

survivalGPU : Analyses de survie sur cartes graphiques

Alexis van STRAATEN¹ Jean FEYDY² Anne-Sophie JANNOT^{2,3}

¹Assistance Publique-Hôpitaux de Paris (AP-HP), Service d'informatique Médicale, Biostatistiques Et Santé Publique, Hôpital Européen Georges Pompidou, Paris

²HeKA, Centre de Recherche des Cordeliers, INSERM, INRIA, Paris

³Cellule opérationnelle de la Banque Nationale de Données Maladies Rares, AP-HP, Paris

ASSISTANCE
PUBLIQUE  HÔPITAUX
DE PARIS

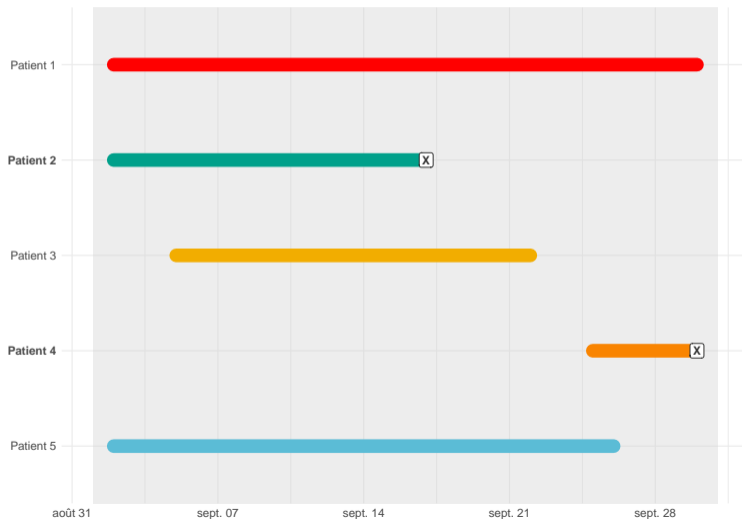

EPI-PHARE
épidémiologie des produits de santé

Inria


HeKA

Contexte des données censurées

Données avec une fenêtre d'observation qui dépend du patient.



Modèle de régression avec données censurées : Cox (1972)

Implémentation sur R : package `survival` de Terry Therneau, avec la fonction `coxph()`

Actuellement dans la recherche médicale:

- ▶ Accès à de larges bases de données de santé.
- ▶ `survival` est acceptable pour des petites études, mais non utilisable pour des études à grande échelle.

Solution : utiliser les ressources des cartes graphiques.

Objectif : package `survival` avec support GPU.

Implémentation de 2 modèles :

- ▶ Cox
- ▶ WCE (Weighted Cumulative Exposure) pour la pharmacovigilance

Plus-values de survivalGPU :

- ▶ Scalable
- ▶ Rapide
- ▶ Gestion des bootstraps
- ▶ Rétro-compatibilité avec survival



Construction de survivalGPU

Etapas de construction du package :

1. Revue du code du package `survival`
2. Traduction du solveur de Cox (Newton sur fonction convexe) en python avec `pytorch` pour la compatibilité avec les cartes graphiques Nvidia
3. Construction du package R avec les fonctions python en utilisant `reticulate`
4. Tests unitaires : reprise des tests de `survival` et résultats identiques entre `survival` et `survivalGPU` (en cours)

Tests unitaires

Exemple d'un test unitaire dans survival

```
test3 <- data.frame(futime=1:12,  
                   fustat=c(1,0,1,0,1,0,0,0,0,0,0,0),  
                   x1=rep(0:1,6),  
                   x2=c(rep(0,6), rep(1,6)))
```

```
test3  
  
   futime fustat x1 x2  
1         1     1  0  0  
2         2     0  1  0  
3         3     1  0  0  
4         4     0  1  0  
5         5     1  0  0  
6         6     0  1  0  
7         7     0  0  1  
8         8     0  1  1  
9         9     0  0  1  
10        10     0  1  1  
11        11     0  0  1  
12        12     0  1  1
```

```
library(survival)  
fit3 <- coxph(Surv(futime, fustat) ~ x1 + x2, test3, iter=25)  
  
# Actual solution  
# time 1, 12 at risk, 3 each of x1/x2 = 00, 01, 10, 11  
# time 2, 10 at risk, 2, 3, 2, 3  
# time 5, 8 at risk, 1, 3, 1, 3  
# Let r1 = exp(beta1), r2= exp(beta2)  
# loglik = -log(3 + 3r1 + 3r2 + 3 r1*r2) - log(2 + 2r1 + 3r2 + 3 r1*r2) -  
#          log(1 + r1 + 3r2 + 3 r1*r2)  
  
true <- function(beta) {  
  r1 <- exp(beta[1])  
  r2 <- exp(beta[2])  
  loglik <- -log(3*(1+ r1+ r2+ r1*r2)) - log(2+ 2*r1 + 3*r2 + 3*r1*r2) -  
            log(1 + r1 + 3*r2 + 3*r1*r2)  
  loglik  
}
```

```
paste("log vraisemblance avec coxph() :", round(fit3$loglik[2], 3))
```

```
[1] "log vraisemblance avec coxph() : -1.792"
```

```
paste("log vraisemblance avec true() :", round(true(fit3$coef), 3))
```

```
[1] "log vraisemblance avec true() : -1.792"
```

Applications avec le jeu de données synthétique drugdata

Exemples et tests avec WCE::drugdata

```
library(WCE)  
drugdata[43:53,]
```

	Id	Event	Start	Stop	sex	age	dose
43	1	0	42	43	1	40	2.5
44	1	0	43	44	1	40	2.5
45	1	0	44	45	1	40	2.5
46	1	0	45	46	1	40	2.5
47	1	0	46	47	1	40	2.5
48	1	0	47	48	1	40	2.5
49	1	1	48	49	1	40	2.5
366	2	0	0	1	0	48	0.0
367	2	0	1	2	0	48	0.0
368	2	0	2	3	0	48	0.0
369	2	0	3	4	0	48	0.0

Description drugdata

Nombre de données		77 038
Nombre de patients		500
Evenements		383 (77%)
Sexe	Hommes	365 (73%)
	Femmes	135 (27%)
Age		40.76 (13.13)
Suivi		154.08 (101.11)

mean (sd) for numeric variables
N (%) for categorical variables

Comparison survival et survivalGPU

```
library(survival)

## Cox model
coxph_model <- coxph(Surv(Start, Stop, Event) ~ dose,
                    data = drugdata)
round(exp(coxph_model$coefficients), 3) # HR (Hazard Ratio)
```

```
dose
1.246
```

```
## WCE model
wce_model <- WCE(drugdata, "Cox", nknots = 1, cutoff = 90, id = "Id",
                event = "Event", start = "Start", stop = "Stop",
                expos = "dose", covariates = c("sex", "age"))
summary(wce_model)
```

Unconstrained estimated WCE function (Proportional hazards model).***

Estimated coefficients for the covariates:

	coef	exp(coef)	se(coef)	z	p
sex	0.6876	1.9889	0.1189	5.7849	0.0000
age	0.0116	1.0116	0.0040	2.9097	0.0036

Partial log-likelihood: -1891.644 BIC: 3824.924

Number of events: 383

Use `plot(wce_model)` to see the estimated weight function corresponding to this model.

If you report these results, please cite Sylvestre MP, Abrahamowicz M. Flexible Modeling of the Effects of Time-Dependent Exposures on the Hazard. *Statistics in Medicine* 2009; 28(27):3437-3453.

```
library(survivalGPU)

## Cox model
coxphGPU_model <- coxphGPU(Surv(Start, Stop, Event) ~ dose,
                           data = drugdata)
round(exp(coxphGPU_model$coefficients), 3)
```

```
dose
[1,] 1.246
```

```
## WCE model
wceGPU_model <- wceGPU(drugdata, nknots = 1, cutoff = 90, id = "Id",
                      event = "Event", start = "Start", stop = "Stop",
                      expos = "dose", covariates = c("sex", "age"))
summary(wceGPU_model)
```

Estimated coefficients for the covariates :

	coef	CI 2.5 %	CI 97.5 %	exp(coef)	se(coef)	z	p
sex	0.6876	0.4546	0.9206	1.9889	0.1189	5.8	7e-09 ***
age	0.0116	0.0038	0.0194	1.0116	0.0040	2.9	0.004 **

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1	

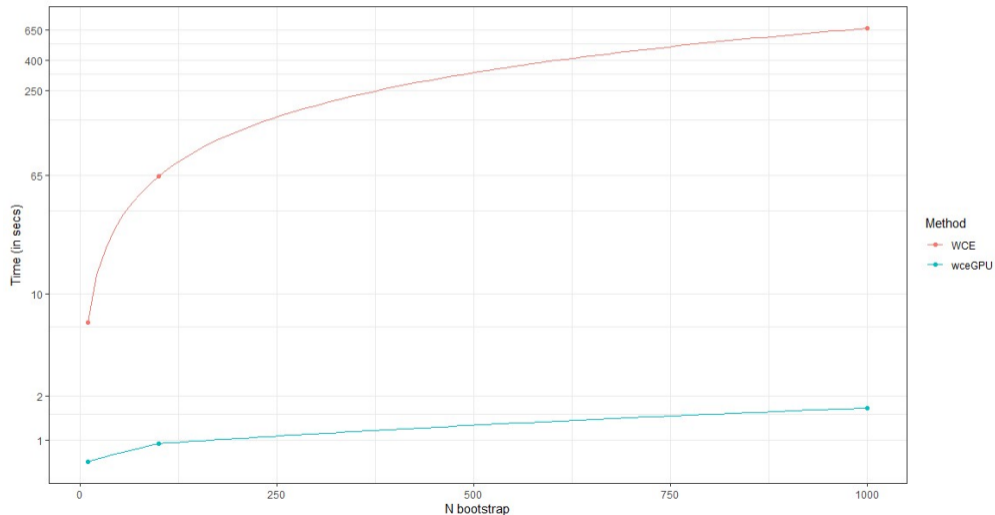
Number of events : 383

Partial log-Likelihoods : -1891.64

BIC : 3824.92

Benchmark du modèle WCE

Benchmark du modèle WCE avec drugdata



- ▶ Implémentation des features de `survival`, et de nouvelles options tels que les tests de permutation
- ▶ Validation du package à travers des simulations intensives

Package disponible sur Github : <https://github.com/jeanfeydy/survivalGPU>