PA196: Pattern Recognition

04. Classifier performance: parameters, estimation and comparison

Dr. Vlad Popovici
popovici@recetox.muni.cz

RECETOX
Masaryk University, Brno

Specific bibliography

- W.J. Krzanowski, D.J. Hand: ROC curves for continuous data. CRC Press. 2009
- M.S. Pepe: The statistical evaluation of medical tests for classification and prediction. Oxford Univ Press. 2003
- N. Japkowicz, M. Shah. Evaluating Learning Algorithms: A Classification Perspective. Cambridge Univ Press. 2011

Outline

1. Performance parameters
   - Introduction
   - Performance parameters for binary classifiers
   - Performance parameters for continuous outputs
2. Performance estimation
3. Performance comparison
4. An example

Outline

1. Performance parameters
   - Introduction
   - Performance parameters for binary classifiers
   - Performance parameters for continuous outputs
2. Performance estimation
3. Performance comparison
4. An example

Context

- binary classifiers: $Y(\mathbf{x}) \in \{0, 1\}$ is the predicted class label
- Y is obtained usually from some discriminant function $h(\mathbf{x}) \in \mathbb{R}$: $Y = \mathbb{I}[h(\mathbf{X}) \geq \theta]$
- $h(\mathbf{x})$ (be it margin, posterior probability, etc) can be interpreted as a *score*
- let C be the true label (0 or 1): *gold standard* or *ground truth*
- we assume symmetric loss

In medical applications...

- a classifier is often called *a test*
- the class of interest usually refers to an abnormal condition (e.g. "diseased")
- "positive test" indicates that the abnormal condition is predicted
- tests:
  - *diagnostic*: detect the presence of disease
  - *prognostic*: predict a clinical outcome (e.g. "recurrence" vs "non-recurrence")
  - *screening*: a test is applied to a large population to detect the presence of an abnormal condition with low prevalence; it is usually followed by other tests

Confusion matrix

	Gold standard		
	$C = 0$	$C = 1$	
$Y = 0$	true negative	false negative	
$Y = 1$	false positive	true positive	

Goal

Estimate conditional and marginal probabilities.

Confusion matrix

	Gold standard		
	$C = 0$	$C = 1$	
$Y = 0$	true negative $P[Y = 0 \mid C = 0]$	false negative $P[Y = 0 \mid C = 1]$	$P[Y = 0]$
$Y = 1$	false positive $P[Y = 1 \mid C = 0]$	true positive $P[Y = 1 \mid C = 1]$	$P[Y = 1]$
	$P[C = 0]$	$P[C = 1]$	

Goal

Estimate conditional and marginal probabilities.

- estimation is based on a *finite* test sample $\{(Y_i, C_i)|i = 1, \ldots, n\}$ i.i.d. drawn from the population
- the probabilities will be estimated in terms of fractions/ proportions from the test sample

Confusion matrix based on the test sample:

	Gold standard		
	$C = 0$	$C = 1$	
$Y = 0$	$n^-_{\bar{C}}$	n^-_C	n^-
$Y = 1$	$n^+_{\bar{C}}$	n^+_C	n^+
	$n_{\bar{C}}$	n_C	n

C indicates the "positive class" ($C = 1$) and $\bar{C}$ indicates the "negative class" $C = 0$.

Notes on the *sampling* of the test set: the most frequent ways of selecting the test set are

- i.i.d. from the underlying distribution → it means that it also approximates well the class priors (prevalence); in clinical studies this is called "cohort study"
- "case-control": a fixed number of positive (cases) and negative (controls) samples are randomly selected from the population → the class priors are not respected

In the following, i.i.d. sampling is implied, unless stated otherwise.

Outline

1. Performance parameters
   - Introduction
   - Performance parameters for binary classifiers
   - Performance parameters for continuous outputs
2. Performance estimation
3. Performance comparison
4. An example

Basic performance parameters

- a performance parameter P is a random variable, and we only estimate it as $\hat{P}$
- however, to simplify notation we will denote the parameter simply as P even when referring to its estimate - the meaning is clear from context
- error rate or proportion of misclassified samples:
  $\text{Err} = P[Y \neq C] \rightarrow \frac{n^+_{\bar{C}} + n^-_C}{n}$
- false positive fraction: $\text{FPF} = P[Y = 1 | C = 0] \rightarrow \frac{n^+_{\bar{C}}}{n^+_{\bar{C}} + n^-_{\bar{C}}}$
  (aka 1-Specificity (Sp))
- true positive fraction: $\text{TPF} = P[Y = 1 | C = 1] \rightarrow \frac{n^+_C}{n^+_C + n^-_C}$
  (aka Sensitivity (Se))

- let $\rho = P[C = 1]$ be the *prevalence* of the positive cases
- then

$$\text{Err} = \rho(1 - \text{TPF}) + (1 - \rho)\,\text{FPF}$$

- Positive Predicted Value:
  $\text{PPV} = P[C = 1|Y = 1] \to \frac{n_C^+}{n_C^+ + n_{\bar{C}}^+}$
- Negative Predicted Value:
  $\text{NPV} = P[C = 0|Y = 0] \to \frac{n_{\bar{C}}^-}{n_{\bar{C}}^- + n_C^-}$
- perfect classifier/test: $\text{PPV} = \text{NPV} = 1$
- totally uninformative classifier/test:
  $\text{PPV} = \rho, \text{NPV} = 1 - \rho$
-

$$\text{PPV} = \frac{\rho\,\text{TPF}}{\rho\,\text{TPF} + (1-\rho)\,\text{FPF}}$$

$$\text{NPV} = \frac{(1-\rho)(1-\text{FPF})}{(1-\rho)(1-\text{FPF}) + \rho(1-\text{TPF})}$$

- in information-retrieval applications: recall stands for TPF and precision stands for PPV
- *F*–measure:

$$F_\alpha = \frac{(1-\alpha)(\text{precision} \times \text{recall})}{\alpha \times \text{precision} + \text{recall}}$$

- *Matthews correlation coefficient*

$$MCC = \frac{n_C^+ \times n_{\bar{C}}^- - n_{\bar{C}}^+ \times n_C^-}{\sqrt{(n_C^+ + n_{\bar{C}}^+)(n_C^+ + n_C^-)(n_{\bar{C}}^- + n_{\bar{C}}^+)(n_{\bar{C}}^- + n_C^-)}}$$

Correcting for chance...

- Example 1: let the prevalence of positive cases be $\rho = 0.75$ and consider a classifier that predicts "1" or "0" with equal probabilities (flip of a coin)
- simply by chance, the classifier will be right in $0.5 \times 0.75 = 0.375$ proportion of cases
- Example 2: medical imaging: the true labels are not known, but there is an expert producing a labelling and the classifier produces another set of labels
- how can we compare the two, taking into account the concordances due to mere chance?

...using agreement statistics

- probability of observed agreement between classifier and the true labels: $P_o = \frac{n_{\bar{C}}^{-} + n_C^{+}}{n}$
- S–coefficient is defined as $S = 2P_o - 1$
- by taking into account the chance agreement (P_e): what is the ratio between the difference between observed and expected chance agreement ($P_o - P_e$) and maximum possible agreement beyond chance:

$$\frac{P_o - P_e}{1 - P_e}$$

- if the estimation of chance agreement is

$$P_e = \left(\frac{n^+ + n_C}{2n}\right)^2 + \left(\frac{n^- + n_{\bar{C}}}{2n}\right)^2$$

  the fraction is denoted as $\pi = \frac{P_o - P_e}{1 - P_e}$ and is called *Scott's π coefficient*
- if the estimation of chance agreement is

$$P_e = \frac{n^+ \times n_C}{n^2} + \frac{n^- \times n_{\bar{C}}}{n^2}$$

  the fraction is denoted as $\kappa = \frac{P_o - P_e}{1 - P_e}$ and is called *Cohen's kappa coefficient*
- in medical applications, κ is usually used for measuring the agreement between an expert and another system

Confidence intervals (CI)

- need ways for characterizing the uncertainty in the estimates
- informally, CI is a measure of reliability of the estimates; sample-based (observed)
- confidence level: how often the confidence interval contains the estimated value
- the values within the CI can be seen as alternative estimates of the parameter of interest
- smaller the sample size, larger the CI
- the (TPF, FPF) and (PPV, NPV) are r.v. from a Bernoulli trial

- *Bernoulli trial*: experiment with a random binary outcome
- *binomial distribution*: discrete pdf of the number of successes in n independent Bernoulli trials with success probability p
- $X \sim \mathcal{B}(n, p)$:

$$P[X = k] = \binom{n}{k} p^k (1-p)^{n-k}$$

$$\mathrm{E}[X] = np$$

$$\mathrm{Var}[X] = np(1-p)$$

- as $n \to \infty$,

$$\frac{X - np}{\sqrt{np(1-p)}} \sim \mathcal{N}(0, 1)$$

- simplest CI: normal approximation: a $1 - \alpha$ CI for the binomial parameter p (*proportion of successes (between 0 and 1)* in n trials) is

$$p \pm z_{1-\alpha/2} \sqrt{\frac{p(1-p)}{n}}$$

where $z_{1-\alpha/2}$ is the $1 - \alpha/2$ percentile of standard normal distribution (e.g. for 95% CI, $\alpha = 0.05$ and $z_{0.975} \approx 1.96$)

Warning

The normal approximation is poor for FPF or TPF close to 0 or 1.

Agresti-Coull $1 - \alpha$ CI:

- let n be the number of trials and p *the number of successes*, then let $\tilde{n} = n + z^2_{1-\alpha/2}$ and $\tilde{p} = \frac{1}{\tilde{n}}\left(p + \frac{1}{2}z^2_{1-\alpha/2}\right)$, then a good approximation for the CI is

$$\tilde{p} \pm z_{1-\alpha/2}\sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{n}}}$$

- other formulas for CI: Wilson score intervals; Clopper–Pearson interval, Bayesian CIs

Example: a test for predicting pCR in breast cancer yields

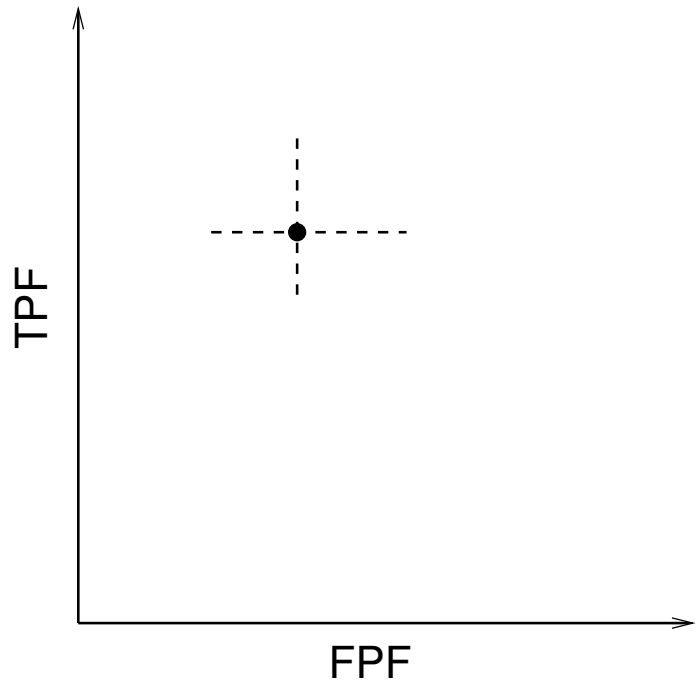
	pCR=0	pCR=1
predicted 0	61	5
predicted 1	24	10

$$\text{TPF} = 0.67, \quad \text{FPF} = 0.28$$
$$\text{PPV} = 0.29, \quad \text{NPV} = 0.92$$

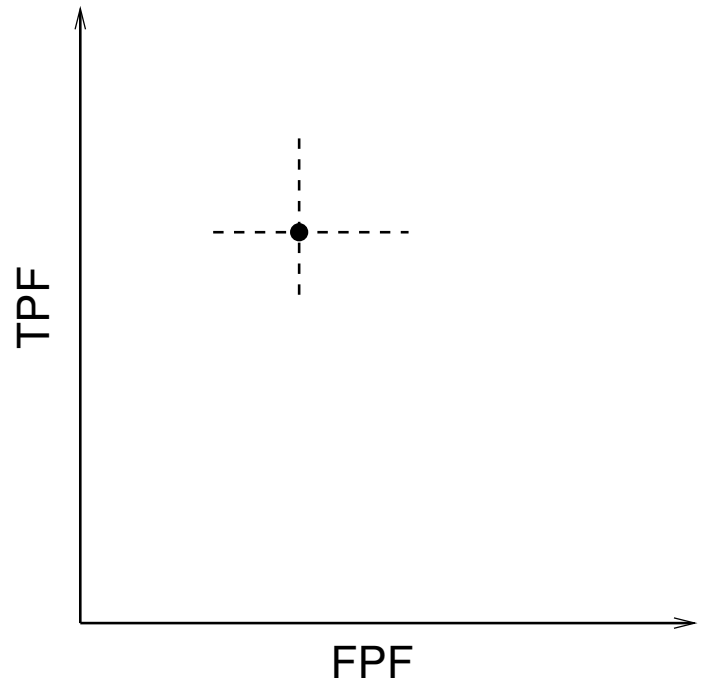
95% confidence intervals:

- normal approx.: $\text{FPF} \in (0.197, 0.391)$, $\text{TPF} \in (0.428, 0.905)$
- Wilson: $\text{FPF} \in (0.208, 0.398)$, $\text{TPF} \in (0.417, 0.848)$
- Bayesian: $\text{FPF} \in (0.205, 0.397)$, $\text{TPF} \in (0.416, 0.860)$

Joint confidence intervals

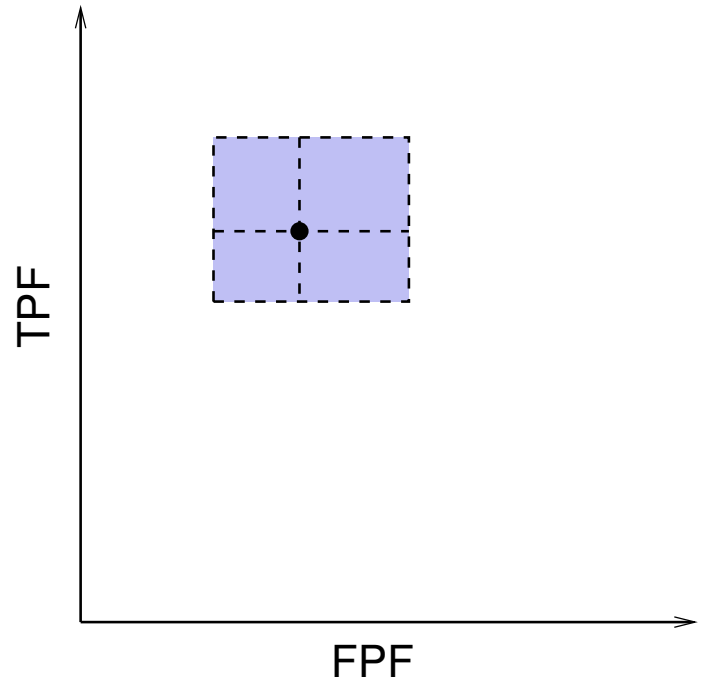


Joint confidence intervals



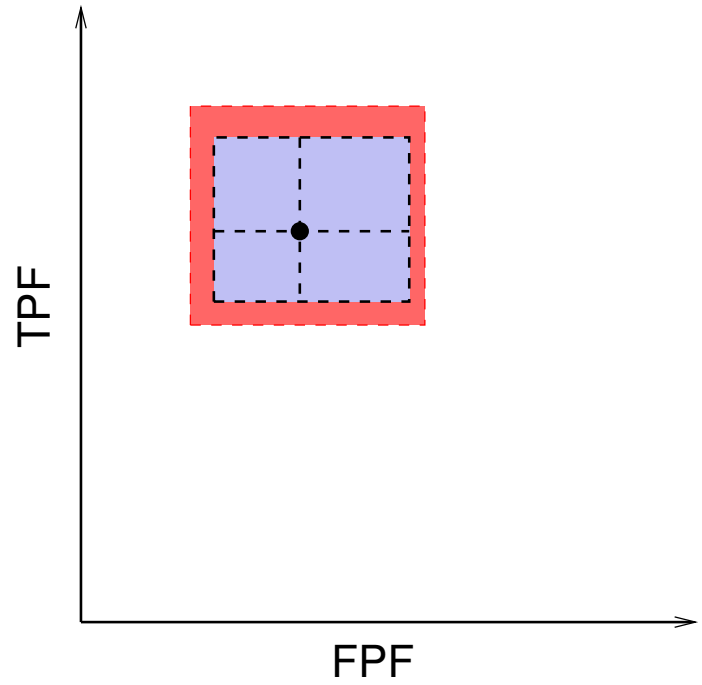
- what is the joint 100(1 – α)% confidence region for (FPF, TPF)?

Joint confidence intervals



- what is the joint 100(1 – α)% confidence region for (FPF, TPF)?

Joint confidence intervals



- what is the joint $100(1-\alpha)\%$ confidence region for (FPF, TPF)?

Rectangular confidence regions

If (P_{low}, P_{up}) and (Q_{low}, Q_{up}) are the $1 - \alpha^*$ univariate confidence intervals for two binomial random variables P and Q, then the rectangle

$$R \equiv (P_{low}, P_{up}) \times (Q_{low}, Q_{up})$$

is a $(1 - \alpha)$ confidence region for (P, Q), where $\alpha = 1 - (1 - \alpha^*)^2$.

Rectangular confidence regions

If (P_{low}, P_{up}) and (Q_{low}, Q_{up}) are the $1 - \alpha^*$ univariate confidence intervals for two binomial random variables P and Q, then the rectangle

$$R \equiv (P_{low}, P_{up}) \times (Q_{low}, Q_{up})$$

is a $(1 - \alpha)$ confidence region for (P, Q), where $\alpha = 1 - (1 - \alpha^*)^2$.

Examples:

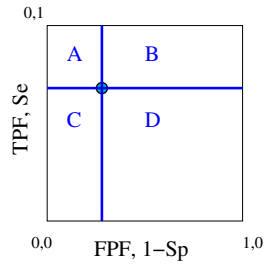
- 95% univariate CI lead to a 90.25% confidence region
- for a 95% confidence region, 97.5% univariate CIs are needed

Outline

1. Performance parameters
   - Introduction
   - Performance parameters for binary classifiers
   - Performance parameters for continuous outputs
2. Performance estimation
3. Performance comparison
4. An example

A motivating example

Using (FPF, TPF) for comparing tests:

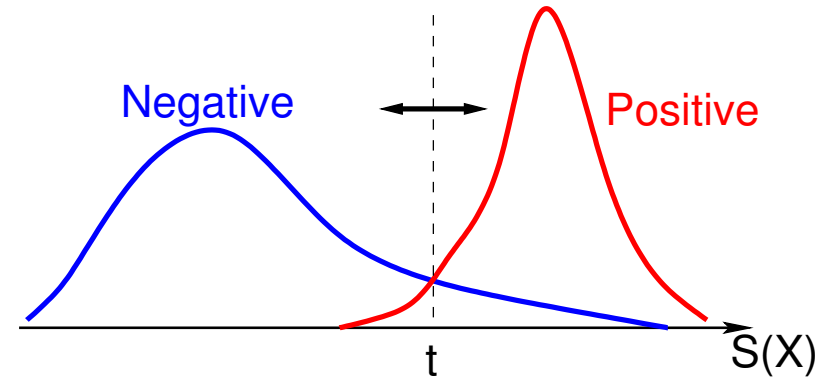


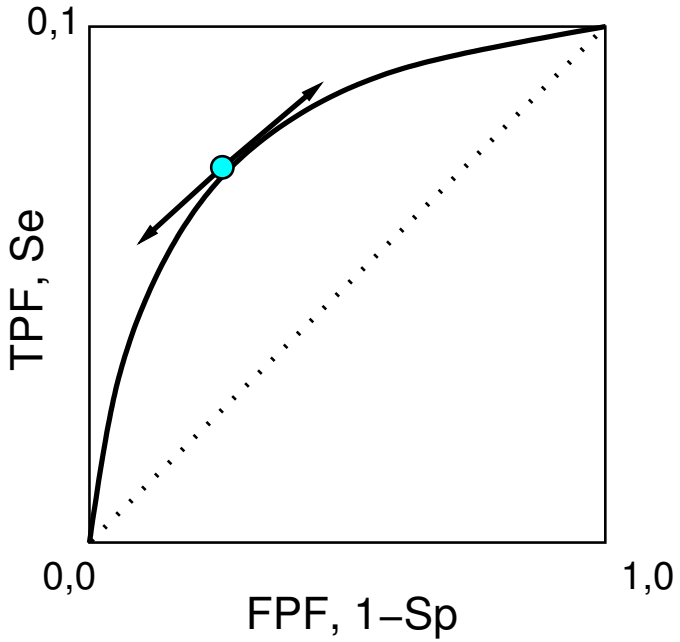
- single point performance measure: partition the space in 4 regions
- region A: better than current test
- region D: worse than current test
- regions B,C: less clear

Other issues with single point performance metrics:

- difficulty in selecting the optimal threshold: different context may need different *operating regimes*
- additive batch effects may spoil the single–point performance

ROC curves: Theory





- continuous test score $Y = S(\mathbf{x})$ (could be margin $h(\mathbf{x})$)
- $\text{FPF}(t) = P[y \geq t | C = 0]$
- $\text{TPF}(t) = P[Y \geq t | C = 1]$
- $\text{ROC} = \{(\text{FPF}(t), \text{TPF}(t)) | \forall t \in \mathbb{R}\}$
- $\lim_{t \to \infty} \text{FPF}(t) = \lim_{t \to \infty} \text{TPF}(t) = 0$
- $\lim_{t \to -\infty} \text{FPF}(t) = \lim_{t \to -\infty} \text{TPF}(t) = 1$

Properties of the ROC curves:

- monotone increasing function
- ROC curve is invariant to strictly increasing transformations of the scores $Y = \psi(S(\mathbf{x}))$
- parametric model:

$$\text{ROC} = \left\{(\alpha, \text{TPF}(\text{FPF}^{-1}(\alpha)))|\forall \alpha \in (0,1)\right\}$$

- $\text{ROC}(0) = 0$, $\text{ROC}(1) = 1$, and

$$\frac{\partial \, \text{ROC}(t)}{\partial t} = \frac{f_C(\text{FPF}^{-1}(t))}{f_{\bar{C}}(\text{FPF}^{-1}(t))},$$

  where f_C and $f_{\bar{C}}$ are the probability densities of the scores within diseased and healthy populations, respectively.
- *the* ROC *curve describes the relationship between the two distributions, and is independent of them*

Note that

$$\frac{\partial \mathrm{ROC}(t)}{\partial t} = \frac{P[Y = t|C = 1]}{P[Y = t|C = 0]} = \mathcal{L}R(t)$$

$\rightarrow$ the likelihood ratio at threshold t.

- if $\mathcal{L}R$ is monotonically increasing, then the classification rule of the form $\mathcal{L}R > t$ is optimal
- the ROC curve based on $\mathcal{L}R$ is uniformly above all other curves
- the optimal ROC curve is *concave*; $\Rightarrow$ its slope is a monotone decreasing function

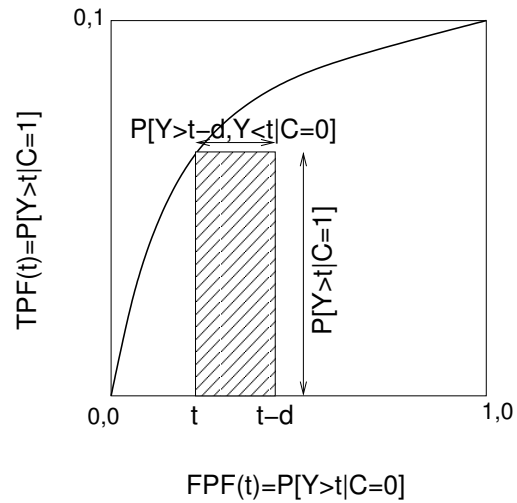
Summary indices

Area under the ROC curve (AUC):

$$\text{AUC} = \int_0^1 \text{ROC}(t)dt$$

Properties:

- $0.5 \leq \text{AUC} \leq 1$
- $\text{AUC} = P[Y_C > Y_{\bar{C}}] \rightarrow$ the probability of correctly ordering a random pair of cases (Mann–Whitney–Wilcoxon U–statistic)



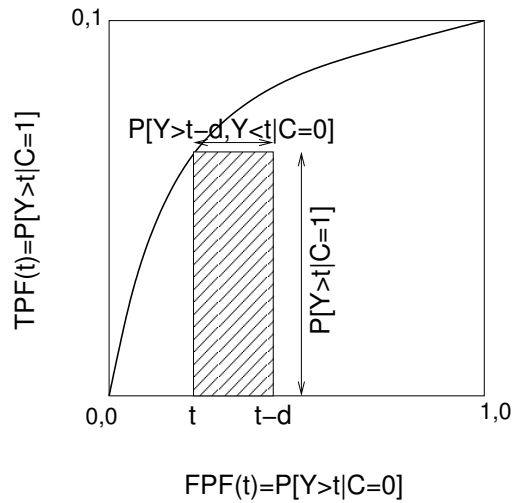
Summary indices

Area under the ROC curve (AUC):

$$\mathrm{AUC} = \int_0^1 \mathrm{ROC}(t)\,dt$$

Properties:

- $0.5 \leq \mathrm{AUC} \leq 1$
- $\mathrm{AUC} = P[Y_C > Y_{\bar{C}}] \rightarrow$ the probability of correctly ordering a random pair of cases (Mann–Whitney–Wilcoxon U–statistic)
- $\mathrm{AUC} = \int_0^1 \mathrm{TPF}(\mathrm{FPF}^{-1}(t))\,dt = -\int_{-\infty}^{\infty} \mathrm{TPF}(t)\,d\,\mathrm{FPF}(t)$



The binormal ROC curve

Assuming normal distributions for the scores:

$$Y_C \sim \mathcal{N}(\mu_C, \sigma_C^2); \qquad Y_{\bar{C}} \sim \mathcal{N}(\mu_{\bar{C}}, \sigma_{\bar{C}}^2),$$

ROC becomes:

$$\mathrm{ROC}(t) = \Phi\left(\frac{\mu_C - \mu_{\bar{C}}}{\sigma_C} + \frac{\sigma_{\bar{C}}}{\sigma_C}\Phi^{-1}(t)\right)$$

The binormal ROC curve

Assuming normal distributions for the scores:

$$Y_C \sim \mathcal{N}(\mu_C, \sigma_C^2); \qquad Y_{\bar{C}} \sim \mathcal{N}(\mu_{\bar{C}}, \sigma_{\bar{C}}^2),$$

ROC becomes:

$$\text{ROC}(t) = \Phi\left(\frac{\mu_C - \mu_{\bar{C}}}{\sigma_C} + \frac{\sigma_{\bar{C}}}{\sigma_C}\Phi^{-1}(t)\right)$$

General form

$$\text{ROC}(t) = \Phi(\alpha + \beta\Phi^{-1}(t))$$

where $\alpha, \beta > 0$ and Φ is the standard normal CDF.

Properties:

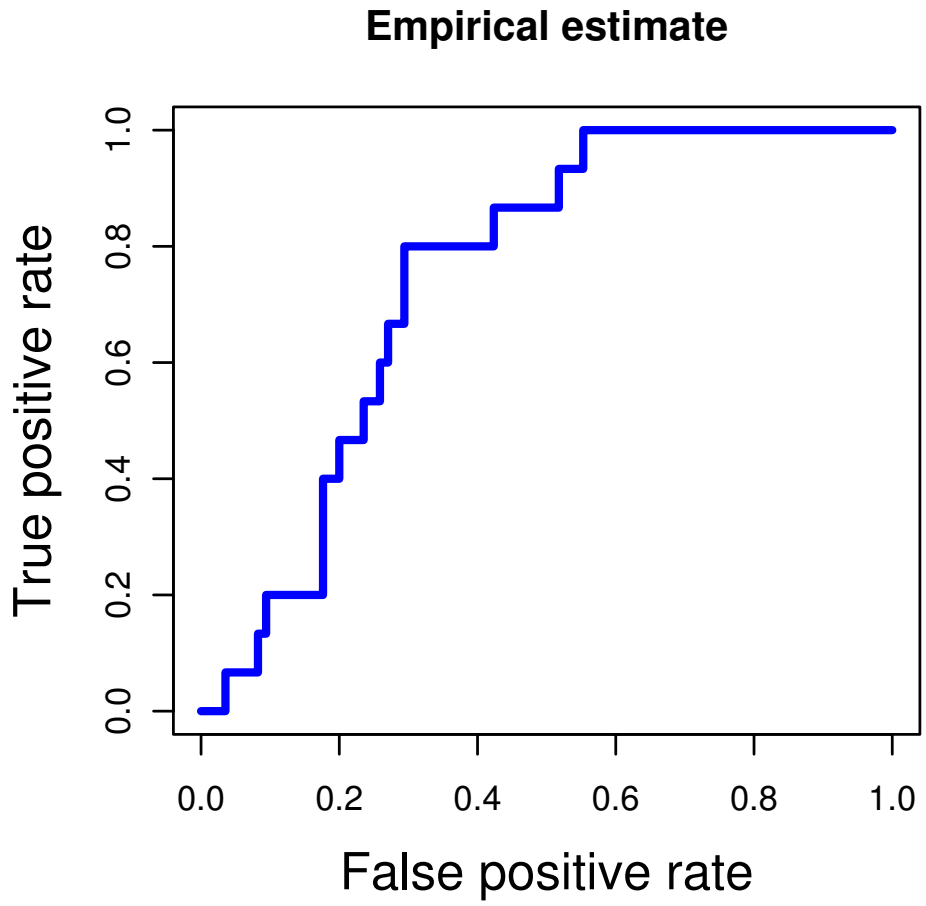
- $\text{AUC} = \Phi\left(\frac{\alpha}{\sqrt{1+\beta^2}}\right)$
- binormal assumption: there exists some monotone strictly increasing function $h(\cdot)$ which makes Y_C and $Y_{\bar{C}}$ normally distributed
- if the ROC is binormal, $\text{ROC}(t) = \Phi(\alpha + \beta\Phi^{-1}(t))$, then $h(s) = -\Phi^{-1}(\text{FPF}(s))$ transforms the scores Y_C and $Y_{\bar{C}}$ into normally distributed random variables.

Empirical estimates of ROC

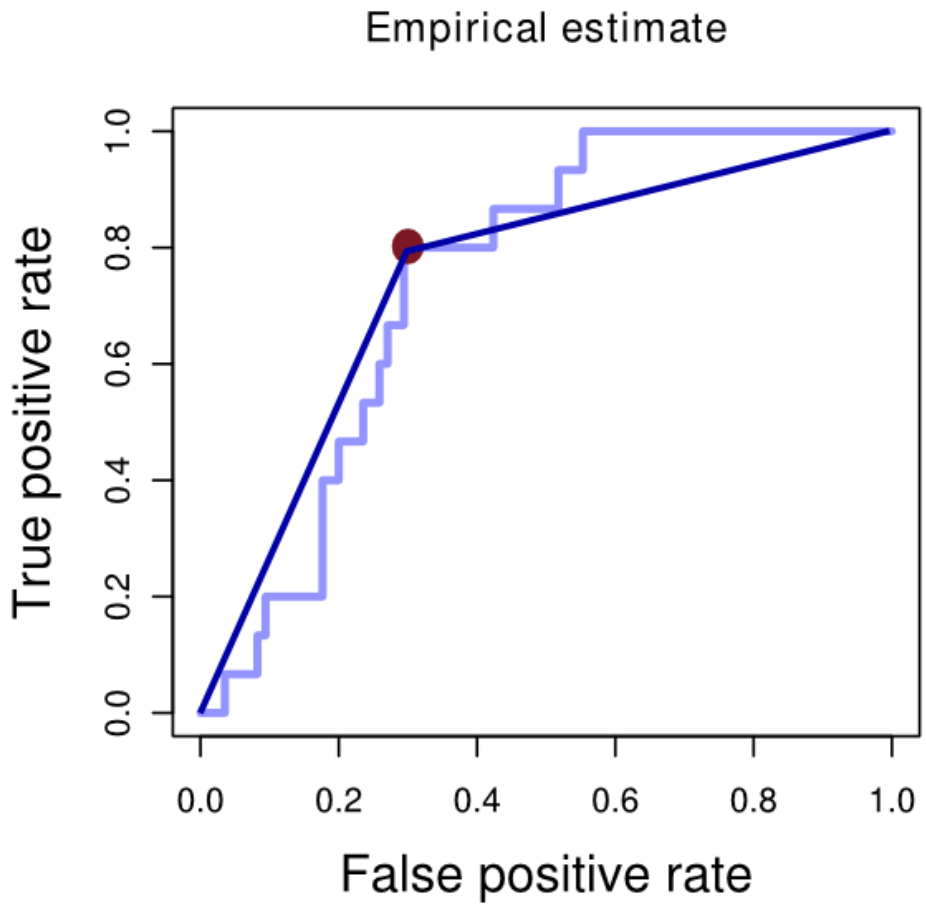
$$\mathrm{ROC}_e(t) = \mathrm{TPF}(\mathrm{FPF}^{-1}(t)) :$$

$$\mathrm{TPF}(t) = \sum_{i=1}^{n_C} \mathbb{I}[Y_{Ci} \geq t]$$

$$\mathrm{FPF}(t) = \sum_{i=1}^{n_{\bar{C}}} \mathbb{I}[Y_{\bar{C}i} \geq t]$$



"ROC" for single threshold



Empirical estimates of AUC

Mann–Whitney–Wilcoxon U–statistic:

$$\mathrm{AUC}_e = \frac{1}{n_C n_{\bar{C}}} \sum_{i=1}^{n_C} \sum_{j=1}^{n_{\bar{C}}} \left(\mathbb{I}[Y_{Ci} > Y_{\bar{C}j}] + 0.5\mathbb{I}[Y_{Ci} = Y_{\bar{C}j}] \right)$$

Note: if only one point in the (FPF, TPF) space is given, $\mathrm{AUC} = 0.5(1 + \mathrm{TPF} - \mathrm{FPF})$.

AUC: sampling variability

$$\mathrm{Var}(\mathrm{AUC}_e) = \frac{1}{n_C n_{\bar{C}}}[\mathrm{AUC}(1-\mathrm{AUC}) + (n_C - 1)(Q_1 - \mathrm{AUC}^2) + (n_{\bar{C}} - 1)(Q_2 - \mathrm{AUC}^2)]$$

where

$$Q_1 = P[Y_{Ci} \geq Y_{\bar{C}j}, Y_{Ck} \geq Y_{\bar{C}j}]$$
$$Q_2 = P[Y_{Ci} \geq Y_{\bar{C}j}, Y_{Ci} \geq Y_{\bar{C}k}].$$

Semi–parametric models

Start from

$$\mathrm{ROC}(t) = \mathrm{TPF}(\mathrm{FPF}^{-1}(t|\alpha)|\beta)$$

and assume some parametric form for TPF and FPF for which estimate the parameters from data.

Ex. of semi–parametric model:

$$Y_{Ci} = \mu_C + \sigma_C \varepsilon_i$$
$$Y_{\bar{C}i} = \mu_{\bar{C}} + \sigma_{\bar{C}} \varepsilon_i$$

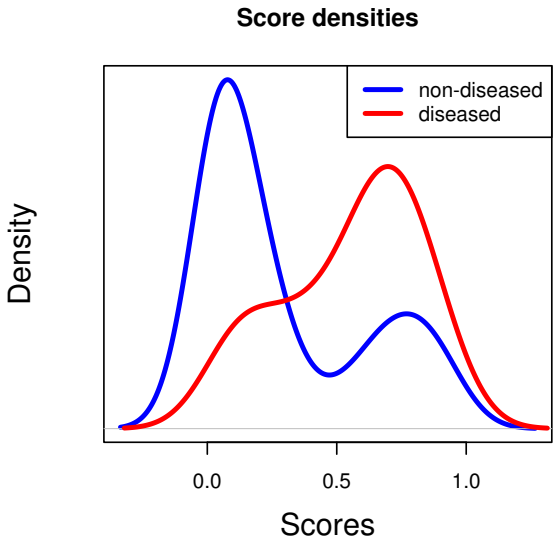
where ε have mean 0 and variance 1 and follow some distribution function S.

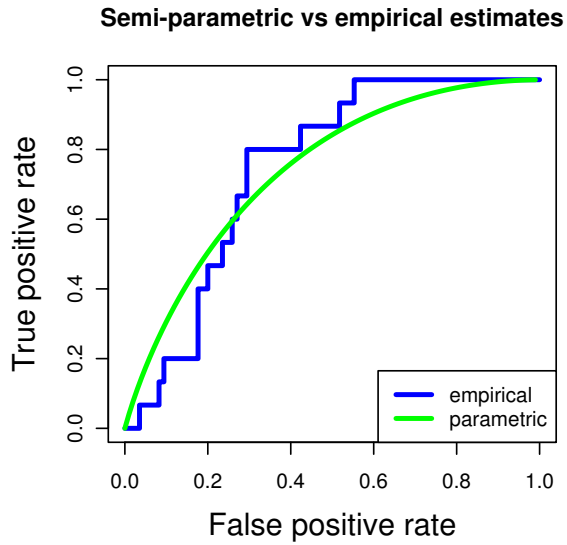
$$S(t) = \frac{1}{n_C + n_{\bar{C}}} \left\{ \sum_i \mathbb{I}\left[\frac{Y_{Ci} - \mu_C}{\sigma_C} \geq t \right] + \sum_i \mathbb{I}\left[\frac{Y_{\bar{C}i} - \mu_{\bar{C}}}{\sigma_{\bar{C}}} \geq t \right] \right\}$$

which leads to

$$\text{ROC}(t) = S\left((\mu_{\bar{C}} - \mu_C)/\sigma_C + (\sigma_{\bar{C}}/\sigma_C) S^{-1}(t) \right)$$

Ex: empirical vs. semi-parametric estimation





$$\mathrm{AUC}_e \approx 0.7475; \quad \mathrm{AUC}_{sp} \approx 0.7418$$

Outline

1. Performance parameters
   - Introduction
   - Performance parameters for binary classifiers
   - Performance parameters for continuous outputs
2. Performance estimation
3. Performance comparison
4. An example

Why estimation?

- finite training data
- no formula for CI without distribution assumptions
- often, a single data set is available for both model building and performance measuring
- performance estimated on the modeling data is optimistically biased

Idea

Split (maybe repeatedly) the available data into a training and a validation set, and assess the performance only on the data that has not been used in building the model.

WARNING

All the processing steps that depend on the sampling and which lead to the final model, MUST BE REPEATED IDENTICALLY ON EVERY TRAIN-VALIDATION SPLIT!

This includes, but is not limited to: data normalization, feature selection, classifier training, meta-parameter optimization.

Notes:

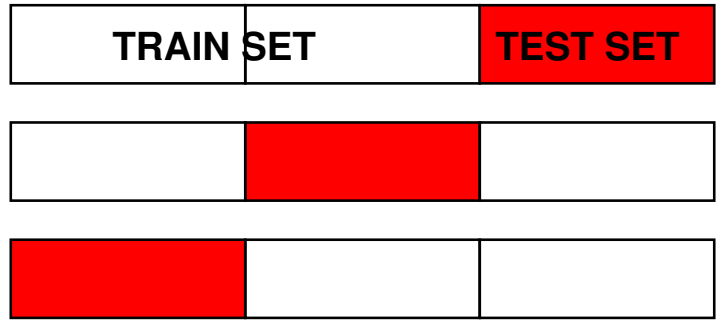
- any two training sets generated from the full data set by resampling will usually overlap to some extent → the models are not totally independent
- the variability is usually under-estimated
- the procedure is easy to be parallelized, but attention must be paid to the parallel RNG (to avoid repeating the same sequences)

Resampling methods

- simple split–sample approach
- k–fold cross–validation
- Monte–Carlo cross-validation
- repeated k–fold cross–validation
- leave–one–out
- bootstrapping
- ...

k–fold cross–validation

- separated train and test sets
- randomly dived data into k subsets (folds) – you may also choose to enforce the proportion of the classes (stratified CV)
- train on k – 1 folds and test on the holdout fold
- estimate the error as the average error measured on holdout folds



- usually $k = 5$ or $k = 10$
- if $k = n \Rightarrow$ leave–one–out estimator
- improved estimation: repeated k–CV (e.g. $100 \times (5 - CV)$)

k–fold cross–validation

From k folds:

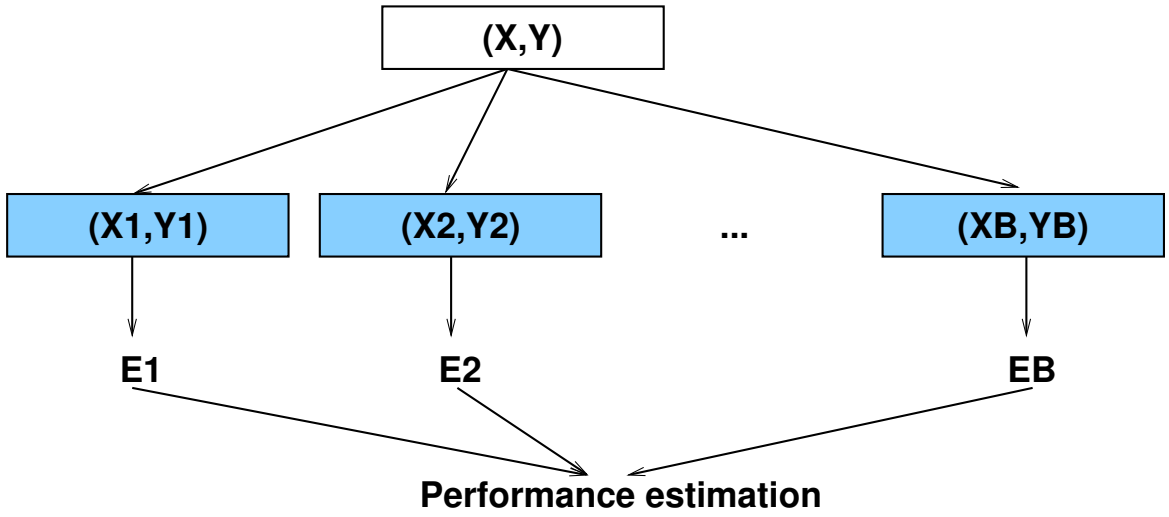
- $\epsilon_1, \ldots, \epsilon_k$ errors on the test folds (*any other performance parameter*)
- $\hat{E}_{k-CV} = \frac{1}{k} \sum_{j=1}^{k} \epsilon_j$
- estimated standard deviation

Confidence intervals (simple version – normal approximation):

$$E \approx \hat{E} \pm \left(\frac{0.5}{n} + z \sqrt{\frac{\hat{E}(1 - \hat{E})}{n}} \right)$$

where n is the dataset size and $z = \Phi^{-1}(1 - \alpha/2)$, for a $1 - \alpha$ confidence interval (e.g. $z = 1.96$ for 95% conf. interval)

Bootstrap error estimation



1. generate a new dataset (X_b, Y_b) by *resampling with replacement* from the original dataset (X, Y)
2. train the classifier on (X_b, Y_b) and test on the left out data, to obtain an error $\hat{E}_b$.
3. repeat 1–2 for $b = 1, \ldots, B$ and collect $\hat{E}_b$.

Bootstrap error estimation

- estimate the error: for example, use the *.632 estimator*

$$\hat{E} = 0.368 E_0 + 0.632 \frac{1}{B} \sum_{b=1}^{B} \hat{E}_b$$

  where E_0 is the error rate on the full training set (X, Y).
- use the empirical distribution of $\hat{E}_b$ to obtain confidence intervals

LPO bootstrap

Classification rule:

$$\hat{h}(\mathbf{x}) \underset{\bar{C}}{\overset{C}{\gtrless}} \theta$$

where $\hat{h}$ is the estimated log-likelihood ratio and C and $\bar{C}$ are the class labels.
Empirical AUC (conditioned on the training set) can be approximated by:

$$\widehat{\mathrm{AUC}} = \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \psi\left(\hat{h}(\mathbf{x}_i|C), \hat{h}(\mathbf{x}_j|\bar{C})\right)$$

where ψ is the Mann-Whitney kernel,

$$\psi(a,b) = \begin{cases} 1 & a > b \\ \frac{1}{2} & a = b \\ 0 & a < b \end{cases}$$

Yousef et al., Estimating the uncertainty in the estimated mean area under the ROC curve of a classifier,

PRL 2005

Estimation of the *expected* AUC by LPO bootstrap:

$$\widehat{\mathrm{AUC}}^{LPO} = \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \widehat{\mathrm{AUC}}_{i,j}$$

$$\widehat{\mathrm{AUC}}_{i,j} = \frac{\sum_{b=1}^{B} I_j^b I_i^b \psi(\hat{h}_b(\mathbf{x}_i), \hat{h}_b(\mathbf{x}_j))}{\sum_{b=1}^{B} I_j^b I_l^b}$$

When 2 independent data sets are available, one can estimate:

- the expected value of the conditional AUC: expectation over the population of training sets *of the same size*;
- variability of the performance estimate due to finite train set;
- variability of the performance estimate due to finite validation sets;

Yousef et al., Assessing classifiers from two independent data sets using ROC analysis: a nonparametric approach, PAMI 2006

Conclusions

What we do learn from CV (and related):

- the expected performance of the modeling recipe;
- the imprecision in estimating the performance;
- we can have a look at:
  - what are the most stable features
  - what are the points always missclassified

Conclusions

What we do learn from CV (and related):

- the expected performance of the modeling recipe;
- the imprecision in estimating the performance;
- we can have a look at:
  - what are the most stable features
  - what are the points always missclassified

What we do not learn from CV:

- the best features
- the best classifier
- the best meta–parameters

We obtain these by training on the full dataset (no CV).

Outline

1. Performance parameters
   - Introduction
   - Performance parameters for binary classifiers
   - Performance parameters for continuous outputs
2. Performance estimation
3. Performance comparison
4. An example

General considerations:

- comparison of methods/algorithms or models?
- let there be two models M_1 and M_2 and a performance parameter P
- what differences are relevant?
- proper planning of the experimental design
- hypothesis testing (equivalence/difference and inferiority/superiority):
  H_0 : there is no difference in performance
  $P(M_1) = P(M_2)$
  $H_1 : P(M_1) \neq P(M_2)$ (two sided test) or
  $H_1 : P(M_1) \gtrless P(M_2)$ (single sided or inferiority/superiority test)
- informally, one can check the overlap between CIs
- ideally, one would have a very large test set for comparison

In everyday applications...

- one has limited data → use the resampling (like cross-validation) for testing
- let $P_{11}, \ldots, P_{1K}$ be the performance of the 1st model on the K test sets and $P_{21}, \ldots, P_{2K}$ the performance of the 2nd model on the same K test sets
- simple tests: paired t–test and Wilcoxon signed rank test
- warning: variability is underestimated,hence t-test has inflated Type I error; there is a "corrected t-test" to alleviate the problem
- the two samples $\{P_{1.}\}$ and $\{P_{2.}\}$ are not independent!

McNemar's test

- consider a single test set of size m, on which both models are applied
- the following contingency table is constructed:

		Model M_2 0	Model M_2 1
Model M_1	0	c_{00}	c_{01}
	1	c_{10}	c_{11}

with

- c_{00} counting how many times both models misclassified the same sample
- c_{11} counting how many times both models correctly classified the same sample
- c_{10} and c_{01} counting how many times M_1 correctly classified a sample the M_2 misclassified, and vice-versa

- McNemar's test: H_0 both classifiers have the same performance (same error rates)
- construct the test statistic

$$\chi^2_{Mc} = \frac{(|c_{01} - c_{10}| - 1)^2}{c_{01} + c_{10}}$$

- χ^2_{Mc} has an approximate χ^2 distribution with 1 df
- χ^2_{Mc} is to be compared with $\chi^2_{1,1-\alpha}$ values for $1 - \alpha$ significance level
- rule-of-thumb: the test needs a sample size large enough such that $c_{01} + c_{10}$ is at least 30

Wrap-up

- many performance parameters, depend on the intended usage
- performance estimation is a key step of classifier building process
- pay attention of proper application of resampling methods for performance estimation
- always (ALWAYS!) report the uncertainty in the estimates
- classifier performance comparison depends, again, on the intended application
- McNemar's test and CIs provide indications on performance differences

Outline

1. Performance parameters
   - Introduction
   - Performance parameters for binary classifiers
   - Performance parameters for continuous outputs
2. Performance estimation
3. Performance comparison
4. An example

ARTICLES

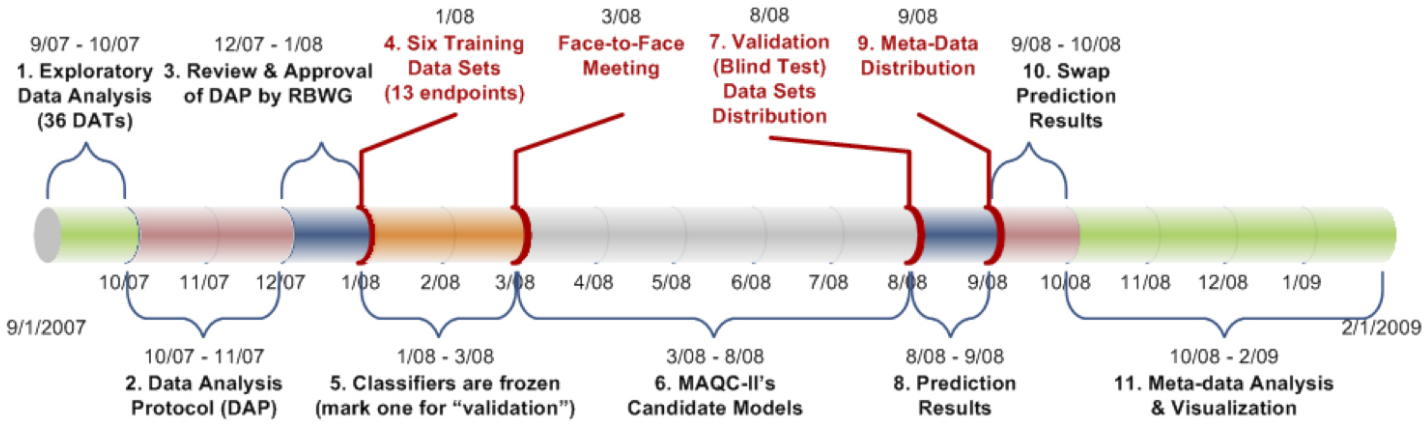
nature biotechnology

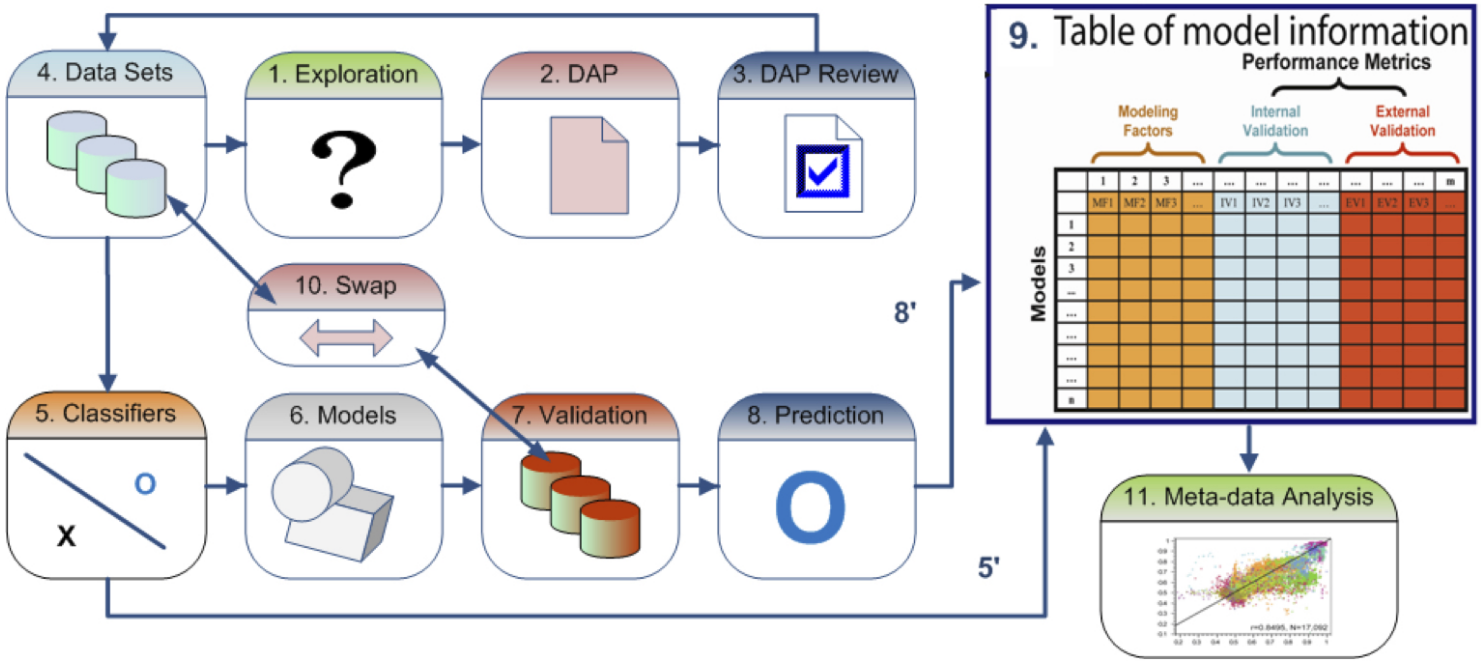
The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models

MAQC Consortium*

MAQC-II:

- $\sim$ 300 participants, from 5 countries:
  - data providers
  - data analysis teams (DATs): 36 teams, ($\sim$ 100 people)
  - regulatory board (mainly FDA)
- 6 datasets, 13 endpoints
- $>$ 30000 "models"
- each Data Analysis Plan (DAP) is peer-reviewed
- each DAT selects a single candidate model for each endpoint
- MAQC-II consortium selects 2 models for each endpoint, before the release of the validation sets


9/07 - 10/07
1. Exploratory Data Analysis (36 DATs)
12/07 - 1/08
3. Review & Approval of DAP by RBWG
1/08
4. Six Training Data Sets (13 endpoints)
3/08
Face-to-Face Meeting
8/08
7. Validation (Blind Test) Data Sets Distribution
9/08
9. Meta-Data Distribution
9/08 - 10/08
10. Swap Prediction Results
10/07
11/07
12/07
1/08
2/08
3/08
4/08
5/08
6/08
7/08
8/08
9/08
10/08
11/08
12/08
1/09
9/1/2007
2/1/2009
10/07 - 11/07
2. Data Analysis Protocol (DAP)
1/08 - 3/08
5. Classifiers are frozen (mark one for "validation")
3/08 - 8/08
6. MAQC-II's Candidate Models
8/08 - 9/08
8. Prediction Results
10/08 - 2/09
11. Meta-data Analysis & Visualization


4. Data Sets
1. Exploration
2. DAP
3. DAP Review
10. Swap
8'
5. Classifiers
6. Models
7. Validation
8. Prediction
5'
9. Table of model information
Performance Metrics
Modeling Factors
Internal Validation
External Validation
Models
11. Meta-data Analysis

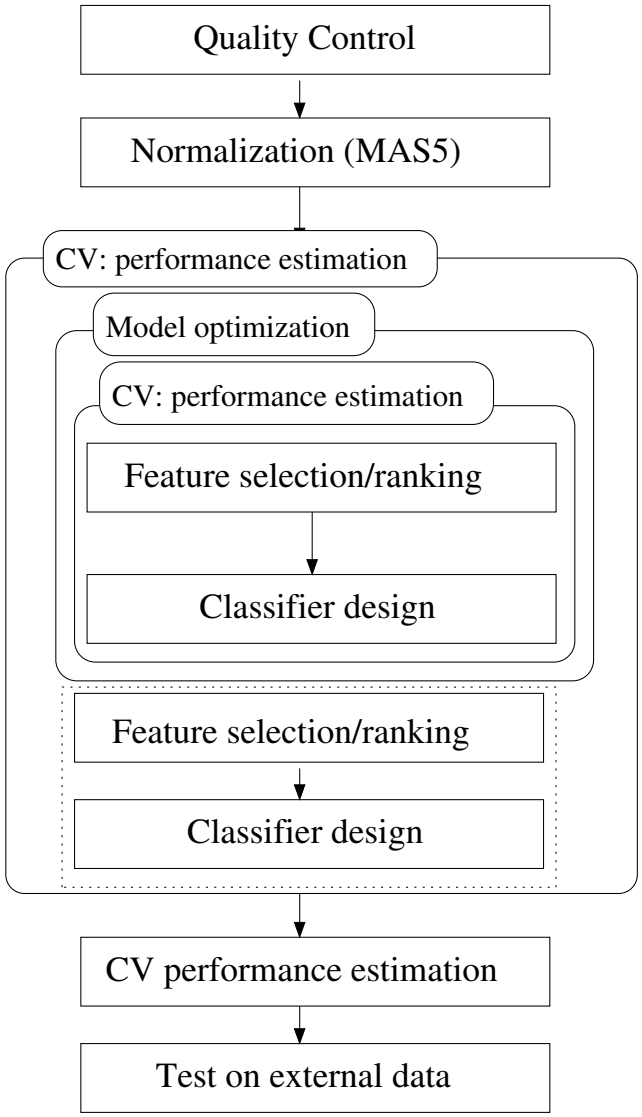
DAP

Constraints:

- should be generally applicable, independent on dataset/endpoint
- trade-off: undestandability/reproducibility vs. performance/complexity
- the models should make single-chip predictions

Solution:

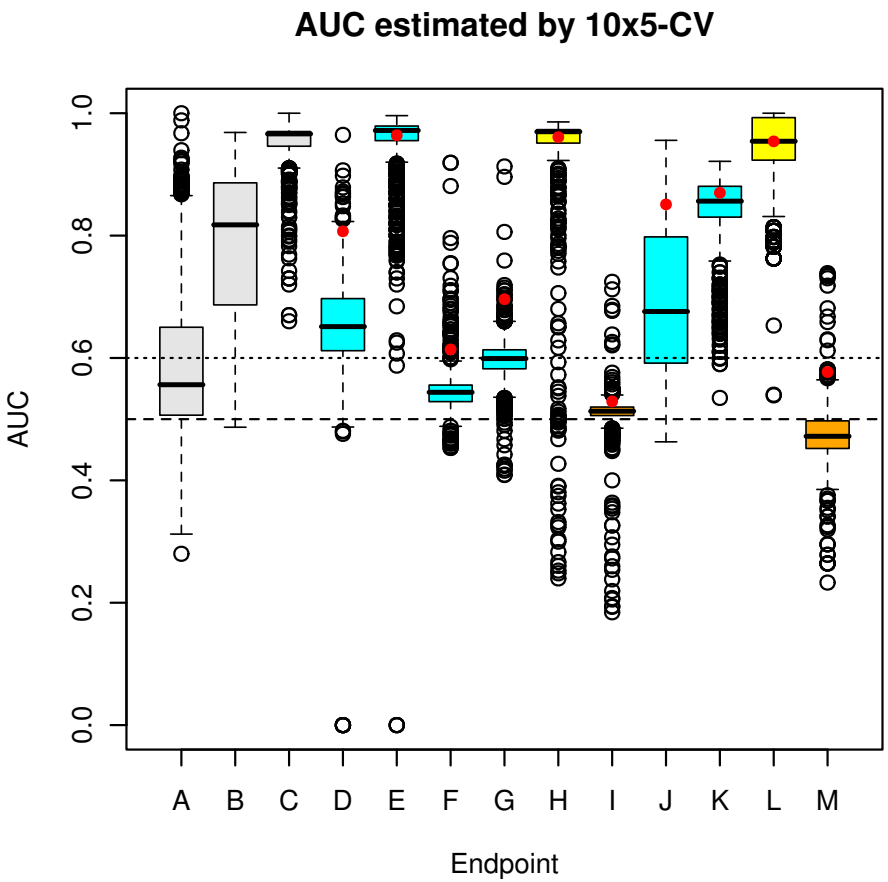
- use MAS5 for normalization
- favor "simple" classifiers
- nested $10 \times 5 - CV$
- use AUC as main performance criterion

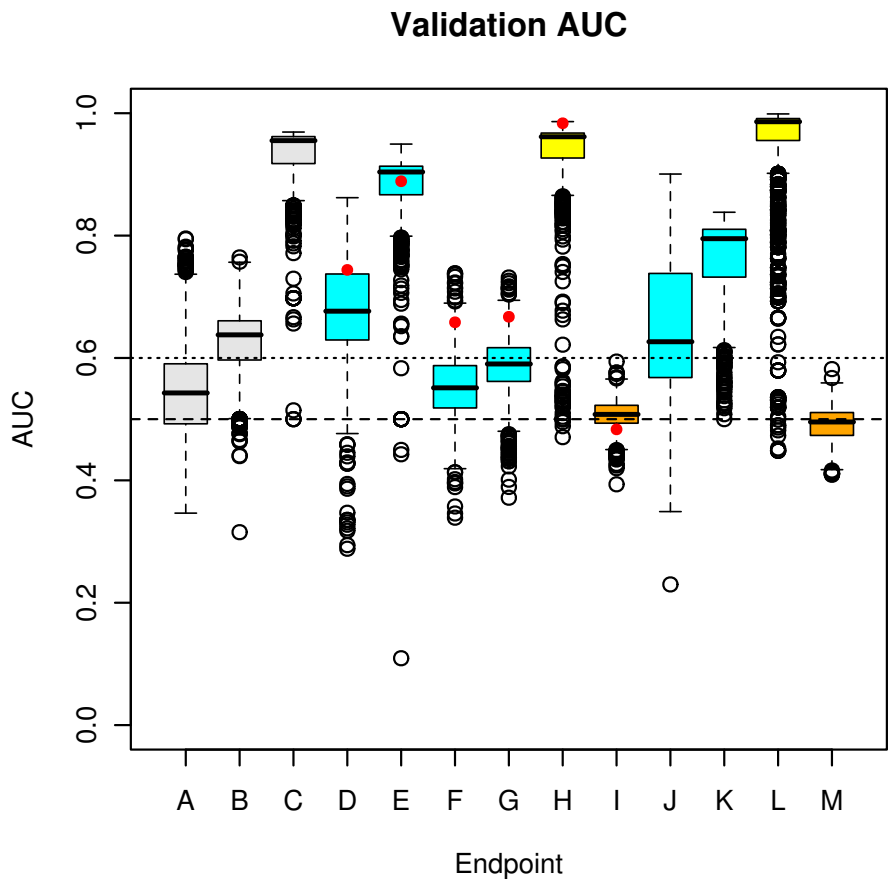


- classifiers: DLDA, LDA, k–NN, CART, logistic regression
- meta-parameters: number of features, k, ...
- inner CV: optimize the meta-parameter
- outer CV: estimate the performance of the system

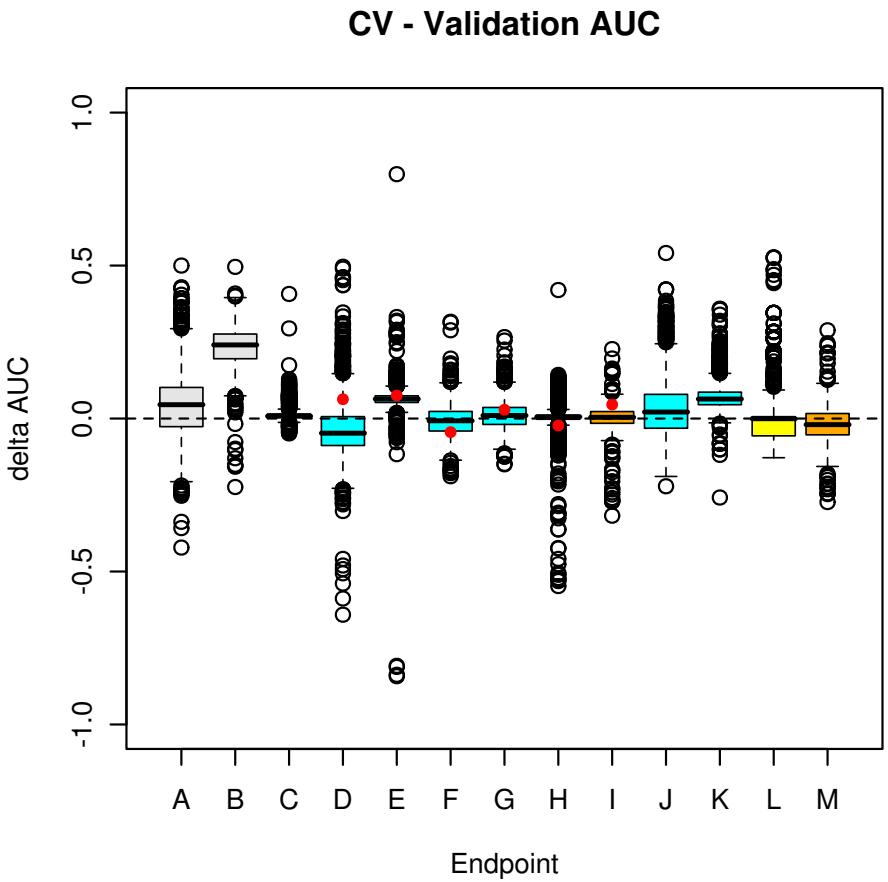
Some performance results

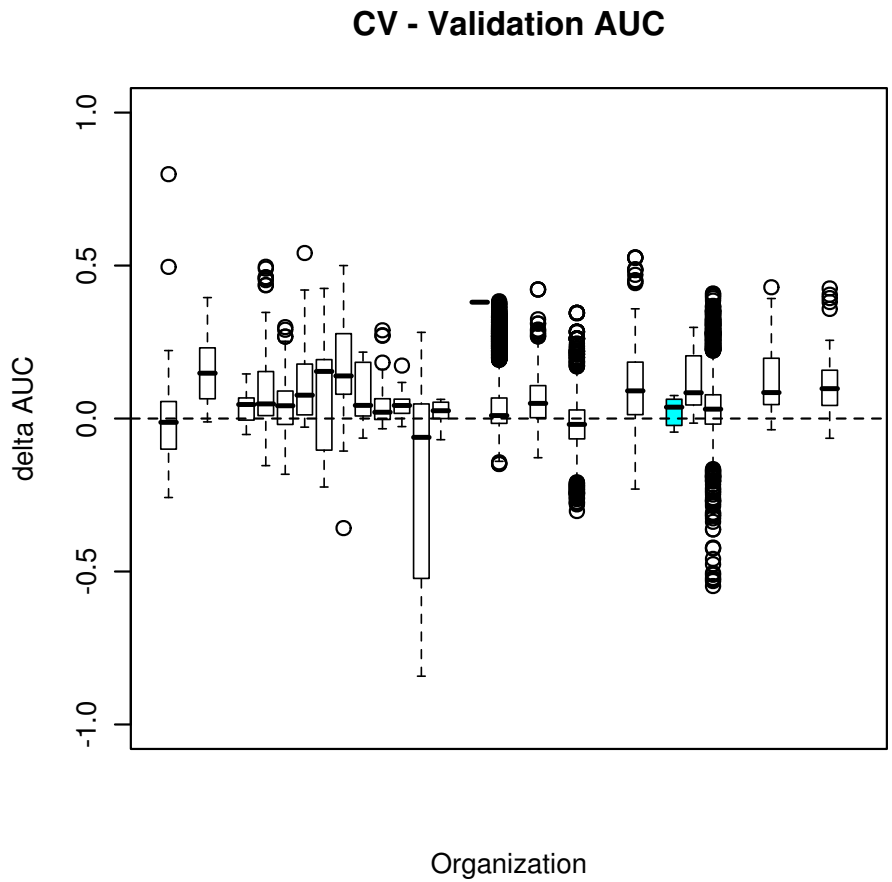
Estimated vs. validation performance (AUC)

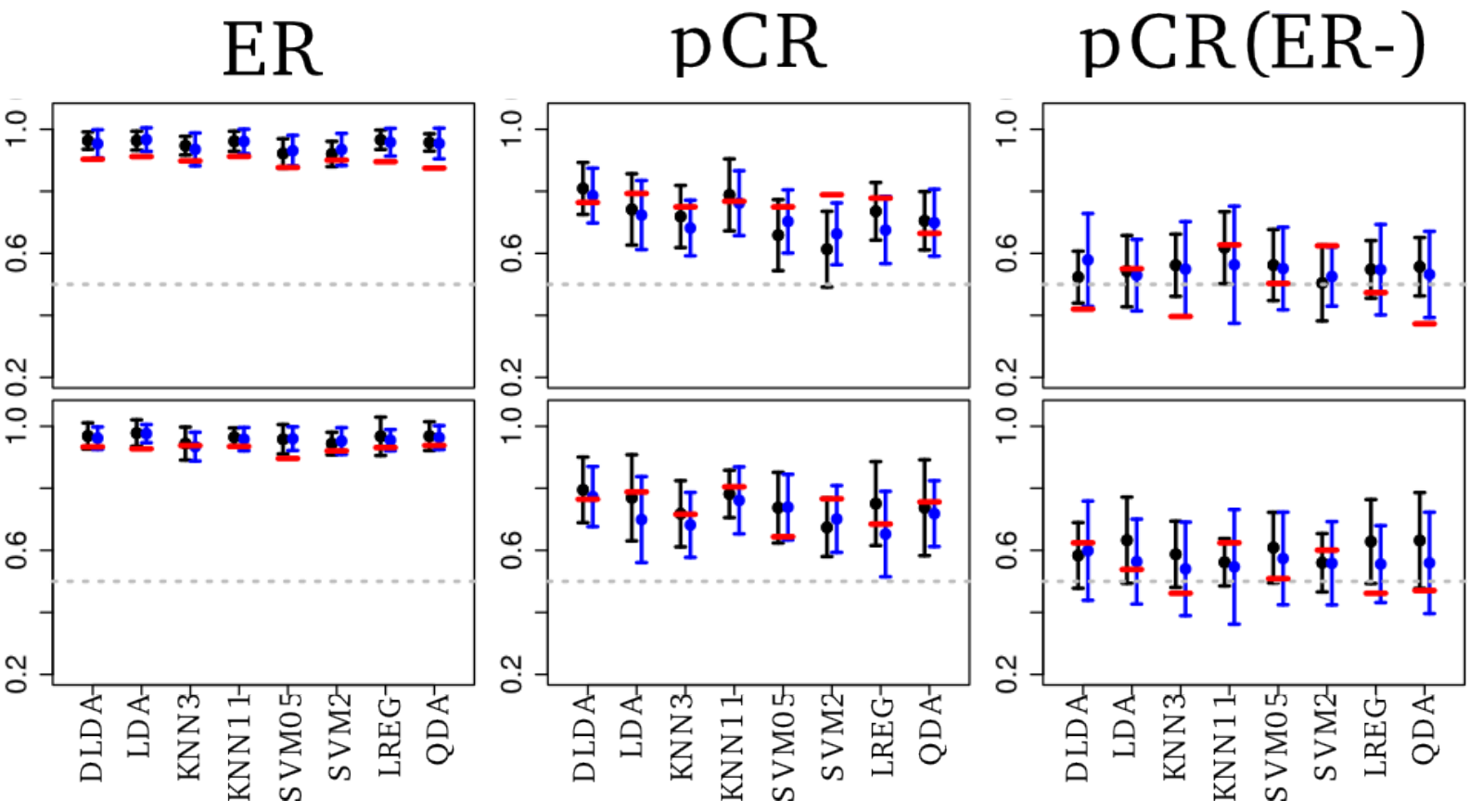




Estimation bias






ER
pCR
pCR(ER-)
1.0
0.6
0.2
DLDA
LDA
KNN3
KNN11
SVM05
SVM2
LREG
QDA