Model Evaluation

Considerations for Time-to-Event Studies

Daniel Saggau

11/12/2020

Overview

1) Time to Event Studies
2) Classical Model Evaluation: Brier Score and AUC
3) TTS Model Evaluation: IBS and c-index
4) Discussion
5) Considerations

Time-to Event Studies

- Diagnostic and Prognostic Study
- Working with censored data
- Highly relevant for clinical/epidemiological studies
- In Economics e.g. to examine when a subject/borrower will default
- Survival time T, the probability of death a time point h(t,x), cumulative hazard H, and survival function S

Non-parametric hazard model (Kaplan Meier Estimator):

$h(t) = \frac{d}{dt}[logS(t)]$
$S(t) = exp(-H(t))$
Semi-parametric proportional hazard model (Cox Estimator):

$h(t|x\beta) = h_0(t)exp(\beta^T x)$

Components of Model Evaluation

1. **Discrimination vs. Calibration (vs. Clinical Usefulness)**
   - Discrimination: Are we able to discriminate between e.g. sick and healthy patients ?
   - Calibration: How concise is our prediction accuracy ?
   - Clinical Usefulness: Will our model create more benefits than harm?
2. **Label vs. Probability**
   - AUC (label based error measure via specificity and sensitivity)
   - Brier Score (probability from true class label)

Brier Score

- Score is based on loss function at a certain point in time
- Other loss measures are the log loss or the integrated log loss
- Can Plot brier score via prediction error curves (pec)
- MSE: Scores range from 0 to infinity and closer to 0 is better
- Brier Score: range from 0 to 1

Formula for the Brier Score

MSE for Regression (L2 Loss):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)}) - \hat{y}^{(i)})^2$$

The Brier Score is the MSE for Classification:

$$\text{BS} = \frac{1}{n}\sum_{i=1}^{n}(\hat{\pi}(x^{(i)}) - y^{(i)})^2$$

AUC - Talking about the Curve

- Plotting TPR and TNR at different thresholds
- We integrate over all thresholds to get the AUC
- Scores range from 0 to 1
- A higher score is better and a score of 0.5 is basically a random model

Components of the ROC

Sensitivity or true positive rate:

$$\mathrm{TPF} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Specificity or true negative rate:

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

Limitations of traditional ME tools

- Working with censored data
- Working with hazards and survival function
- Account for time dependent covariates

Early approaches:

- Excluding subjects with right censored data and only evaluate on the complete data
- Problem: Losing a lot of data and potentially inducing bias

Solution:

- Inverse of the probability of censoring weighted estimate (IPCW)

From AUC to c-index

- AUC: *"is individual A likely to have a stroke within the next 5 years?"*
- c-index: *"is individual A or individual B more likely to have a stroke?"*
- Concordance (consistency) & discordance (inconsistency) pairs
- Kendall rank correlation coefficient test as conservative basis
- Popular assumption: right censored data

Differentiation AUC and c-index

$$\mathrm{AUC} = \Pr(\mathrm{Risk_t(i)} > \mathrm{Risk_t(j)} | \mathrm{i\ has\ event\ before\ t\ and\ j\ has\ event\ after\ t})$$

$$\mathrm{C} = \Pr(\mathrm{Risk_t(i)} > \mathrm{Risk_t(j)} | \mathrm{i\ has\ event\ before\ t})$$

Example of formula for c-index

Formula for the c-index

$$\frac{\text{Concordant Pairs}}{\text{Concordant Pairs} + \text{Discordant Pairs}}$$

Mathematically, we can define the c-index for time dependent covariates as:

$$C^{td} = \frac{\Pr(Risk_t(i) > Risk_t(j) \& T_i < T_j \& D_i = 1)}{\Pr(Ti < Tj | D_i = 1)}$$

Integrated Brier Score (IBS)

- In e.g. 'pec' the score is called the cumulative predictive error curves
- Area under the prediction error curve
- Working with time dependent survival probabilities

Formula for IBS (Population)

(integrated == T):

$$L(S) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{j=1}^{T} L(S_i, t_i | t_j^*)$$

- N is the number of observations
- S_i is the predicted survival function
- t is the time of the event
- t^* the time before event

Implementation: Basic Setup

- Using simulated survival data with 10000 observations
- Data entails: eventtime, censtime, time, event, X1, X2, status

```
set.seed(123)
library("prodlim")
library("survival")
library("pec")
dat <- SimSurv(10000)
models <- list(
  "Cox.X1" = coxph(Surv(time, status) ~ X1,
    data = dat, x = TRUE, y = TRUE),
  "Cox.X2" = coxph(Surv(time, status) ~ X2,
    data = dat, x = TRUE, y = TRUE),
  "Cox.X1.X2" = coxph(Surv(time, status) ~ X1 + X2,
    data = dat, x = TRUE, y = TRUE))
```

Implementation: Integrated Brier Score

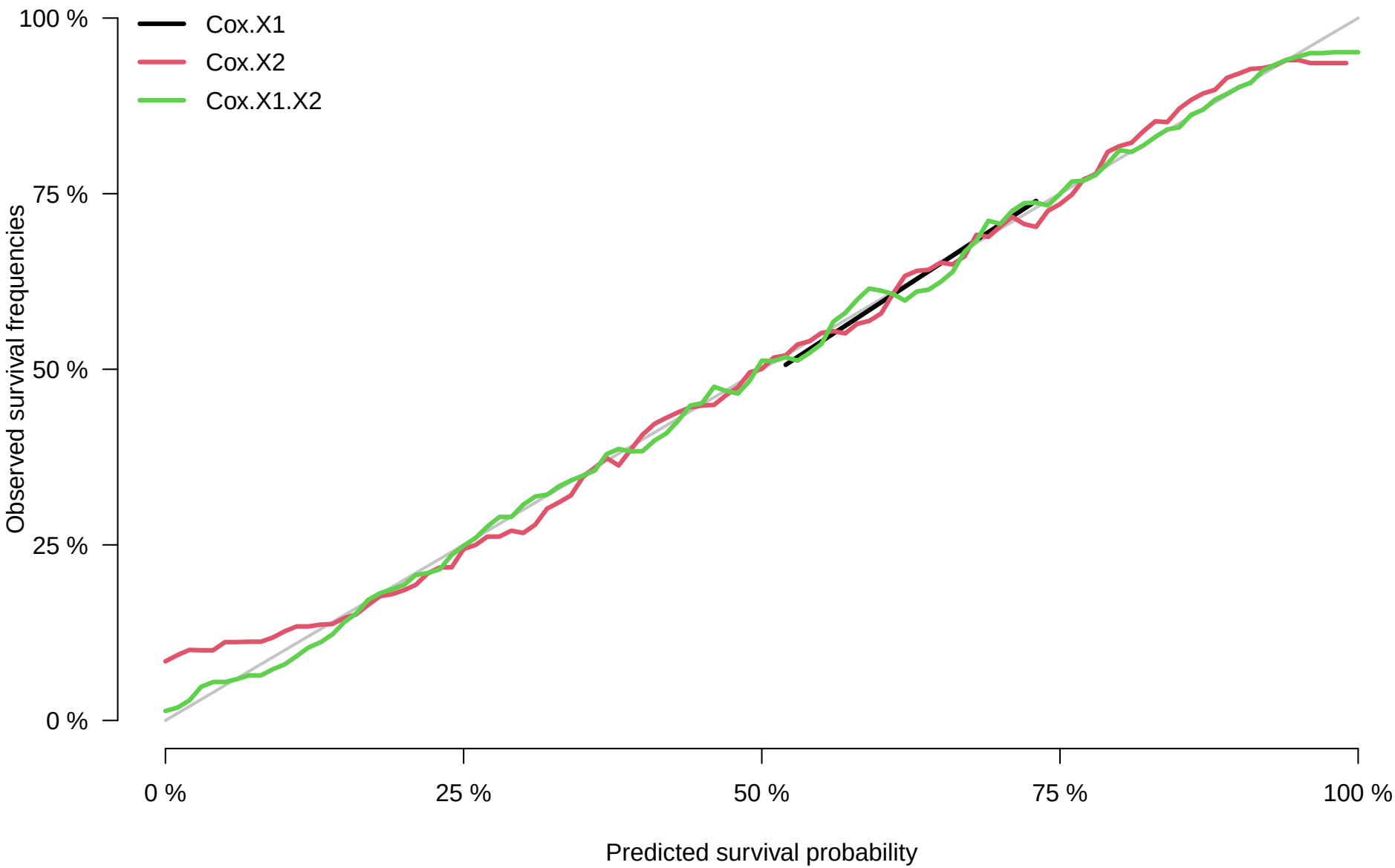
IPCW based on Kaplan Meier estimates:

```
perror <- pec(
  object = models,
  formula = Surv(time, status) ~ 1, # ,~X1 +X2, for cox
  data = dat, exact = TRUE, cens.model = "marginal", # .model="cox"
  splitMethod = "none",
  B = 0
)
```

- **cens.model** is our ipcw estimator
- **splitMethod** is the internal validation design
- **B** is the number bootstrap samples & **M** the bootstrap size
- Optional: **cause** for competing risks
- If **exact** is equal to T then we estimate pec at all the unique values of response

Implementation: Calibration Plot

```
calPlot(models)
```



Implementation: Prediction Error Curve

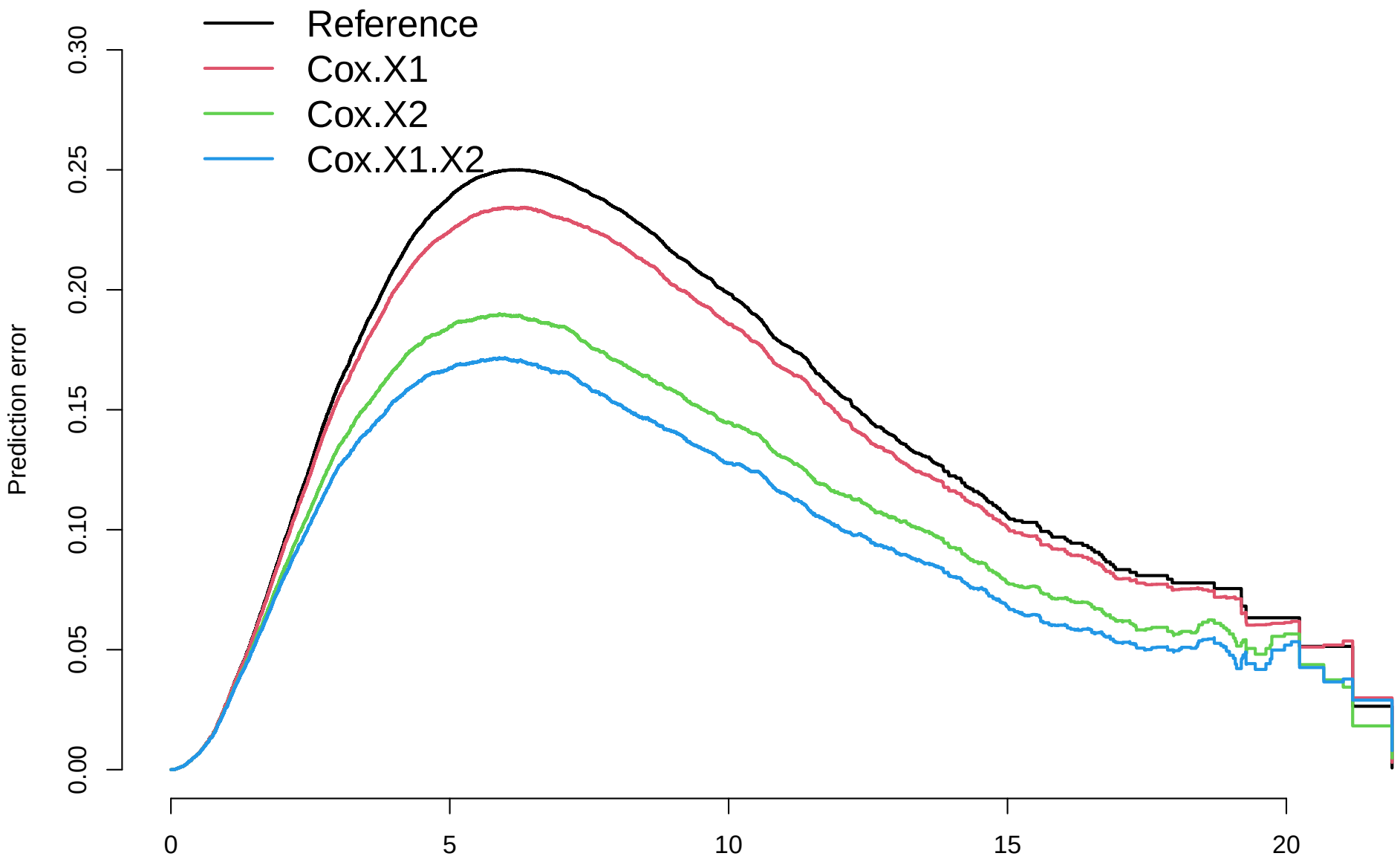
```
summary(perror, times = quantile(dat$time[dat$status == 1], c(.25, .5, .75, 1)))
```

```
##
## Prediction error curves
##
##
## No data splitting: either apparent or independent test sample performance
##
##  AppErr
##     time n.risk Reference Cox.X1 Cox.X2 Cox.X1.X2
## 1  2.568   7892     0.132  0.128  0.112     0.106
## 2  4.270   5644     0.220  0.208  0.174     0.159
## 3  6.513   3179     0.249  0.233  0.188     0.169
## 4 21.189      1     0.026  0.030  0.018     0.029
```

- A lower score is better here
- Comparing at quantiles
- Frequently people only use the first 3 quantifies

Plotting the prediction error curve

```
plot(perror)
```



Implementation: Cumulative Prediction Error Score (IBS)

```r
crps(perror, times = quantile(dat$time[dat$status == 1], c(.25, .5, .75, 1)))
```

```
##
## Integrated Brier score (crps):
##
##           IBS[0;time=2.6) IBS[0;time=4.3) IBS[0;time=6.5) IBS[0;time=21.2)
## Reference           0.051           0.102           0.150            0.142
## Cox.X1              0.050           0.099           0.143            0.134
## Cox.X2              0.046           0.086           0.120            0.108
## Cox.X1.X2           0.044           0.081           0.111            0.097
```

```r
# ibs(perror,times= quantile(dat$time[dat$status==1], c(.25, .5, .75, 1)))
```

- A lower score is better with scores ranging from 0 to 1
- Looking at different time points thresholds
- Score can also be derived for the individual or a specific time point in various packages

Implementation: c-index

Components of the c-index function

```
cindex <- cindex(models,
  formula = Surv(time, status) ~ 1,
  cens.model = "marginal", data = dat,
  eval.times = quantile(dat$time[dat$status == 1], c(.25, .5, .75, 1))
)
```

- **formula** is our survival formula
- **cens.model** is our method for estimating the IPCW
- **splitMethod** is the internal validation design
- **B** the number of boostrap samples & **M** the size of the boostrap sample
- Extensions: **cause** used for competing risks

Implementation: c-index summary value

```
cindex$response
```

```
##
## Right-censored response of a survival model
##
## No.Observations: 10000
##
## Pattern:
##                  Freq
##  event           6045
##  right.censored 3955
```

```
cindex$AppCindex
```

```
## $Cox.X1
## [1] 0.6053041 0.6024758 0.5964374 0.5883673
##
## $Cox.X2
## [1] 0.7477848 0.7317839 0.7206638 0.7101860
##
## $Cox.X1.X2
## [1] 0.7728949 0.7615609 0.7538084 0.7435431
```

```
cindex$time
```

```
##       25%       50%       75%      100%
##  2.568333  4.269680  6.513200 21.188677
```

```
cindex$cens.model
```

```
## [1] "marginal"
```

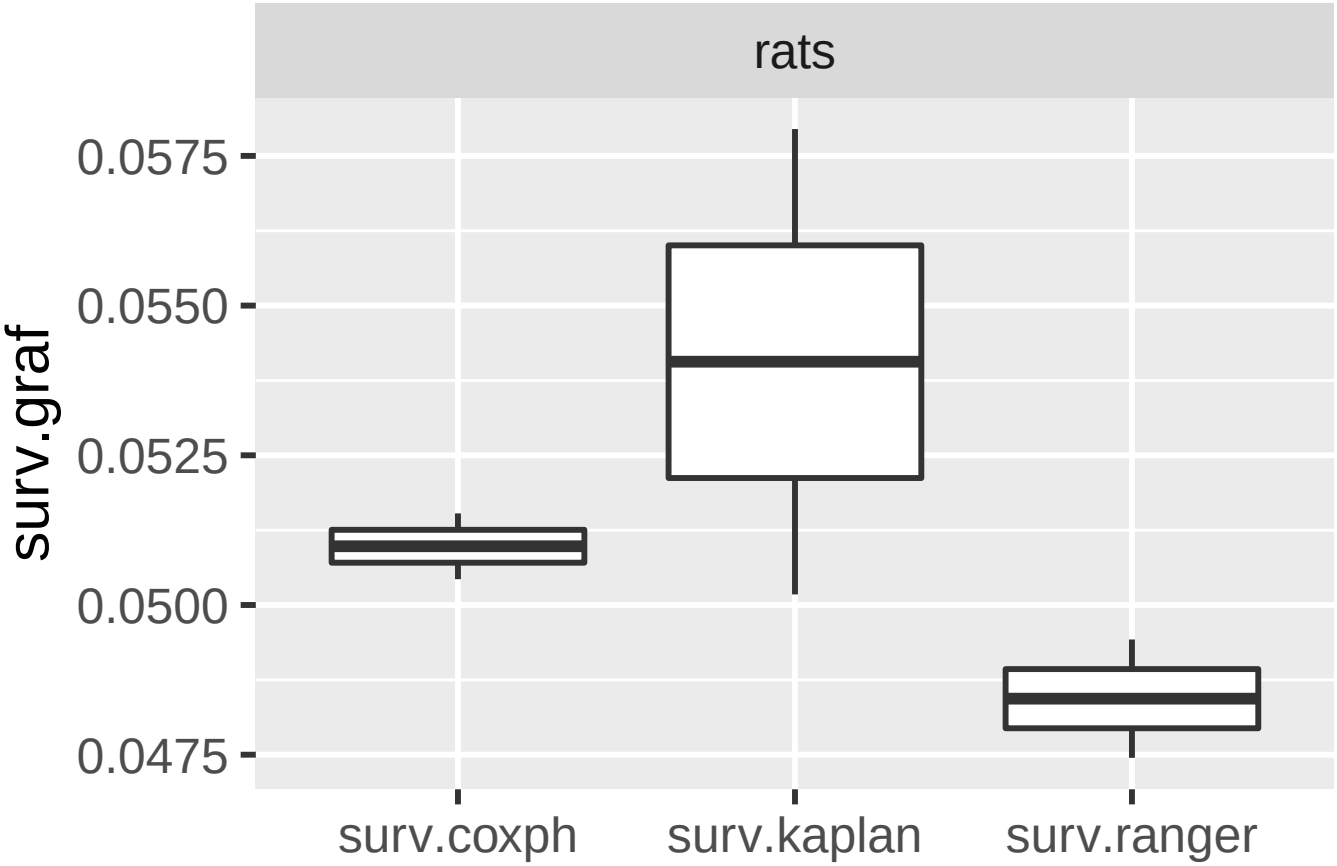
Implementation: Measures in mlr3proba

```
library("mlr3")
library("mlr3learners")
library("mlr3proba")
library("mlr3viz")

##' measure = msr("surv.graf") # for c-index you can use surv.cindex
##' bmr = benchmark(benchmark_grid(task, learners, rsmp("cv", folds = 3)))
##' bmr$aggregate(measure)

# Modification via:
#' MeasureSurvGraf$new(integrated = TRUE, times, method = 2, se = FALSE)
```

- If integrate == T then: times = vector of time-points over which to integrate the score; otherwise: single time point
- method ==1: Approx. to integration by dividing sample mean weighted equally
- method ==2: Approx. to integration via mean weighted by difference between time points (default in 'pec')

Implementation: mlr3Proba Benchmark Example

```
autoplot(bmr, measure = measure)
```



Discussion

- c-index has gained popularity because of it's interpretability
- Integrated Brier Score accounts for both calibration and discrimination
- For predictive modelling, you want to account for both components
- IBS allows for differentiation of 'useless' and 'harmful' models
- Clinical consequences problematic

Considerations

- Decision Curve Analysis (clinical consequences)
- Net Reclassification Improvement (clinical consequences)
- Other estimators like SVM estimators for the censored data
- Time dependent ROC/AUC

Literature and Recommendations

Introduction:

- Steyerberg, E. W., Vickers, A. J., Cook, N. R., Gerds, T., Gonen, M., Obuchowski, N., ... & Kattan, M. W. (2010). Assessing the performance of prediction models: a framework for some traditional and novel measures. Epidemiology (Cambridge, Mass.), 21(1), 128.

Modifications:

- Blanche, P., Kattan, M. W., & Gerds, T. A. (2019). The c-index is not proper for the evaluation of-year predicted risks. Biostatistics, 20(2), 347-357.
- Khosla, A., Cao, Y., Lin, C. C. Y., Chiu, H. K., Hu, J., & Lee, H. (2010, July). An integrated machine learning approach to stroke prediction. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 183-192).

Use Cases:

Decision Curve Analysis:

https://rdrr.io/github/ddsjoberg/dca/man/stdca.html

Concordance Related Model Evaluation:

- https://rpubs.com/kaz_yos/survival-auc
- https://datascienceplus.com/time-dependent-roc-for-survival-prediction-models-in-r/