# k-fold cross-validation

Anders Munch and Thomas Alexander Gerds

University of Copenhagen, Department of Public Health, Section of Biostatistics, Copenhagen, Denmark

June 22, 2019

## 1 Introduction

This document discusses the different cross-validation options of the `Score` function.

## 2 Formula

The observed outcome status at time $t$ for subject $i$ is $\tilde{Y}_i = 1_{\{\min(T_i,C_i)\leq t\}}$ where $T_i$ is the event time and $C_i$ the right censoring time. A risk prediction is a value between 0 and 1. The risk prediction of a statistical model at time $t$ for subject $i$ is based on baseline predictor variables $X_i$ and given by $\hat{R}(t|X_i)$. The inverse probability of censoring weights (IPCW) are given by $\hat{W}_i(t)$ based on a model for the conditional censoring survival distribution $P(C > t|X)$.

The data $D_n = \{(\tilde{T}_i, \Delta_i, X_i)\}_{i=1}^n$, where also $\tilde{T}_i = \min(T_i, C_i)$ and $\Delta_i = 1_{\{T_i\leq C_i\}}$) are divided into disjoint sets:

$$D_n = \underbrace{D_l}_{\text{Training set}} \cup \underbrace{D_v}_{\text{Validation set}}$$

### 2.1 `loob`

- `B` number of bootstrap samples
- $D_l^b$ = b'th bootstrap sample
- $D_v^b = D_n \setminus D_l^b$ = samples out-of-bag
- $\hat{R}_b$ the model fitted in $D_l^b$
- $0 \leq r_i \leq B$ the number of bootstrap samples where subject $i$ is out-of-bag

$$\text{loob} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{r_i}\sum_{b:i\in D_n\setminus D_b}\hat{W}_i(t)\{\tilde{Y}_i(t) - \hat{R}_b(t|X_i)\}^2$$

### 2.2 `bootcv`

- `B` number of bootstrap samples
- $D_l^b$ = b'th bootstrap sample
- $D_v^b = D_n \setminus D_l^b$ = samples out-of-bag
- $\hat{R}_b$ the model fitted in $D_l^b$
- $0 \leq m_b \leq n$ the size of $D_v^b$

$$\text{bootcv} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{m_b}\sum_{i\in D_n\setminus D_b}\hat{W}_i(t)\{\tilde{Y}_i(t) - \hat{R}_b(t|X_i)\}^2.$$

### 2.3 `cv-K` once

- `K` number of folds
- $D_l^k = D_n \setminus D_v^k$ = data without fold-k
- $D_v^k$ = data in fold-k
- $\hat{R}_k$ the model fitted in $D_l^k$
- $\hat{R}_{k_i}$ the prediction of model fitted without the fold $k_i$ where $i \in D_v^k$

$$\text{cv-K} = \frac{1}{n}\sum_{i\in D_n}\hat{W}_i(t)\{\tilde{Y}_i(t) - \hat{R}_{k_i}(t|X_i)\}^2.$$

### 2.4 `cv-K` repeated B times

Same as `cv-K` but now we repeat K-fold B times. Then, we have two possibilities:

1. average the B values of `cv-K` (like `bootcv`)
2. for each $i$ collect the B predictions (like `loob`)

TODO: formula

## 3 Testing

Copy/paste some functionality form other vignette.
Packages and source for reloading after edits.

```
library(riskRegression)
sessionInfo()
```

```
Registered S3 methods overwritten by 'ggplot2':
  method         from
  [.quosures     rlang
  c.quosures     rlang
  print.quosures rlang
riskRegression version 2019.06.13
R version 3.6.0 (2019-04-26)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: OS X El Capitan 10.11.6

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] riskRegression_2019.06.13

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.1          mvtnorm_1.0-10      lattice_0.20-38
 [4] zoo_1.8-6           assertthat_0.2.1    digest_0.6.19
 [7] foreach_1.4.4       R6_2.4.0            plyr_1.8.4
[10] backports_1.1.4     acepack_1.4.1       MatrixModels_0.4-1
[13] ggplot2_3.1.1       pillar_1.4.1        rlang_0.3.4
[16] lazyeval_0.2.2      multcomp_1.4-10     rstudioapi_0.10
[19] data.table_1.12.2   SparseM_1.77        rpart_4.1-15
[22] Matrix_1.2-17       checkmate_1.9.3     splines_3.6.0
[25] stringr_1.4.0       foreign_0.8-71      htmlwidgets_1.3
[28] munsell_0.5.0       numDeriv_2016.8-1.1 compiler_3.6.0
[31] xfun_0.7            pkgconfig_2.0.2     base64enc_0.1-3
[34] htmltools_0.3.6     nnet_7.3-12         tidyselect_0.2.5
[37] tibble_2.1.3        gridExtra_2.3       htmlTable_1.13.1
[40] prodlim_2018.04.18  Hmisc_4.2-0         rms_5.1-3.1
[43] codetools_0.2-16    crayon_1.3.4        dplyr_0.8.1
[46] MASS_7.3-51.4       timereg_1.9.3       grid_3.6.0
[49] nlme_3.1-140        polspline_1.1.14    gtable_0.3.0
[52] magrittr_1.5        scales_1.0.0        stringi_1.4.3
[55] latticeExtra_0.6-28 sandwich_2.5-1      Formula_1.2-3
[58] TH.data_1.0-10      lava_1.6.5          RColorBrewer_1.1-2
[61] iterators_1.0.10    tools_3.6.0         cmprsk_2.2-8
[64] glue_1.3.1          purrr_0.3.2         survival_2.44-1.1
```

```
[67] colorspace_1.4-1    cluster_2.0.9       knitr_1.23
[70] quantreg_5.40
```

Setup data

```
set.seed(18)
astrain <- simActiveSurveillance(278)
astest <- simActiveSurveillance(208)
astrain[,Y1:=1*(event==1 & time<=1)]
astest[,Y1:=1*(event==1 & time<=1)]
lrfit.ex <- glm(Y1~age+lpsaden+ppb5+lmax+ct1+diaggs,data=astrain,
    family="binomial")
lrfit.inc <- glm(Y1~age+lpsaden+ppb5+lmax+ct1+diaggs+erg.status,data
    =astrain,family="binomial")
## Score(list("Exclusive ERG"=lrfit.ex,"Inclusive ERG"=lrfit.inc),
    data=astest,formula=Y1~1,se.fit=0L,metrics="brier",contrasts=
    FALSE)
```

Now also works with for `bootcv` without errors, now also returning no-NA IPA. These are negativ, however, which I don't know if make sense.

```
X1 <- Score(list("Exclusive ERG"=lrfit.ex,"Inclusive ERG"=lrfit.inc),
    data=astest,
        formula=Y1~1,summary="ipa",se.fit=0L,metrics="brier",
    contrasts=FALSE,
        split.method = "bootcv", B=100)
```

```
X1
```

```
Metric Brier:

Results by model:

           model Brier     IPA
1:    Null model 0.157  0.0000
2: Exclusive ERG 0.169 -0.0781
3: Inclusive ERG 0.163 -0.0396

Bootstrap cross-validation based on 100 bootstrap samples (drawn with replacement) each of s
```

And gives some result for `cv` when just using the same method as for `bootcv`. Not sure these are correct however.

```
X1 <- Score(list("Exclusive ERG"=lrfit.ex,"Inclusive ERG"=lrfit.inc),
    data=astest,
        formula=Y1~1,summary="ipa",se.fit=0L,metrics="brier",
    contrasts=FALSE,
        split.method = "cv5", B=100)
```

X1

Metric Brier:

Results by model:

| | model | Brier | IPA |
|---|---|---|---|
| 1: | Null model | 0.154 | 0.00000 |
| 2: | Exclusive ERG | 0.160 | -0.03601 |
| 3: | Inclusive ERG | 0.153 | 0.00392 |

5-fold cross-validation repeated 100 times.