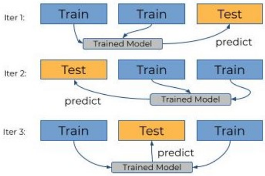
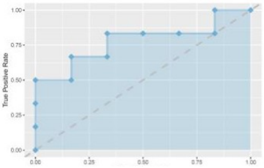
Introduction to Machine Learning

Evaluation: Introduction and Remarks





Learning goals

- Understand the goal of performance estimation
- Know the definition of generalization error
- Understand the difference between outer and inner loss

EXAMPLE PRACTICAL METHOD

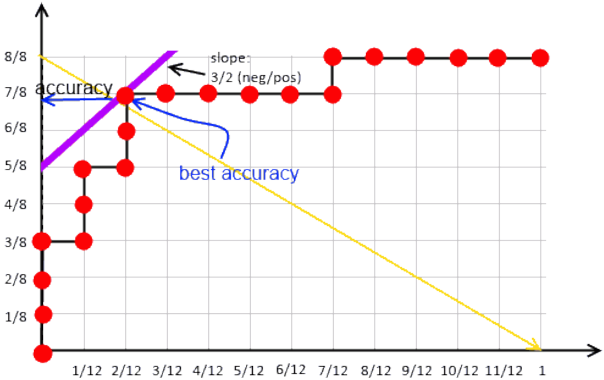
Given: 20 training observations, 12 negative and 8 positive

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
C	N	N	N	N	N	N	N	N	N	N	N	N	P	P	P	P	P	P	P	P
Score	.18	.24	.32	.33	.4	.53	.58	.59	.6	.7	.75	.85	.52	.72	.73	.79	.82	.88	.9	.92

$\Rightarrow$ sort by score and draw the curves:

#	20	19	18	12	17	16	11	15	14	10	9	8	7	6	13	5	4	3	2	1
C	P	P	P	N	P	P	N	P	P	N	N	N	N	N	P	N	N	N	N	N
Score	.92	.9	.88	.85	.82	.79	.75	.73	.72	.7	.6	.59	.58	.53	.52	.4	.33	.32	.24	.18

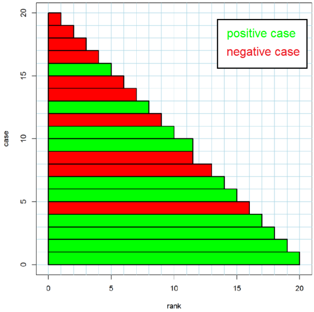
EXAMPLE PRACTICAL METHOD

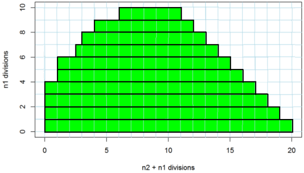


- Best accuracy achieved with observation # 18.
- Setting $\theta = 0.88 \Rightarrow$ accuracy of $15/20 \hat{=}$ 75%.

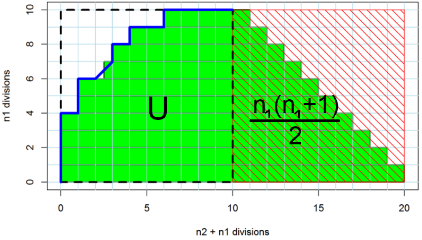
EXPLANATION MANN-WHITNEY-U TEST

- First we plot the ranks of all the scores as a stack of horizontal bars, and color them by the labels.
- Stack the green bars on top of one another, and slide them horizontally as needed to get a nice even stairstep on the right edge (See: practical method example for ROC curves):



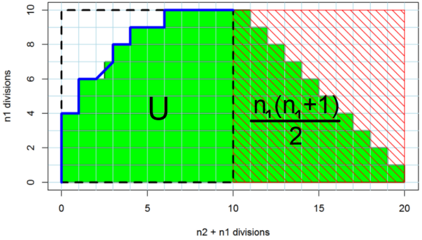


EXPLANATION MANN-WHITNEY-U TEST



- Definition of the U statistic: $U = R_1 - \dfrac{n_1(n_1+1)}{2}$
  - R_1 is the sum of ranks of positive cases (the area of the green bars)
  - n_1 is the number of positive cases
- The area of the green bars on the right side is equal to $\dfrac{n_1(n_1+1)}{2}$.

EXPLANATION MANN-WHITNEY-U TEST



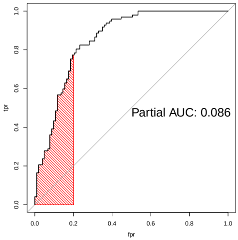
- $U =$ area of the green bars on left side
- area of dashed rectangle = $n_1 \cdot n_2$
- *AUC* is U normalized to the unit square,

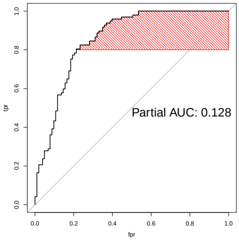
$$\Longrightarrow AUC = \frac{U}{n_1 \cdot n_2}$$

with $n_1 = \text{POS}$ and $n_2 = \text{NEG}$.

PARTIAL AUC

- Sometimes it can be useful to look at a specific region under the ROC curve $\Rightarrow$ partial AUC (pAUC).
- Let $0 \leq c_1 < c_2 \leq 1$ define a region.
- For example, one could focus on a region with low fpr ($c_1 = 0, c_2 = 0.2$) or a region with high tpr ($c_1 = 0.8, c_2 = 1$):





PARTIAL AUC

- pAUC $\in [0, c_2 - c_1]$.
- The partial AUC can be corrected (see McClish), to have values between 0 and 1, where 0.5 is non discriminant and 1 is maximal:

$$\text{pAUC}_{\text{corrected}} = \frac{1 + \frac{\text{pAUC} - \text{min}}{\text{max} - \text{min}}}{2}$$

- min is the value of the non-discriminant AUC in the region
- max is the maximum possible AUC in the region

MULTICLASS AUC

- Consider multiclass classification, where a classifier predicts the probability p_k of belonging to class k for each class.
- Hand and Till (2001) proposed to average the AUC of pairwise comparisons (1 vs. 1) of a multiclass classifier.
  - estimate $AUC(i, j)$ for each pair of class i and j
  - $AUC(i, j)$ is the probability that a randomly drawn member of class i has a lower probability of belonging to class j than a randomly drawn member of class j.
  - for K classes, we have $\binom{K}{2} = \frac{K(K-1)}{2}$ values of $AUC(i, j)$ that are then averaged to compute the Multiclass AUC.

CALIBRATION AND DISCRIMINATION

We consider data with a binary outcome y.

- **Calibration:** When the predicted probabilities closely agree with the observed outcome (for any reasonable grouping).
  - **Calibration in the large** is a property of the *full sample*. It compares the observed probability in the full sample (e.g. proportion of observations for which $y = 1$) with the average predicted probability in the full sample.
  - **Calibration in the small** is a property of *subsets* of the sample. It compares the observed probability in each subset with the average predicted probability in that subset.
- **Discrimination:** Ability to perfectly separate the population into $y = 0$ and $y = 1$. Measures of discrimination are, for example, AUC, sensitivity, specificity.

CALIBRATION AND DISCRIMINATION

A well calibrated classifier can be poorly discriminating, e.g.

Obs. Nr.	truth	Pred Rule 1	Pred Rule 2
1	1	1	0
2	1	1	0
3	0	0	1
4	0	0	1
Avg Prob	50%	50%	50%

- Both prediction rules have identical calibration in the large (50%), however, rule 1 is better than rule 2.

CALIBRATION AND DISCRIMINATION

A well discriminating classifier can have a bad calibration, e.g.

Obs. Nr.	truth	Pred Rule 1	Pred Rule 2
1	1	0.9	0.9
2	1	0.9	0.9
3	0	0.1	0.7
4	0	0.1	0.7
Avg Prob	50%	50%	80%

- Both prediction rules are well discriminating (e.g., setting thresholds $\theta_1 = 0.5$, $\theta_2 = 0.8$)
- Prediction rule 2 is rather poorly calibrated. The proportion of observations for which $y = 1$ would be estimated with 80%.

ROC ANALYSIS IN R

- `generateThreshVsPerfData` calculates one or several performance measures for a sequence of decision thresholds from 0 to 1.
- It provides S3 methods for objects of class `Prediction`, `ResampleResult` and `BenchmarkResult` (resulting from `predict.WrappedModel`, `resample` or `benchmark`).
- `plotROCCurves` plots the result of `generateThreshVsPerfData` using `ggplot2`.
- More infos `http://mlr-org.github.io/mlr-tutorial/release/html/roc_analysis/index.html`

EXAMPLE 1: SINGLE PREDICTIONS

small code chunk

EXAMPLE 1: SINGLE PREDICTIONS

We calculate fpr, tpr and compute error rates:

one line of code

- `generateThreshVsPerfData` returns an object of class `ThreshVsPerfData`, which contains the performance values in the `$data` slot.
- By default, `plotROCCurves` plots the performance values of the first two measures passed to `generateThreshVsPerfData`.
- The first is shown on the x-axis, the second on the y-axis.

EXAMPLE 1: SINGLE PREDICTIONS

one line of code + figure

EXAMPLE 1: SINGLE PREDICTIONS

The corresponding area under curve auc can be calculated by
one line of code

`plotROCCurves` always requires a pair of performance measures that are plotted against each other.

EXAMPLE 1: SINGLE PREDICTIONS

If you want to plot individual measures vs. the decision threshold, use

one line of code + figure

EXAMPLE 2: BENCHMARK EXPERIMENT

small code chunk

Calling `generateThreshVsPerfData` and `plotROCCurves` on the `BenchmarkResult` produces a plot with ROC curves for all learners in the experiment.

EXAMPLE 2: BENCHMARK EXPERIMENT

one line of code + figure

EXAMPLE 2: BENCHMARK EXPERIMENT

one line of code + figure