Comparing binary classification rules. ROC curve and other methods

Pedro Delicado
Departament d'Estadística i Investigació Operativa
Universitat Politècnica de Catalunya

REFERENCES:

Sections 9.2.5 in Hastie, Tibshirani, and Friedman (2009)

Sections 4.4.3 and 9.6.3 in James, Witten, Hastie, and Tibshirani (2013)

The binary classification problem, or binary discrimation

- Let $(\boldsymbol{X}, Y)$ be a r.v. with support $\mathcal{X} \times \{0,1\} \subseteq \mathbb{R}^p \times \{0,1\}$.
- Binary classification problem, or binary prediction, or binary disrimination:
  - Training sample: $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, i.i.d. from $(\boldsymbol{X}, Y)$.
  - The goal is to define a classification function (or discrimination rule) (depending on the sample)

$$h_S : \mathcal{X} \mapsto \{0,1\}$$

such that for a new independent observation $(\boldsymbol{x}_{n+1}, y_{n+1})$, from which we only know $\boldsymbol{x}_{n+1}$, it happens that

$$\Pr(h_S(\boldsymbol{x}_{n+1}) = y_{n+1}) \text{ is close to } 1.$$

A more general approach to the binary classification

- A more general goal in binary classification is to define a function $g_S : \mathcal{X} \mapsto [0,1]$ such that for a new independent observation $(\boldsymbol{x}_{n+1}, y_{n+1})$, from which we only know $\boldsymbol{x}_{n+1}$, $g_S(\boldsymbol{x}_{n+1})$ is close to y_{n+1} (in some sense).
- Observe that from such a function it is possible to define a classification function:
  - Let $c \in [0,1]$ be a cut point.
  - From $g_S : \mathcal{X} \mapsto [0,1]$ and c, define $h_S : \mathcal{X} \mapsto \{0,1\}$ as

$$h_S(\boldsymbol{x}) = \begin{cases} 0 & \text{if} \quad g_S(\boldsymbol{x}) \leq c, \\ 1 & \text{if} \quad g_S(\boldsymbol{x}) > c. \end{cases}$$

An example: Estimating $\Pr(Y = 1|\boldsymbol{X} = \boldsymbol{x})$

- Let $(\boldsymbol{X}, Y)$ be a r.v. with support $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^p \times \{0, 1\}$.
- Let $(\boldsymbol{X}_1, Y_1), \dots, (\boldsymbol{X}_n, Y_n)$ i.i.d.r.v. distributed as $(\boldsymbol{X}, Y)$.
- Sample: $(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)$, realizations of $(\boldsymbol{X}_1, Y_1), \dots, (\boldsymbol{X}_n, Y_n)$.
- Conditioning on $(\boldsymbol{X}_1 = \boldsymbol{x}_1, \dots, \boldsymbol{X}_n = \boldsymbol{x}_n)$, $Y_1, \dots, Y_n$ are independent random variables, each with distribution

$$(Y_i \mid \boldsymbol{X}_i = \boldsymbol{x}_i) \sim \text{Bernoulli}(p_i = \Pr(Y = 1 \mid \boldsymbol{X}_i = \boldsymbol{x}_i) = E(Y \mid \boldsymbol{X}_i = \boldsymbol{x}_i)).$$

- Let $p(\boldsymbol{x}) = \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$.
- Given an estimation $\hat{p}(\boldsymbol{x})$ of the function $p(\boldsymbol{x})$, a classification rule $h_S : \mathcal{X} \mapsto \mathcal{Y}$ can be defined as

$$h_S(\boldsymbol{x}_{n+1}) = \begin{cases} 0 & \text{if} \quad \hat{p}(\boldsymbol{x}_{n+1}) \leq 1/2 \\ 1 & \text{if} \quad \hat{p}(\boldsymbol{x}_{n+1}) > 1/2 \end{cases}$$

- It is possible to use other cut point c different from $1/2$.

Evaluating a binary classification rule h_S.

Estimated value: $h_S(x)$

		1	0	Total
Real	Positive case: 1	p_{11}	p_{10}	$p_{1\cdot}$
value: y	Negative case: 0	p_{01}	p_{00}	$p_{0\cdot}$
	Total	$p_{\cdot 1}$	$p_{\cdot 0}$	$p_{\cdot\cdot} = 1$

- Eror rate, $p_{01} + p_{10}$: Probability of misclassification.
- Accuracy $=$ 1- Eror rate, $p_{00} + p_{11}$: Probability of right classification.
- Sensitivity, $p_{11}/p_{1\cdot}$: Probability of classifying correctly a positive case: True positive rate. Also known as Recall.
- Specificity, $p_{00}/p_{0\cdot}$: Probability of classifying correctly a negative case: True negative rate.
- Precision or Positive predicted value: $p_{11}/p_{\cdot 1}$, probability of having a true positive case among those classified as positive.
- F-score (or $F1$-score): combines Precision and Recall as their harmonic mean:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2p_{11}}{2p_{11} + p_{01} + p_{10}}.$$

A test sample or cross-validation techniques are required to estimate these quantities.

Evaluating a binary classification rule

From Wikipedia, the free encyclopedia

		Predicted condition			
	Total population $= P + N$	**Predicted Positive (PP)**	**Predicted Negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP), hit**	**False negative (FN), type II error, miss, underestimation**	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	**False positive (FP), type I error, false alarm, overestimation**	**True negative (TN), correct rejection**	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2 PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

https://en.wikipedia.org/wiki/Template:Diagnostic_testing_diagram

Error rate, sensitivity and specificity. Estimation

- Apparent error rate. The training data are used twice: first for estimating the classification rule $h_S(\boldsymbol{x})$, and then to validate it. Too optimistic. Biased estimation.
- Error rate in a test set. Unbiased estimation, because the test set is independent from the training set used to estimate $h_S(\boldsymbol{x})$.
- k-fold cross-validation. Almost unbiased estimation.
  - The sample (of size n) is randomly divided into k parts.
  - The model is fitted with $(k-1)$ parts and the other one is used as validation sample.
  - The k possible combinations are done.
  - At the end, a classification is obtained for each element in the sample. The cross table of these n estimated classes and the true ones provide estimations for p_{00}, p_{01}, p_{10} and p_{11}.
- Leave-one-out. It coincides with n-fold cross-validation.
- It is also possible to take m random sample divisions into two set (training and test) and then average the m cross tables.

Different cut points lead to different classification rules

- In practice, the ususal classification rules (logistic regression, k nearest neighbors, neural netwotks for binary classification, etc.) provide an estimation of the probability of being in the class 1:

$$g_S(\boldsymbol{x}) = \widehat{\Pr}(Y = 1|\boldsymbol{X} = \boldsymbol{x}).$$

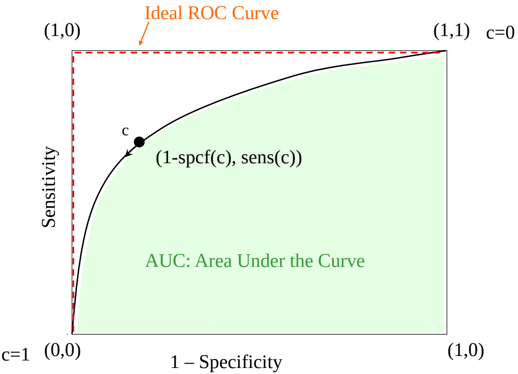
- Given a cut point $c \in [0, 1]$, which divide the range of these estimated probabilities into two parts, a classification rule is obtained: $h_c(\boldsymbol{x}) = \mathbb{I}_{(c,1]}(\widehat{\Pr}(Y = 1|\boldsymbol{X} = \boldsymbol{x}))$.
- So each cut point $c \in [0, 1]$ defines a *Sensitivity(c)* and a *Specificity(c)*.

ROC Curve (Receiver Operating Characteristic Curve)

The ROC curve is a parametric curve,

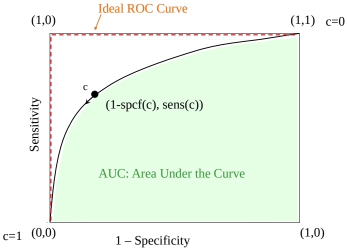
$$\{(1 - \text{Specificity}(c), \text{Sensitivity}(c)) : \; c \in [0,1]\} \in [0,1] \times [0,1],$$

starting at $(1,1)$ when $c = 0$ and finishing at $(0,0)$ when $c = 1$.



ROC Curve: A way to evaluate a rule $g_S : \mathcal{X} \to [0, 1]$.

- It does not depend on the cut point c.
- Summary measure: AUC, Area Under the Curve.
- We are proving that AUC coincides with the probability that, given two new independent cases, one negative $(-)$ and the other positive $(+)$, the classification rule assigns a greater probability of being positive to the true positive case than to the negative one.



AUC = probability or rightly order two cases, + and −

- Remember that $g_S(\boldsymbol{x}) = \widehat{\Pr}(Y = 1|\boldsymbol{X} = \boldsymbol{x})$.
- Let G_0 and G_1 be two independent random variables, with

$$G_0 \sim g_S(\boldsymbol{X}|Y = 0),\ G_1 \sim g_S(\boldsymbol{X}|Y = 1).$$

- The binary classification rule $h_c(\boldsymbol{x}) = \mathbb{I}_{(c,1]}(g_S(\boldsymbol{x}))$ has (1 - specificity) and sensitivity values, respectively,

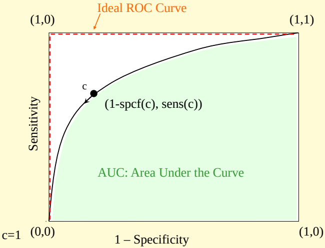
$$u = 1 - \text{spcf}(c) = \Pr(g_S(\boldsymbol{X}) > c|Y = 0) = 1 - F_{G_0}(c),$$

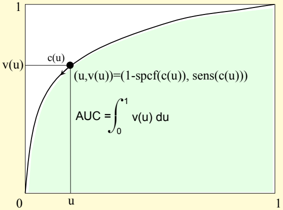
$$v = \text{sens}(c) = \Pr(g_S(\boldsymbol{X}) > c|Y = 1) = 1 - F_{G_1}(c),$$

where F_{G_0} and F_{G_1} are the distribution functions of G_0 and G_1, respectively, which we assume to be continuous with densities f_{G_0} and f_{G_1}.

AUC = probability or rightly order two cases, + and −

$$u = 1 - \mathrm{spcf}(c) = 1 - F_{G_0}(c) \Rightarrow c = F_{G_0}^{-1}(1-u),\ v = \mathrm{sens}(c) = 1 - F_{G_1}(c) = \Pr(G_1 > F_{G_0}^{-1}(1-u))$$





$$\mathrm{AUC} = \int_0^1 v(u)du = \int_0^1 \Pr(G_1 > F_{G_0}^{-1}(1-u))du \overset{(1-u=F_{G_0}(t))}{=}$$

$$-\int_1^0 \Pr(G_1 > t) f_{G_0}(t)dt = \int_0^1 \int_0^1 \mathbb{I}_{\{s>t\}} f_{G_1}(s) f_{G_0}(t) ds\, dt = \Pr(G_1 > G_0)$$

Estimating the ROC curve and the AUC

- If a test sample is available, AUC can be estimated by the Mann-Whitney-Wilcoxon test statistic value, used to test the null hypothesis that the medians of G_0 and G_1 are equal, from the data $g_S(\boldsymbol{x}_i^0)$, $i = 1, \ldots, n_0$, and $g_S(\boldsymbol{x}_j^1)$, $j = 1, \ldots, n_1$.
- The Mann-Whitney-Wilcoxon test statistic estimates the probability $\Pr(G_0 \leq G_1) = \Pr(g_S(\boldsymbol{X}|Y = 0) \leq g_S(\boldsymbol{X}|Y = 1))$.

Estimating the ROC curve and the AUC

- The full estimation of the ROC curve requires the estimation of probabilities $p_{00}(c_j)$, $p_{01}(c_j)$, $p_{10}(c_j)$ y $p_{11}(c_j)$ for a large number J of evenly spaced values c_j, $j = 0, \ldots, J$, $c_0 = 0$, $c_J = 1$.
- This problem is analogous to that of estimating the error rate $p_{01}(c_j) + p_{10}(c_j)$.
- The same techniques are used: test sample, k-fold cross validation.
- Once the ROC curve has been estimated, the AUC is computed by numerical integration:

$$\widehat{\text{AUC}} = \sum_{j=1}^{J} \frac{1}{2}(\widehat{\text{sens}}(c_j) + \widehat{\text{sens}}(c_{j+1}))(\widehat{\text{spcf}}(c_{j+1}) - \widehat{\text{spcf}}(c_j)),$$

where

$$\widehat{\text{sens}}(c_j) = \frac{\hat{p}_{11}(c_j)}{\hat{p}_{10}(c_j) + \hat{p}_{11}(c_j)}, \quad \widehat{\text{spcf}}(c_j) = \frac{\hat{p}_{00}(c_j)}{\hat{p}_{00}(c_j) + \hat{p}_{01}(c_j)}.$$

Some R implementations:

- `roc` function in package `pROC`,
- `ROC` function in package `ROC632`.

They plot the ROC curve and compute the AUC, among other.

Other ways to evaluate rules $g_S : \mathcal{X} \to [0,1]$

- We assume that the rule $g_S : \mathcal{X} \to [0,1]$ provides an estimation for the probability that a case belongs to class 1 when the explanatory variable takes the valuer $\boldsymbol{x}$ at this case:

$$g_S(\boldsymbol{x}) = \widehat{\Pr}(Y = 1 | \boldsymbol{X} = \boldsymbol{x}).$$

- Therefore, the logarithm of the likelihood function for the observed data $(y_i, \boldsymbol{x}_i)$, $i = 1, \ldots, n$, divided by n, is

$$\frac{1}{n} \log \left(\prod_{i=1}^{n} g_S(\boldsymbol{x}_i)^{y_i} (1 - g_S(\boldsymbol{x}_i))^{1-y_i} \right) =$$

$$\frac{1}{n} \sum_{i=1}^{n} \left(y_i \log g_S(\boldsymbol{x}_i) + (1 - y_i) \log(1 - g_S(\boldsymbol{x}_i)) \right).$$

- This measure, as it happens for the apparent error rate, is an optimistic estimation of the expected value of the random variable

$$\log \widehat{\Pr}(Y = Y_{n+1}|\boldsymbol{X}_{n+1}) = Y_{n+1} \log g_S(\boldsymbol{X}_{n+1}) + (1-Y_{n+1}) \log(1-g_S(\boldsymbol{X}_{n+1})),$$

  with $(Y_{n+1}, \boldsymbol{X}_{n+1})$ independent from the first n observations, which were used to estimate the rule $g_S(\boldsymbol{x})$.
- An unbiased estimation is achieved if a validation sample is available: $(y_j^v, \boldsymbol{x}_j^v)$, $j = 1, \ldots, m$, independent from the training sample.
- In this case,

$$\ell_{\mathrm{val}}(g_S) = \frac{1}{m} \sum_{j=1}^{m} \left(y_j^v \log g_S(\boldsymbol{x}_j^v) + (1 - y_j^v) \log(1 - g_S(\boldsymbol{x}_j^v)) \right).$$

- k-fold cross-validation or leave-one-out cross-validation can also be used.

SPAM E-mail Database

- From the description file: *The "spam" concept is diverse: advertisements for products/web sites, make money fast schemes, chain letters, pornography... Our collection of spam e-mails came from our postmaster and individuals who had filed spam. Our collection of non-spam e-mails came from filed work and personal e-mails, and hence the word 'george' and the area code '650' are indicators of non-spam. These are useful when constructing a personalized spam filter. One would either have to blind such non-spam indicators or get a very wide collection of non-spam to generate a general purpose spam filter.*
- Attribute Information:
  - The last column of 'spambase.data' denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail.
  - Most of the attributes indicate whether a particular word or character was frequently occuring in the e-mail.
  - The run-length attributes (55-57) measure the length of sequences of consecutive capital letters.

Practice:

- Use the script `spam.R` to read the data from the SPAM e-mail database.
- Divide the data into two parts: 2/3 for the training sample, 1/3 for the test sample. You should do it in a way that 2/3 of the SPAM e-mails are in the training sample and 1/3 in the test sample, and that the same happens for NO SPAM e-mails.
- Consider the following three classification rules:
  - Logistic regression fitted by maximum likelihood (IRWLS, `glm`).
  - Logistic regression fitted by Lasso (`glment`).
  - k-nn binary regression (you can use your own implementation or functions `knn` and `knn.cv` from the R package `class`).

  Use the training sample to fix the tuning parameters (when needed) and to estimate the model parameters (when needed).
- Use the test sample to compute and plot the ROC curve for each rule.
- Compute also the confusion matrix and the misclassification rate for each rule when using the cut point $c = 1/2$.
- Compute ℓ_{val} for each rule.

Hastie, T., R. Tibshirani, and J. Friedman (2009).
The Elements of Statistical Learning (2nd ed.).
Springer.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013).
An Introduction to Statistical Learning with Applications in R.
Springer.