Generating High-Dimensional Competing Risk Survival Settings

A thing that I do for reasons

Lukas Burk

Leibniz Institute for Prevention Research & Epidemiology – BIPS

2023-05-13

BioWiMium

Introduction

- Simulation setup part of ongoing project
- Motivation: Variable selection in high-dimensional settings with competing risks
- Outcome: 2 competing events, censoring approx. equal prevalence
- Focus more on high-dimensional setting than sophisticated survival outcome

Generating Survival Times (I)

Binder et al. (2009)

BIOINFORMATICS **ORIGINAL PAPER**

Vol. 25 no. 7 2009, pages 890–896
doi:10.1093/bioinformatics/btp088

Gene expression

Boosting for high-dimensional time-to-event data with competing risks

Harald Binder[1,2,*], Arthur Allignol[1,2], Martin Schumacher[2] and Jan Beyersmann[1,2]

[1]Freiburg Center for Data Analysis and Modeling, University of Freiburg, Eckerstr. 1 and [2]Institute of Medical Biometry and Medical Informatics, University Medical Center Freiburg, Stefan-Meier-Str. 26, 79104 Freiburg, Germany

Received on October 16, 2008; revised on January 19, 2009; accepted on February 12, 2009

Advance Access publication February 25, 2009

Associate Editor: Joaquin Dopazo

Binder and Schumacher (2008)

Adapting Prediction Error Estimates for Biased Complexity Selection in High-Dimensional Bootstrap Samples

Harald Binder & Martin Schumacher

Universität Freiburg i. Br.

Nr. 100 December 2007

Bender et al. (2005)

Generating Survival Times to Simulate Cox Proportional Hazards Models

Ralf Bender[1], Thomas Augustin[2], Maria Blettner[1]

[1]Dept. of Epidemiology and Medical Statistics, School of Public Health
University of Bielefeld, Germany

[2]Department of Statistics, University of Munich, Germany

Predictors: Structure

- $N = 400$, $p = 5000$, 16 informative (12 per event)
- Organized in 4 blocks á 250 correlated variables + uncorrelated noise

Blocks

- Block 1: $\rho \approx 0.5$
- Block 2: $\rho \approx 0.35$
- Block 3: $\rho \approx 0.05$
- Block 4: $\rho \approx 0.32$
- Rest: $\rho \approx 0$

Predictors: Generation

- $j = 1, \dots, p$ and $i = 1, \dots, N$
- $\epsilon_{ij} \sim \mathcal{N}(0,1)$ and $u_{i\{1,2,3\}} \sim \mathcal{U}(0,1)$

$$x_{ij} = \begin{cases} -1 + \epsilon_{ij} & i \leq 0.5n \text{ and } j \leq 0.05p \\ 1 + \epsilon_{ij} & i > 0.5n \text{ and } j \leq 0.05p \\ 1.5 \cdot \mathbf{1}\{u_{i1} < 0.4\} + \epsilon_{ij} & 0.05p < j \leq 0.1p \\ 0.5 \cdot \mathbf{1}\{u_{i2} < 0.7\} + \epsilon_{ij} & 0.1p < j \leq 0.2p \\ 1.5 \cdot \mathbf{1}\{u_{i3} < 0.3\} + \epsilon_{ij} & 0.2p < j \leq 0.3p \\ \epsilon_{ij} & j > 0.3p \end{cases}$$

Effect Assignment

- Cause-specific hazards with coefficients β_k
- $\beta_{jk} = \pm 0.5$ for effect variables
- Block 1 (**Mutual**): 4 variables
  - same effect on both cause-specific hazards
- Block 2 (**Reversed**): 4 variables
  - positive effect on cause 1 and negative on cause 2
- Block 3 (**Disjoint**): 4 + 4 variables
  - 3.1: effect on cause 1 only
  - 3.2: effect on cause 2 only

Event and Censoring Times

- Cox-exponential model for outcome $k = 1, 2$
- Baseline hazards for event times T_i and censoring times C_i: $\lambda_k = \lambda_C = 0.1$

$$T_i^{(k)} = -\frac{U_i}{\lambda_k \exp(\mathbf{x}_i^T \beta_k)} C_i = \frac{-U_i}{\lambda_C} \quad (1)$$

$$t_i = \min(T_i^{(1)}, T_i^{(2)}, C_i) \quad (2)$$

$$\delta_i = \begin{cases} 0 & \text{if } t_i = C_i \\ 1 & \text{if } t_i = T_i^{(1)} \\ 2 & \text{if } t_i = T_i^{(2)} \end{cases} \quad (3)$$

Thanks!

Next up: Implementation details nobody will want to see but I but on here any just in case because I don’t have too much else to talk about in this regard sorry

Predictors: Code


```r
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
ui1 <- runif(n); ui2 <- runif(n); ui3 <- runif(n)
j_seq <- seq_len(p)
block1 <- which(j_seq <= 0.05 * p)
X[seq_len(n/2), block1] <- -1 + X[seq_len(n/2), block1]
X[-seq_len(n/2), block1] <- 1 + X[-seq_len(n/2), block1]
block2 <- which((j_seq > (0.05 * p)) & (j_seq <= (0.1 * p)))
X[, block2] <- 1.5 * (ui1 < 0.4) + X[, block2]
block3 <- which((0.1 * p < j_seq) & (j_seq <= 0.2 * p))
X[, block3] <- 0.5 * (ui2 < 0.7) + X[, block3]
block4 <- which((0.2 * p < j_seq) & (j_seq <= 0.3 * p))
X[, block4] <- 1.5 * (ui3 < 0.3) + X[, block4]
```

Efect Assignment: Code


```r
ce <- 0.5
# first block
j_block1 <- which(j_seq <= 0.05 * p)
beta1[j_block1[1:4]] <- ce
beta2[j_block1[1:4]] <- ce
# second block
j_block2 <- which((j_seq > (0.05 * p)) & (j_seq <= (0.1 * p)))
beta1[j_block2[1:4]] <- ce
beta2[j_block2[1:4]] <- -ce
# third block
j_block3 <- which((0.1 * p < j_seq) & (j_seq <= 0.2 * p))
beta1[j_block3[1:4]] <- -ce
beta2[j_block3[5:8]] <- ce # offset by 4
```

Event Times: Code


```r
lp1 <- X %*% beta1
lp2 <- X %*% beta2

Ti1 <- -log(runif(n)) / (lambda1 * exp(lp1))
Ti2 <- -log(runif(n)) / (lambda2 * exp(lp2))
Ci <- -log(runif(n)) / lambda_c

ti <- pmin(Ti1, Ti2, Ci)
di <- as.integer(Ti1 <= Ci | Ti2 <= Ci)
di[which(Ti2 <= Ti1 & Ti2 <= Ci)] <- 2
```

Thank you for your attention

www.leibniz-bips.de/en

Contact
Lukas Burk
Leibniz Institute for Prevention Research
and Epidemiology – BIPS
Achterstraße 30
D-28359 Bremen
burk@leibniz-bips.de

Bibliography

Bender, R., Augustin, T., & Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models. Statistics in Medicine, 24(11), 1713–1723. https://doi.org/10.1002/sim.2059

Binder, H., Allignol, A., Schumacher, M., & Beyersmann, J. (2009). Boosting for high-dimensional time-to-event data with competing risks. Bioinformatics, 25(7), 890–896. https://doi.org/10.1093/bioinformatics/btp088

Binder, H., & Schumacher, M. (2008). Adapting prediction error estimates for biased complexity selection in high-dimensional bootstrap samples. Statistical Applications in Genetics and Molecular Biology, 7(1), Article12. https://doi.org/10.2202/1544-6115.1346