Introduction à l'analyse de survie

Modèle de Kaplan–Meier, modèle de Cox

Paul MINCHELLA
paul.minchella@lyon.unicancer.fr

Sommaire

Idée générale

Le maximum de vraisemblance consiste à estimer les paramètres θ d'un modèle de façon à **rendre les données observées les plus probables** :

$$\widehat{\theta}_{\text{MLE}} = \arg\max_{\theta \in \Theta} L(\theta) \quad \text{avec} \quad L(\theta) = \prod_{i=1}^{n} f(y_i; \theta).$$

Intuition

Parmi tous les modèles possibles $f(y; \theta)$, on retient celui pour lequel les observations $(y_1, \ldots, y_n)$ auraient eu la plus grande probabilité d'être observées.

Log-vraisemblance

On travaille sur

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{n} \log f(y_i; \theta),$$

ce qui simplifie les calculs et ne modifie pas le maximum.

Condition du maximum

$$\frac{\partial \ell(\theta)}{\partial \theta}\Big|_{\theta=\hat{\theta}} = 0, \quad \frac{\partial^2 \ell(\theta)}{\partial \theta^2}\Big|_{\theta=\hat{\theta}} < 0.$$

Propriétés asymptotiques de l'estimateur MLE

Sous des hypothèses générales :

- ✓ **Consistant** : $\widehat{\theta}_{\text{MLE}} \to \theta^\star$ lorsque $n \to \infty$,
- ✓ **Asymptotiquement normal** :

$$\sqrt{n}(\widehat{\theta}_{\text{MLE}} - \theta^\star) \xrightarrow{\mathcal{L}} \mathcal{N}\big(0, \mathcal{I}(\theta^\star)^{-1}\big),$$

- ✓ **Efficace** : atteint la borne de Cramér–Rao.

Propriétés asymptotiques de l'estimateur MLE

Sous des hypothèses générales :

- ✓ **Consistant** : $\widehat{\theta}_{\text{MLE}} \to \theta^\star$ lorsque $n \to \infty$,
- ✓ **Asymptotiquement normal** :

$$\sqrt{n}(\widehat{\theta}_{\text{MLE}} - \theta^\star) \xrightarrow{\mathcal{L}} \mathcal{N}\big(0, \mathcal{I}(\theta^\star)^{-1}\big),$$

- ✓ **Efficace** : atteint la borne de Cramér–Rao.

Cas de la survie (censure à droite)

Chaque individu contribue à la vraisemblance selon son statut :

$$L(\theta) = \prod_{i=1}^{n} \big[f(t_i;\theta)\big]^{\delta_i} \big[\mathcal{S}(t_i;\theta)\big]^{1-\delta_i},$$

où $\delta_i = 1$ si l'évènement est observé et 0 sinon.

Propriétés asymptotiques de l'estimateur MLE

Sous des hypothèses générales :

- ✓ **Consistant** : $\widehat{\theta}_{\text{MLE}} \to \theta^\star$ lorsque $n \to \infty$,
- ✓ **Asymptotiquement normal** :

$$\sqrt{n}(\widehat{\theta}_{\text{MLE}} - \theta^\star) \xrightarrow{\mathcal{L}} \mathcal{N}\big(0, \mathcal{I}(\theta^\star)^{-1}\big),$$

- ✓ **Efficace** : atteint la borne de Cramér–Rao.

Cas de la survie (censure à droite)

Chaque individu contribue à la vraisemblance selon son statut :

$$L(\theta) = \prod_{i=1}^{n} \big[f(t_i;\theta)\big]^{\delta_i} \big[\mathcal{S}(t_i;\theta)\big]^{1-\delta_i},$$

où $\delta_i = 1$ si l'évènement est observé et 0 sinon.

Lien avec le modèle de Cox Le modèle de Cox s'appuie sur une **vraisemblance partielle** : on maximise seulement la composante liée à l'ordre des évènements, sans modéliser explicitement $h_0(t)$.

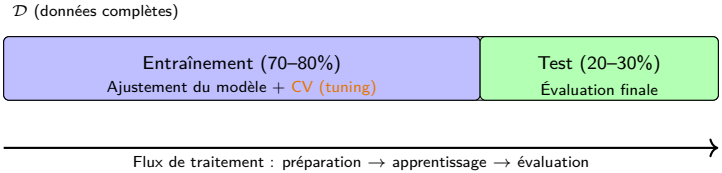
Contexte

Pour un modèle supervisé comme celui de Cox, où l'on connaît les labels $\left(D_i, T_i\right)_{i=1,\ldots,n}$ représentant respectivement le statut d'évènement ($D_i \in \{0,1\}$) et le temps observé (T_i), il est essentiel de séparer les données pour **évaluer la capacité de généralisation du modèle**.

Principe

$$\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}, \quad \mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \varnothing.$$

- $\mathcal{D}_{\text{train}}$: ajuste les paramètres $\widehat{\beta}$ du modèle de Cox ;
- $\mathcal{D}_{\text{test}}$: mesure la qualité de prédiction (C-index, Brier, calibration...).



Bonnes pratiques

Toujours stratifier selon le statut de censure (D_i) pour garantir une proportion comparable d'évènements entre train et test.

Définition

Le modèle de Cox postule une **séparabilité multiplicative** du risque instantané :

$$h(t \mid \mathbb{X}) = h_0(t)\, \exp\big(\beta^\top \mathbb{X}\big),$$

où :

- $h_0(t)$: risque de base non paramétrique (structure temporelle),
- $\exp(\beta^\top \mathbb{X})$: effet multiplicatif des covariables (structure explicative).

Définition

Le modèle de Cox postule une **séparabilité multiplicative** du risque instantané :

$$h(t \mid \mathbb{X}) = h_0(t)\, \exp\big(\beta^\top \mathbb{X}\big),$$

où :

- $h_0(t)$: risque de base non paramétrique (structure temporelle),
- $\exp(\beta^\top \mathbb{X})$: effet multiplicatif des covariables (structure explicative).

Objectif

Estimer le vecteur de coefficients β caractérisant l'effet des covariables sur le risque, **sans modéliser explicitement** $h_0(t)$.

Principe

Cox 1975 propose de maximiser une vraisemblance *partielle*, centrée sur l'ordre des évènements :

$$L_p(\beta) = \prod_{j=1}^{m} \frac{\exp(\beta^\top \mathbb{X}_{(j)})}{\sum_{i \in \mathcal{R}(t_{(j)})} \exp(\beta^\top \mathbb{X}_i)},$$

où $\mathcal{R}(t_{(j)}) = \{i \mid T_i \geq t_{(j)}\}$ contient tous les individus encore vivants (non censurés) juste avant l'instant $t_{(j)}$.

Principe

Cox 1975 propose de maximiser une vraisemblance *partielle*, centrée sur l'ordre des évènements :

$$L_p(\beta) = \prod_{j=1}^{m} \frac{\exp(\beta^\top \mathbb{X}_{(j)})}{\sum_{i \in \mathcal{R}(t_{(j)})} \exp(\beta^\top \mathbb{X}_i)},$$

où $\mathcal{R}(t_{(j)}) = \{i \mid T_i \geq t_{(j)}\}$ contient tous les individus encore vivants (non censurés) juste avant l'instant $t_{(j)}$.

Apprentissage

On cherche :

$$\widehat{\beta} = \arg\max_{\beta} \ \ell_p(\beta) \quad \text{avec} \quad \ell_p(\beta) = \log L_p(\beta).$$

Principe

Cox 1975 propose de maximiser une vraisemblance *partielle*, centrée sur l'ordre des évènements :

$$L_p(\beta) = \prod_{j=1}^{m} \frac{\exp(\beta^\top \mathbb{X}_{(j)})}{\sum_{i \in \mathcal{R}(t_{(j)})} \exp(\beta^\top \mathbb{X}_i)},$$

où $\mathcal{R}(t_{(j)}) = \{i \mid T_i \geq t_{(j)}\}$ contient tous les individus encore vivants (non censurés) juste avant l'instant $t_{(j)}$.

Apprentissage

On cherche :

$$\widehat{\beta} = \arg\max_{\beta} \ \ell_p(\beta) \quad \text{avec} \quad \ell_p(\beta) = \log L_p(\beta).$$

Interprétation

Cette vraisemblance compare, à chaque instant, le risque relatif de l'individu ayant subi l'évènement à celui des individus encore à risque.

Idée

On pénalise la norme quadratique de β pour limiter la variance et le surapprentissage :

$$\widehat{\beta} = \arg\max_{\beta} \left[\ell_p(\beta) - \lambda \|\beta\|_2^2 \right].$$

Idée

On pénalise la norme quadratique de β pour limiter la variance et le surapprentissage :

$$\widehat{\beta} = \arg\max_{\beta} \left[\ell_p(\beta) - \lambda \|\beta\|_2^2\right].$$

Propriétés

- Réduit l'amplitude des coefficients sans les annuler.
- Utile pour données multicolinéaires ou nombreuses covariables.
- Lisse les effets estimés (moindre variance).

Idée

On pénalise la norme quadratique de β pour limiter la variance et le surapprentissage :

$$\widehat{\beta} = \arg\max_{\beta} \left[\ell_p(\beta) - \lambda \|\beta\|_2^2\right].$$

Propriétés

- Réduit l'amplitude des coefficients sans les annuler.
- Utile pour données multicolinéaires ou nombreuses covariables.
- Lisse les effets estimés (moindre variance).

Implémentation

Disponible dans : `sksurv.linear_model.CoxnetSurvivalAnalysis` (Pölsterl 2020) avec $\alpha = 0$, ou `lifelines.CoxPHFitter(penalizer=…)` (Davidson-Pilon et al. 2023).

Principe

On introduit une pénalisation absolue pour effectuer une **sélection de variables** :

$$\widehat{\beta} = \arg\max_{\beta} \left[\ell_p(\beta) - \lambda \|\beta\|_1\right].$$

Principe

On introduit une pénalisation absolue pour effectuer une **sélection de variables** :

$$\widehat{\beta} = \arg\max_{\beta} \left[\ell_p(\beta) - \lambda \|\beta\|_1\right].$$

Effets

- Certaines composantes de $\widehat{\beta}$ sont nulles $\Rightarrow$ sélection automatique.
- Idéal pour jeux de données à forte dimension.

Principe

On introduit une pénalisation absolue pour effectuer une **sélection de variables** :

$$\widehat{\beta} = \arg\max_{\beta} \left[\ell_p(\beta) - \lambda \|\beta\|_1\right].$$

Effets

- Certaines composantes de $\widehat{\beta}$ sont nulles $\Rightarrow$ sélection automatique.
- Idéal pour jeux de données à forte dimension.

Packages

`skglm` (optimisé GPU, scikit-learn developers 2023)
`sksurv.linear_model.CoxnetSurvivalAnalysis` avec $\alpha = 1$ (Pölsterl 2020).

Formulation mixte

$$\widehat{\beta} = \arg\max_{\beta} \left[\ell_p(\beta) - \lambda \left(\alpha \|\beta\|_1 + \frac{1-\alpha}{2} \|\beta\|_2^2 \right) \right],$$

avec $\alpha \in [0, 1]$.

Formulation mixte

$$\widehat{\beta} = \arg\max_{\beta} \left[\ell_p(\beta) - \lambda \left(\alpha \|\beta\|_1 + \frac{1-\alpha}{2} \|\beta\|_2^2 \right) \right],$$

avec $\alpha \in [0, 1]$.

Intérêt

- Combine les effets du LASSO (sélection) et du Ridge (stabilité).
- Adapté aux données corrélées et hautement dimensionnelles.

Formulation mixte

$$\widehat{\beta} = \arg\max_{\beta} \left[\ell_p(\beta) - \lambda \left(\alpha \|\beta\|_1 + \frac{1-\alpha}{2} \|\beta\|_2^2 \right) \right],$$

avec $\alpha \in [0, 1]$.

Intérêt

- Combine les effets du LASSO (sélection) et du Ridge (stabilité).
- Adapté aux données corrélées et hautement dimensionnelles.

Implémentation

Disponible dans `sksurv.linear_model.CoxnetSurvivalAnalysis` (Pölsterl 2020).

Extension du score de risque

Au lieu du modèle linéaire

$$\eta = \beta^\top \mathbb{X},$$

on peut modéliser

$$\eta = f(\mathbb{X}),$$

où f est une fonction non linéaire apprise.

Extension du score de risque

Au lieu du modèle linéaire

$$\eta = \beta^\top \mathbb{X},$$

on peut modéliser

$$\eta = f(\mathbb{X}),$$

où f est une fonction non linéaire apprise.

Exemples

- **Réseaux de neurones** : `DeepSurv` (Katzman et al. 2018)
- **Forêts aléatoires** : `Random Survival Forests` (Ishwaran et al. 2008)
- **Splines / GAMs** : `pyGAM`, `lifelines.SplineFitter` (Servén and Brummitt 2018, Davidson-Pilon et al. 2023)

- **Cox Ridge, LASSO, Elastic Net** : `sksurv.linear_model.CoxnetSurvivalAnalysis` (bibliothèque `scikit-survival`, Pölsterl 2020)
- **Cox PH standard** : `lifelines.CoxPHFitter` (Davidson-Pilon et al. 2023)
- **DeepSurv** : `pycox.models.CoxPH` — implémentation neuronale du modèle de Cox (Katzman et al. 2018)
- **Random Survival Forests** : `sksurv.ensemble.RandomSurvivalForest` (Ishwaran et al. 2008)
- **Splines adaptatives** : `lifelines.SplineFitter`, `pyGAM` (Servén and Brummitt 2018)
- *et encore pleins d'autres...*

C-index

$$\text{C-index} = \frac{\text{nombre de paires ordonnées correctement selon le risque}}{\text{nombre total de paires comparables}}.$$

Mesure la capacité du modèle à hiérarchiser les individus selon leur risque.

AUC dépendante du temps

$$\widehat{\mathrm{AUC}}(t) = \frac{\sum_{i,j} \mathbb{1}_{\{T_i > t\}} \mathbb{1}_{\{T_j \leq t\}} \mathbb{1}_{\{\widehat{\eta}_j > \widehat{\eta}_i\}} \, \delta_j(t)}{\sum_{i,j} \mathbb{1}_{\{T_i > t\}} \mathbb{1}_{\{T_j \leq t\}} \, \delta_j(t)}.$$

Mesure la discrimination à un instant donné.

Définition

$$\mathrm{BS}(t) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \left[\mathbb{1}_{\{T_i \leq t\}} \frac{(0 - \widehat{\mathcal{S}}(t \mid \mathbb{X}_i))^2}{\widehat{G}(T_i)} \delta_i + \mathbb{1}_{\{T_i > t\}} \frac{(1 - \widehat{\mathcal{S}}(t \mid \mathbb{X}_i))^2}{\widehat{G}(t)} \right].$$

Interprétation

→ Mesure l'écart quadratique entre survie observée et prédite.

→ Plus il est faible, meilleure est la calibration.

→ Représentations complémentaires : *Calibration plots*.

- ✓ Le modèle de Cox reste la référence pour l'interprétabilité et la robustesse.
- ✓ Les pénalisations (Ridge, LASSO, Elastic Net) permettent de gérer la haute dimension.
- ✓ Les extensions non linéaires (DeepSurv, RSF, Splines) offrent plus de flexibilité au prix d'une moindre interprétabilité.
- ✓ L'évaluation doit toujours combiner :
  - → **Discrimination** (C-index, AUC),
  - → **Calibration** (Brier Score, plots),
  - → et **Validation croisée.**

L'enjeu moderne de l'analyse de survie est de concilier interprétabilité, précision et robustesse.

Cox, David Roxbee (1975). “Partial Likelihood”. In: *Biometrika* 62.2, pp. 269–276. DOI: 10.1093/biomet/62.2.269.

Davidson-Pilon, Cameron et al. (2023). *lifelines: survival analysis in Python*. https://lifelines.readthedocs.io/. Version 0.27.8.

Ishwaran, Hemant et al. (2008). “Random survival forests for R”. In: *Bioinformatics* 24.4, pp. 587–588.

Katzman, Jared L et al. (2018). “DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network”. In: *BMC medical research methodology* 18, p. 24.

Pölsterl, Sebastian (2020). “scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn”. In: *Journal of Machine Learning Research* 21.212, pp. 1–6.

scikit-learn developers (2023). *skglm: Generalized Linear Models and Survival Models with GPU Acceleration*. https://github.com/scikit-learn-contrib/skglm. scikit-learn-contrib project.

Servén, Daniel and Charlie Brummitt (2018). *pyGAM: Generalized Additive Models in Python*. https://github.com/dswah/pyGAM. GitHub repository.