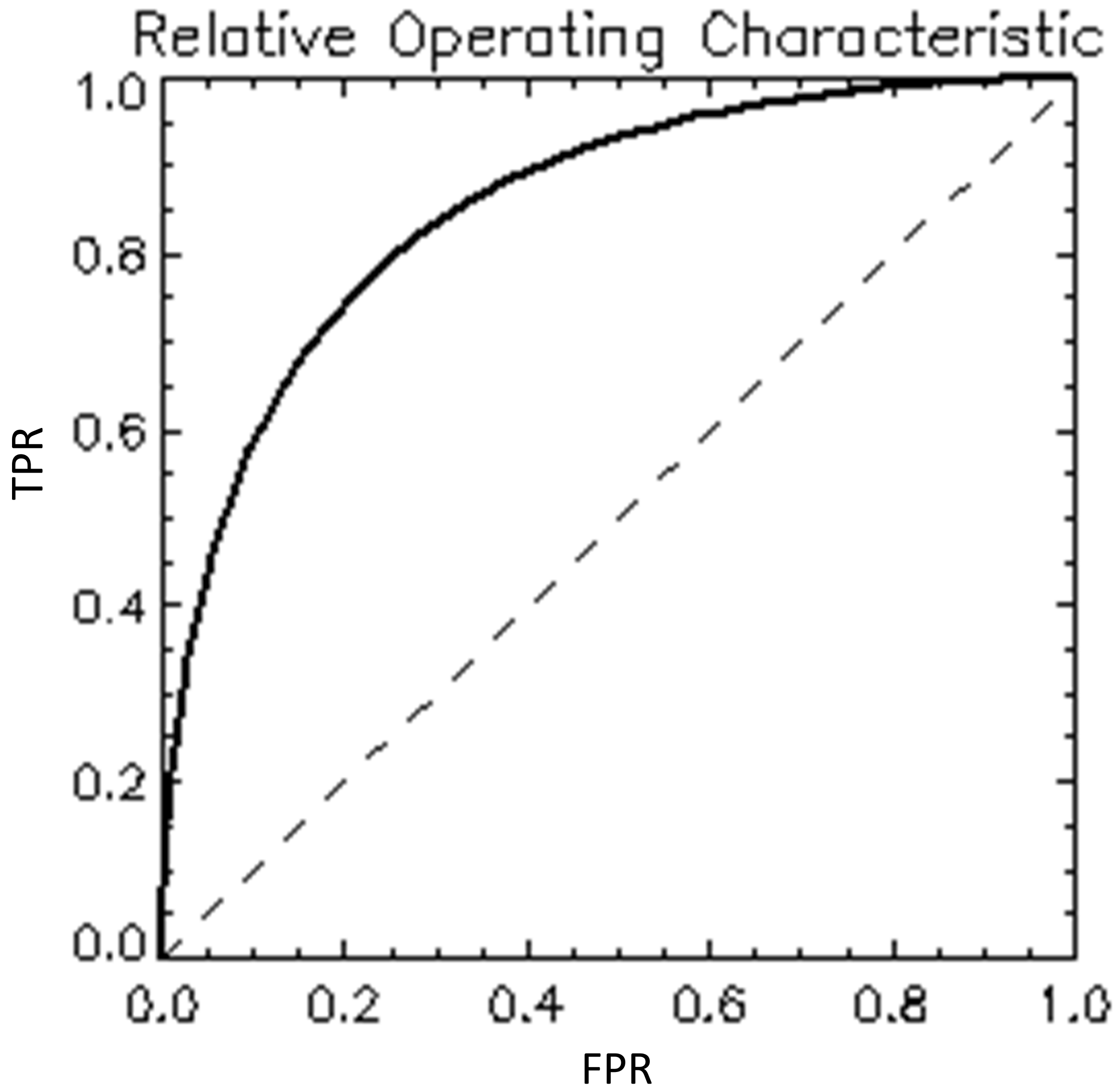
Cost optimization

- Reject classifier can be view as a combination of two classifiers
- The reject option that minimizes the classification cost can be analytically computed

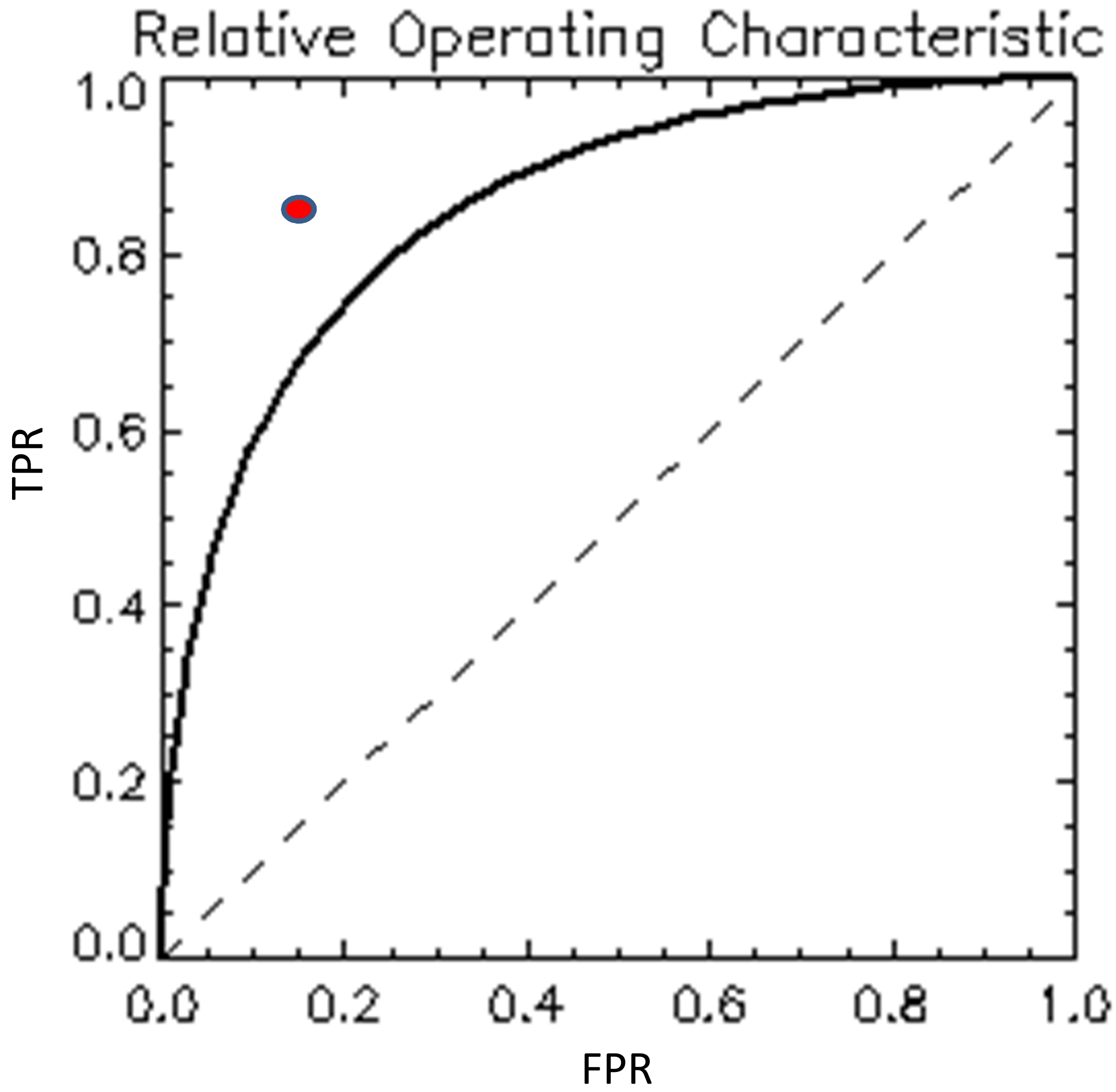
$$-f'_{ROC}(FP^*_\beta) = \frac{C_{RN}}{C_{FN} - C_{RP}} \frac{N}{P}$$

$$-f'_{ROC}(FP^*_\alpha) = \frac{C_{RP}}{C_{FP} - C_{RN}} \frac{N}{P}$$

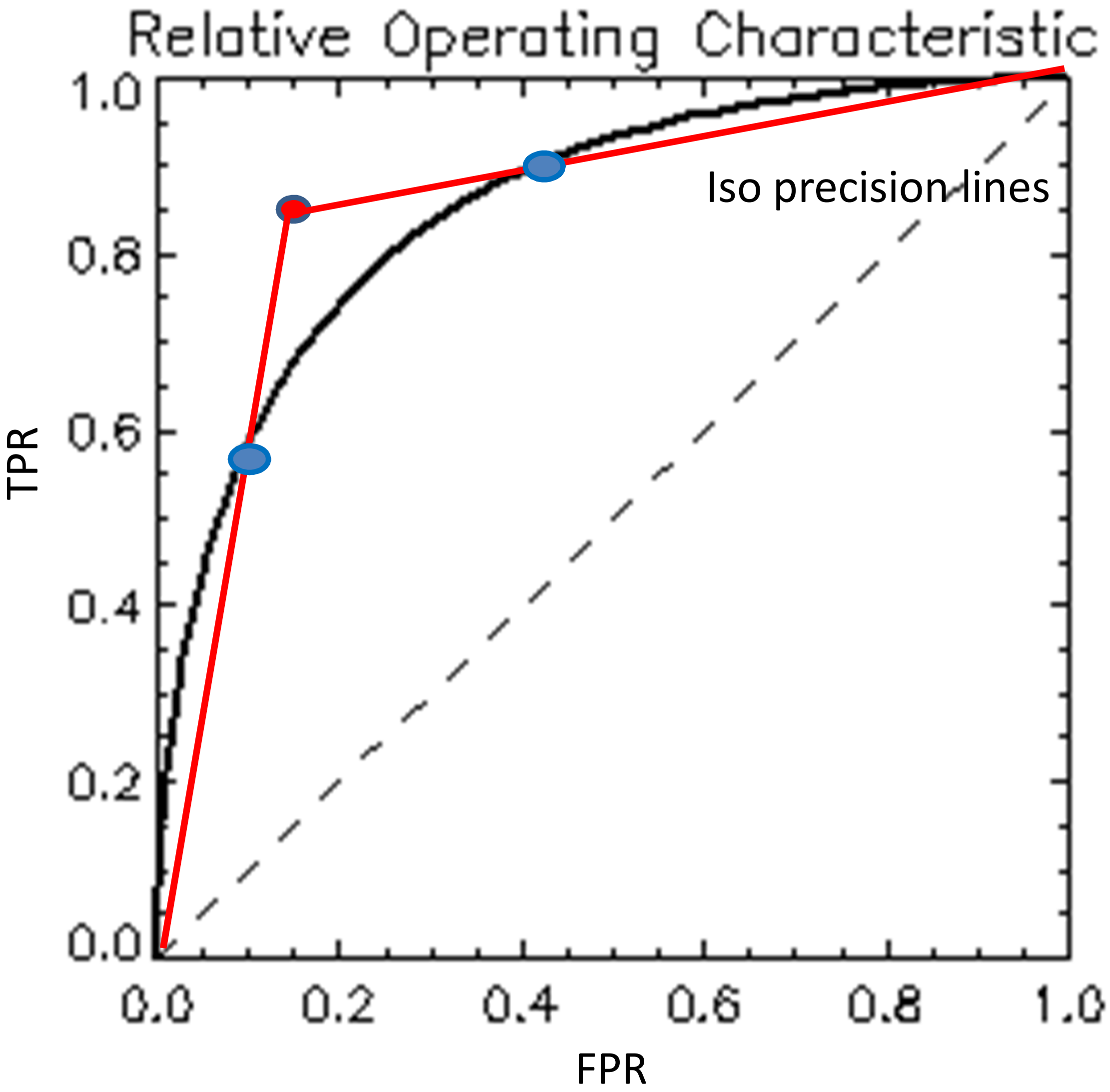
ROC analysis



ROC analysis



ROC analysis



Error-reject Trade-off

- Control the classification cost

Classification cost becomes a parameter of classification algorithm

Learning task:

Minimize the rejection rate given the target false positive and negative

$$\min |t_1 - t_2|$$

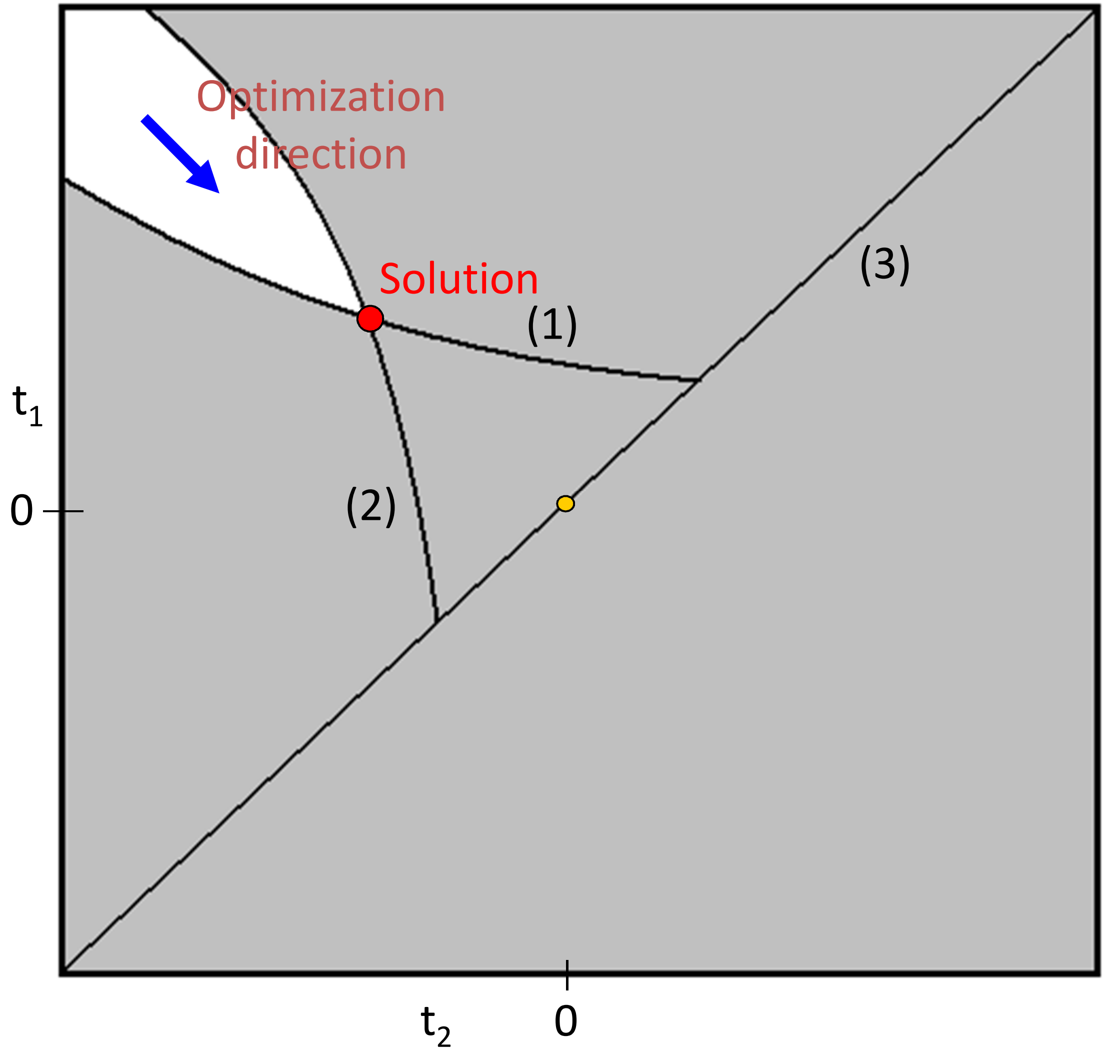
$$\begin{cases} FPR^{cond} \le FPR^* \\ FNR^{cond} \le FNR^* \end{cases}$$

Target false positive and negative

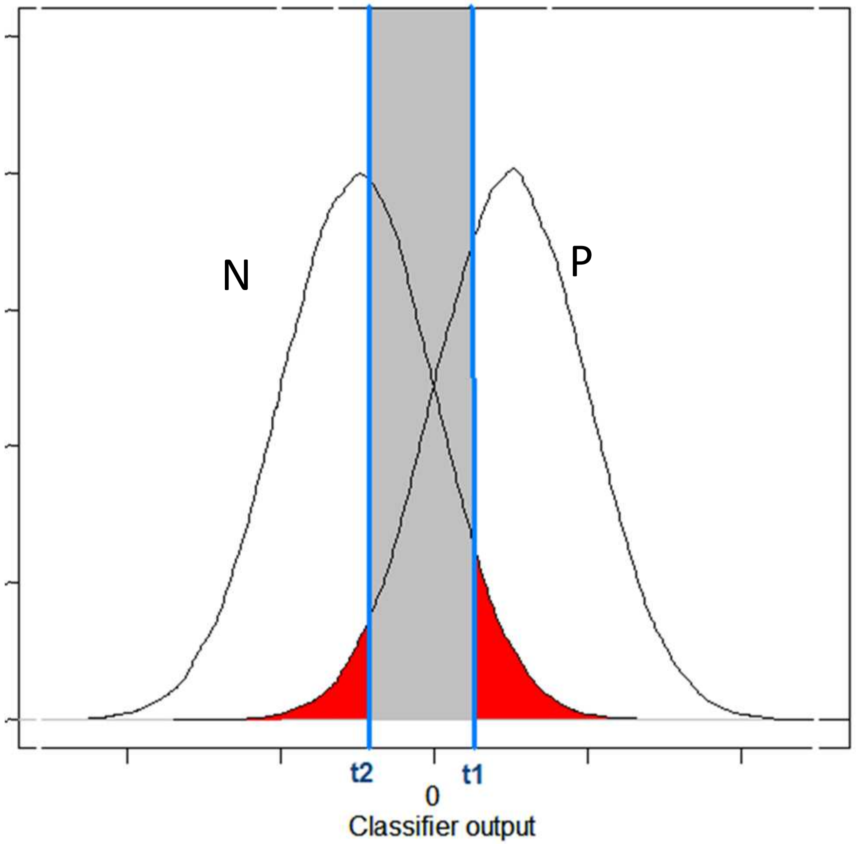
Optimization problem

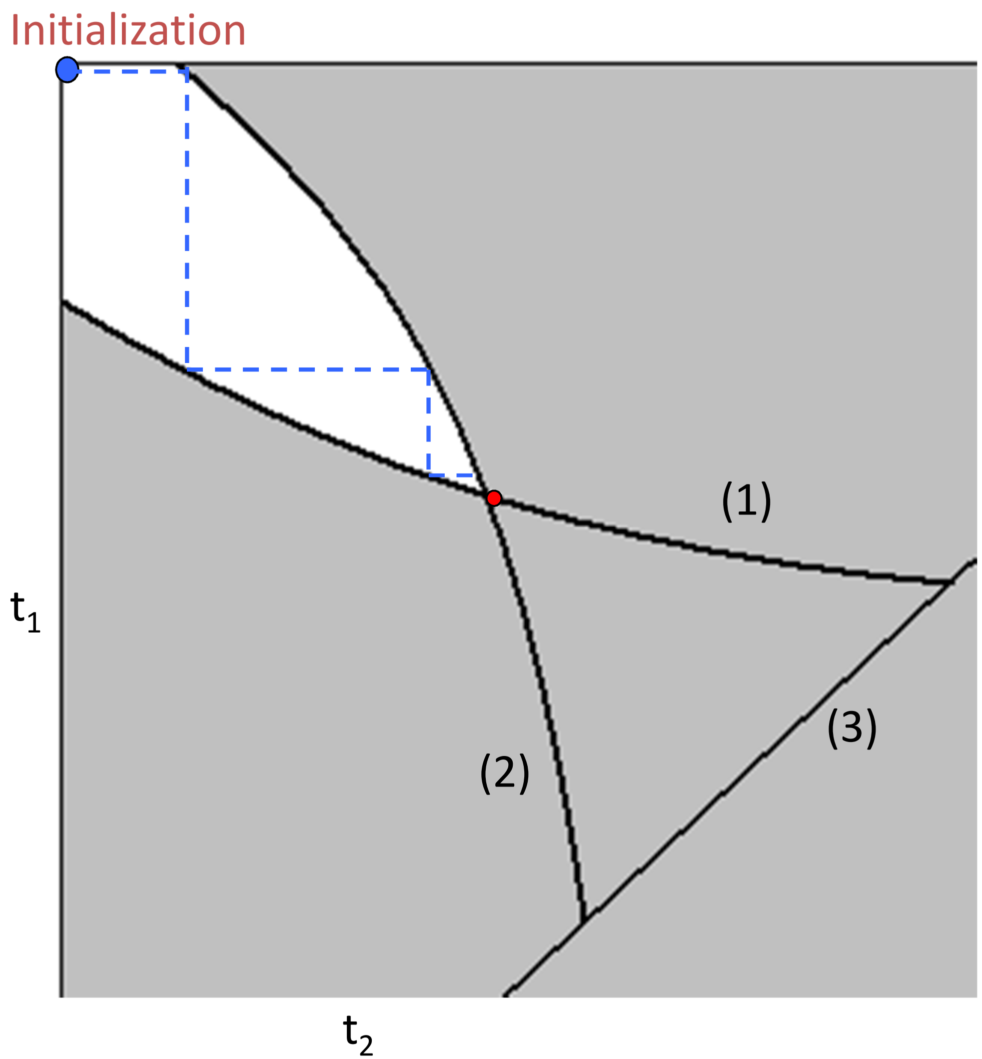
$$\min(t_1 - t_2)$$

$$\begin{cases} \boxed{FNR^{cond} \leq FPR^*} & (1) \\ \boxed{FNR^{cond} \leq FNR^*} & (2) \\ \boxed{t_2 \leq t_1} & (3) \end{cases}$$

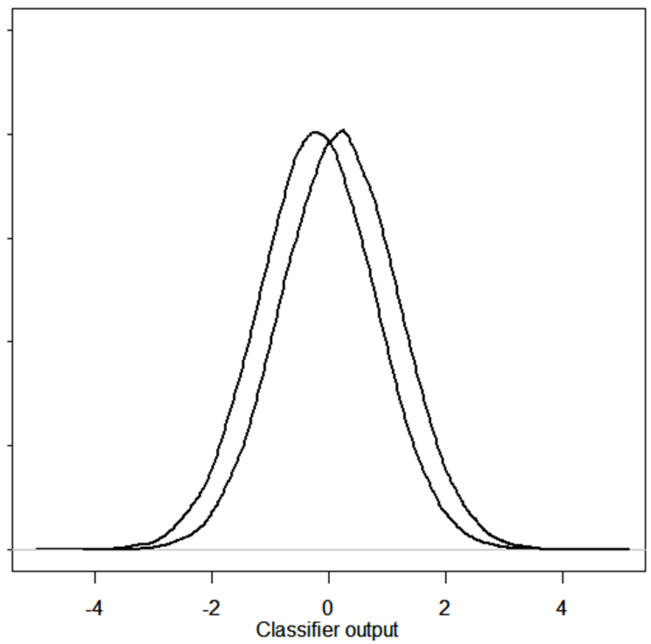


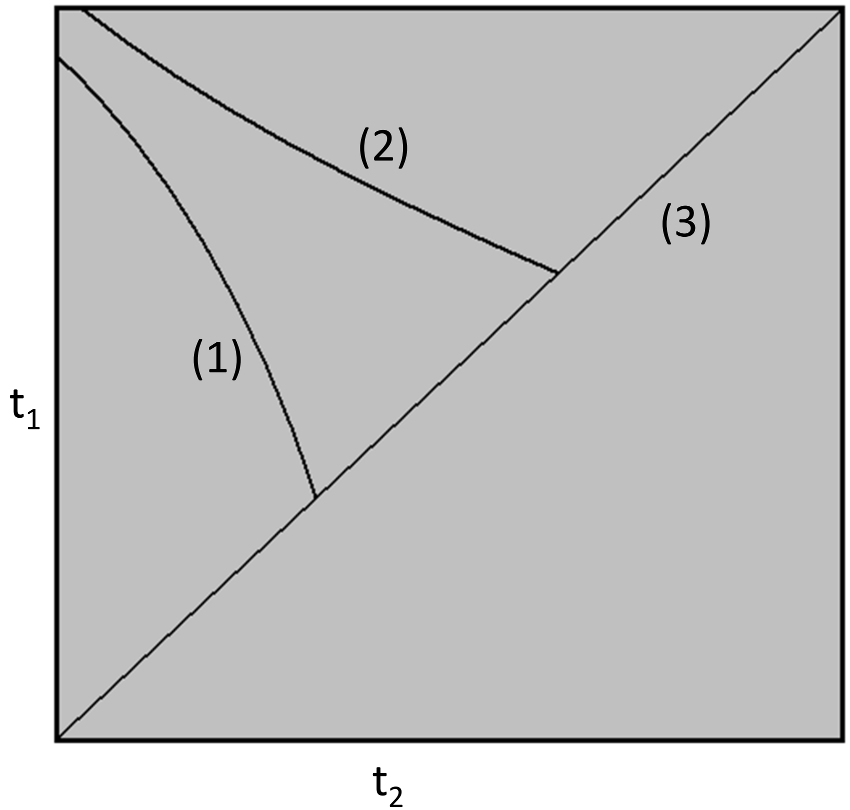
Heuristic search





No solution



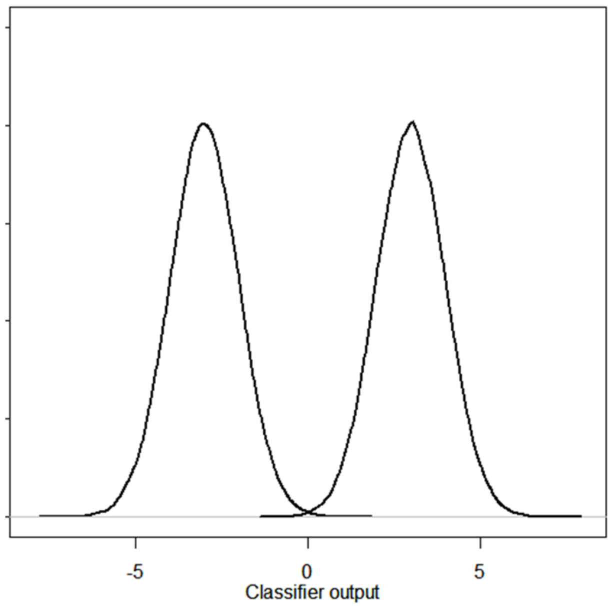


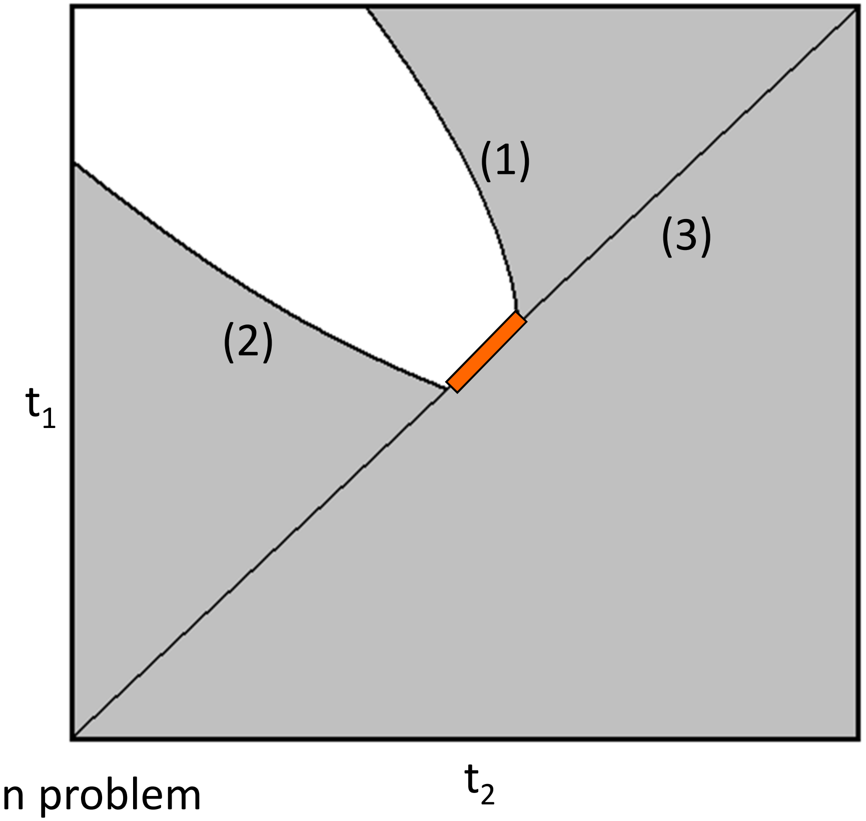
Domain of validity is empty

No solution to the optimization problem

Classification construction fails

Multiple solutions





Several solutions to the optimization problem

Classifier has only one threshold

Reject option not necessary to respect the target error constraints

Muti-class classification

Classification task with K classes $\{C_1, C_2, \ldots C_K\}$

Currently, there is no clear solution to the problem of multi-class classification

3 approaches:

- One vs All
- All vs All
- Hierarchical classification

One vs All

K classifiers: a class VS all other classes

C_1 VS $\{C_2, C_3, \ldots C_K\}$ -> $score_1$

C_2 VS $\{C_1, C_3, \ldots C_K\}$ -> $score_2$

…

C_i VS $\{C_1, \ldots, C_{i-1}, C_{i+1}, \ldots C_K\}$ -> $score_i$

…

C_K VS $\{C_1, C_2, \ldots C_{K-1}\}$ -> $score_K$

C^* <- $\mathrm{argmax}_i\{score_1, score_2, \ldots score_i, \ldots score_K\}$

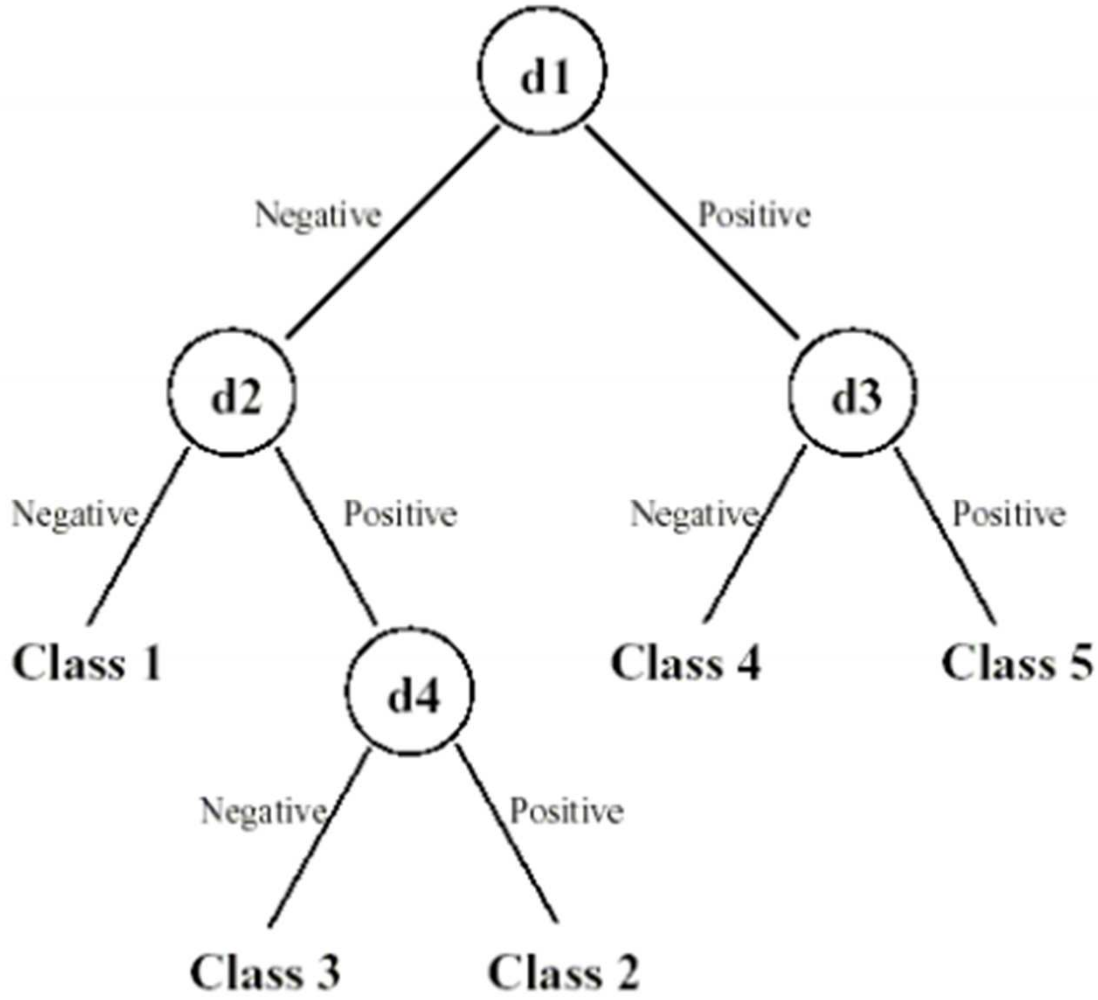
All VS All

K(K-1)/2 binary classifiers: C_i VS C_j for all i>j

	C_1	C_2	…	C_j	…	C_K	Total
C_1		0		1		0	6
C_2	1			1		0	5
…						0	
C_i	1	1				0	10
…							
C_K	0	0		0			2

C^* <- $\mathrm{argmax}_i\{Total_1, Total_2,\ldots Total_i,\ldots Total_K\}$

Hierarchical classification



Performance of multi-class classifiers

Confusion matrix

	C_1	C_2	…	C_j	…	C_K
C_1	M_{11}	M_{12}		M_{1j}		M_{1K}
C_2	M_{21}	M_{22}				
…						
C_i	M_{j1}	M_{j2}		M_{jj}		M_{jK}
…						
C_K	M_{K1}	M_{K2}		M_{Kj}		M_{KK}

Cost Matrix

	C_1	C_2	…	C_j	…	C_K
C_1	α_{11}	α_{12}		α_{1j}		α_{1K}
C_2	α_{21}	α_{22}				
…						
C_i	α_{j1}	α_{j2}		α_{jj}		α_{jK}
…						
C_K	α_{K1}	α_{K2}		α_{Kj}		α_{KK}

No ROC space can be defined

Evaluation done by classification cost