Chapitre 4 : Arbres de décision pour l'analyse de survie :

Introduction.

Les arbres de décision sont un moyen simple et efficace en termes de calcul d'extraire des règles de décision simples. C'est une méthode d'apprentissage automatique qui doit toujours être utilisée comme référence, les modèles de classification plus complexes ne se justifiant que s'ils apportent une amélioration significative. Les arbres sont basés sur un partitionnement récursif des données et contrairement à la plupart des systèmes d'apprentissage, ils utilisent différents ensembles de fonctionnalités dans différentes parties de l'espace de fonctionnalités, effectuant automatiquement une sélection de fonctionnalités locales. Il s'agit d'une propriété importante et unique des algorithmes généraux de diviser pour régner qui n'a pas fait l'objet de beaucoup d'attention.

L'algorithme C4.5 pour générer des arbres est toujours la base de l'approche la plus populaire dans ce domaine. Les tests de partitionnement des données dans les arbres de décision C4.5 sont basés sur le concept d'entropie de l'information et appliqués à chaque caractéristique $x_1, x_2, \dots, x_n$ individuellement. De tels tests créent deux nœuds qui doivent d'une part contenir des données aussi pures que possible (c'est-à-dire appartenant à une seule classe), et d'autre part augmentent la séparabilité globale des données. Les tests basés directement sur des indices mesurant la précision sont optimaux du point de vue bayésien, mais ne sont pas aussi précis que ceux basés sur la théorie de l'information qui peuvent être évalués avec une plus grande précision. Le choix du test est toujours un compromis caché entre le poids accordé à la pureté des échantillons dans un ou les deux nœuds et le gain total obtenu par le partitionnement des données.

Entropies et Gain d'information :

L'entropie mesure le désordre dans les systèmes physiques, ou une quantité d'informations qui peuvent être obtenues par des observations de systèmes désordonnés.

L'entropie de Shannon :

Claude Shannon a défini une mesure formelle de l'entropie par :

$$S = -\sum_{i=1}^{n} p_i log_2(p_i)$$

Avec p_i, la probabilité d'occurrence d'un événement (valeur d'attribut) x_i un élément de l'événement X pouvant prendre des valeurs $\{x_1, \dots, x_n\}$. L'entropie de Shannon est une fonction décroissante d'une dispersion de variable aléatoire, et est maximale lorsque tous les résultats sont également probables.

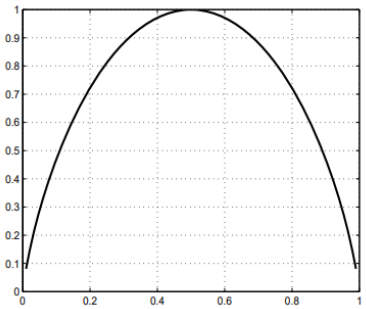

Figure 11 : Courbe de l'entropie de Shannon

L'entropie de Shannon peut être utilisée globalement, pour l'ensemble des données, ou localement, pour évaluer l'entropie des distributions de densité de probabilité autour de certains points. Cette notion

d'entropie peut être généralisée pour fournir des informations supplémentaires sur l'importance d'événements spécifiques, par exemple des valeurs aberrantes ou des événements rares. En comparant l'entropie de deux distributions, correspondant par exemple à deux caractéristiques, l'entropie de Shannon suppose implicitement un certain compromis entre les contributions des queues et la masse principale de cette distribution. Il devrait être intéressant de contrôler explicitement ce compromis, car dans de nombreux cas, il peut être important de distinguer le signal faible qui se chevauche avec un signal beaucoup plus fort. Les mesures d'entropie qui dépendent des puissances de probabilité, , fournissent un tel contrôle :

$$\sum_{i=1}^{n} p(x_i)^{\alpha}$$

Si α a une grande valeur positive, cette mesure est plus sensible aux événements qui se produisent souvent, tandis que pour un α négatif très petit, elle est plus sensible aux événements qui se produisent rarement.

Entropie de Renyi

Alfred Renyi défini l'entropie par :

$$I_{\alpha} = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{n} p_i^{\alpha} \right)$$

Cette entropie a des propriétés similaires avec celles de l'entropie de Shannon :

- Elle est additive.
- Elle atteint le maximum $\ln(n)$ pour $p = \frac{1}{n}$

L'entropie de Renyi contient un paramètre supplémentaire α qui peut être utilisé pour le rendre plus ou moins sensible à la forme des distributions de probabilité

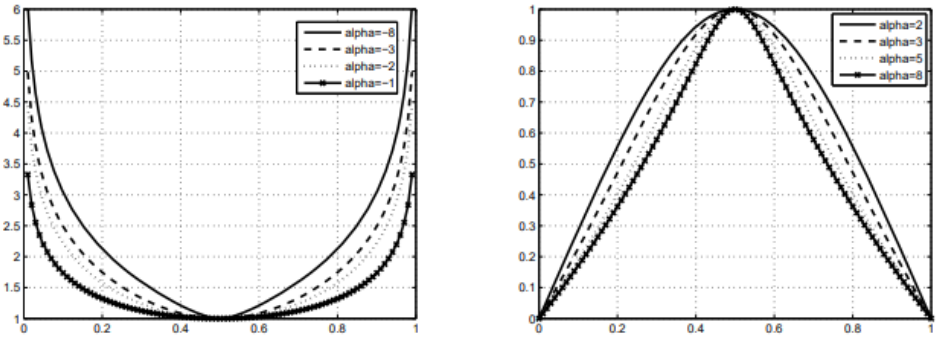

Figure 12 : Courbe de l'entropie de Renyi pour différentes valeur positive te négative de α

L'entropie de Tsallis

Constantino Tsallis définie une autre entropie par :

$$S_{\alpha} = \frac{1}{\alpha - 1} \left(1 - \sum_{i=1}^{n} p_i^{\alpha}\right)$$

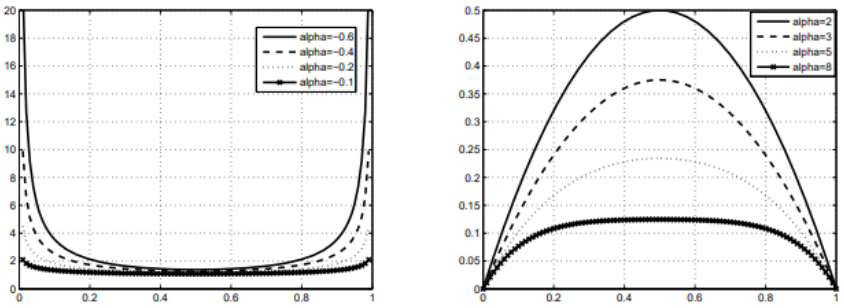

Figure 13 : Courbe d'entropie de Tsallis pour différentes valeur positive te négative de α

Gain d'information

La modification de l'algorithme standard des arbres de décision ; C4.5 a été réalisée en remplaçant simplement l'entropie de Shannon par l'une des deux autres entropies, le but étant ici d'évaluer leur influence sur les propriétés des arbres de décision. Cela signifie que le critère de découpage final est basé sur le rapport de gain : un test sur l'attribut A qui partitionne les données D en deux branches avec les données $D_g et\ D_d$ avec un ensemble de classes ω a une valeur de gain :

$$G(\omega, A/D) = H(\omega/D) - \frac{|D_g|}{|D|}H(\omega, D_g) - \frac{|D_d|}{|D|}H(\omega, D_d)$$

Avec $|D|$ le cardinal de D, et $H(\omega/D)$ l'entropie considérée.

Algorithme général d'Arbre de décision :

Nous allons décrire un algorithme d'arbre de décision générique utilisant le concept de récursivité, qui est à la base de la plupart des algorithmes d'arbre de décision décrits dans la littérature.

Nous pouvons exprimer le processus de création d'un arbre de décision sous la forme d'un algorithme récursif comme suit :

1. Choisir une caractéristique telle que lorsque le nœud parent est divisé, il en résulte un gain d'informations maximal.
2. Arrêtez si les nœuds enfants sont purs ou si aucune amélioration de la pureté de la classe ne peut être apportée.
3. Reprendre l'étape 1 pour chacun des deux nœuds enfants.

Ci-dessous une version plus formelle de cet algorithme :

GénérerArbre(D) :

- *Si* $y = 1 \forall < x; y > \in D$ *ou* $y = 0 \forall < x; y > \in D$:
  - *Return Arbre D*
- *Sinon :*
  - *Choisir la meilleure caractéristique* x_j *:*
    - D_0 à $arbre\_fils_0 : x_j = 0 \forall < x; y > \in D$
    - D_1 à $arbre\_fils_1 : x_j = 1 \forall < x; y > \in D$

    Return Nœud(x_j*, GénérerArbre(*D_0*), GénérerArbre(*D_1*))*

Il existe une variété relativement grande d'algorithmes d'arbre de décision. La plupart diffèrent des manières suivantes :

• Critères de fractionnement : information gain (Entropie de Shanon, Gini impurity, misclassification..)

- Division binaire vs divisions multivoies
- Variables discrètes ou continues
- Pré vs post élagage

Nous en discuterons quelques un des plus populaires.

ID3 – dichotomiseur itératif (Iterative Dichotomizer 3)

Il s'agit de l'un des algorithmes d'arbre de décision les plus anciens. Il prend en charge les caractéristiques discrètes, mais ne peut pas gérer les caractéristiques numériques. Il supporte la divisions binaire et multi-catégories, mais pas l'élagage, ce qui le rend sujet au surajustement (overfitting)

Cet algorithme produit des arbres courts et larges (par rapport à CART), et maximise le gain d'information (minimise l'entropie)

Principe général

ID3 construit l'arbre de décision récursivement. À chaque étape de la récursion, il calcule parmi les attributs restant pour la branche en cours, celui qui maximisera le gain d'information. C'est-à-dire l'attribut qui permettra le plus facilement de classer les exemples à ce niveau de cette branche de l'arbre et minimisera l'entropie.

Algorithme

Fonction ID3(exemples, attributCible, attributsNonCibles) :

Si exemples est vide alors / Nœud terminal */ :*

Retourner un nœud Erreur

Sinon si attributsNonCibles est vide alors / Nœud terminal */ :*

Retourner un nœud ayant la valeur la plus représentée pour attributCible

Sinon si tous les exemples ont la même valeur pour attributCible alors / Nœud terminal */ :*

Retourner un nœud ayant cette valeur

Sinon / Nœud intermédiaire */ :*

attributSélectionné = attribut maximisant le gain d'information parmi attributsNonCibles

attributsNonCiblesRestants = suppressionListe(attributsNonCibles, attributSélectionné)

nouveauNœud = nœud étiqueté avec attributSélectionné

Pour chaque valeur de attributSélectionné faire :

exemplesFiltrés = filtreExemplesAyantValeurPourAttribut(exemples, attributSélectionné, valeur)

nouveauNœud->fils(valeur) = ID3(exemplesFiltrés, attributCible, attributsNonCiblesRestants)

finpour

Retourner nouveauNœud

C4.5 :

Décrit dans Quinlan, J. R. (1993). C4. 5 : Programmation pour l'apprentissage automatique. Kauffmann, 38, 48. Morgan

• Fonctionnalités continues et discrètes (le fractionnement continu des fonctionnalités est très coûteux car il doit prendre en compte tous les seuils possibles)

• Le critère de fractionnement est calculé via le rapport de gain

• Gère les attributs manquants (les ignore dans le calcul du gain d'informations)

• Effectue la post-élagage (élagage ascendante)

C4.5 est un algorithme utilisé pour générer un arbre de décision développé par Ross Quinlan . C4.5 est une extension de l'algorithme ID3. Les arbres de décision générés par C4.5 peuvent être utilisés pour la classification, et pour cette raison, C4.5 est souvent appelé classificateur statistique. En 2011, les auteurs du logiciel d'apprentissage automatique Weka ont décrit l'algorithme C4.5 comme « un programme d'arbre de décision historique qui est probablement le cheval de bataille d'apprentissage automatique le plus largement utilisé dans la pratique à ce jour ».

Algorithme :

C4.5 construit des arbres de décision à partir d'un ensemble de données d'apprentissage de la même manière que ID3, en utilisant le concept d'entropie de gain d'information. Les données d'apprentissage sont un ensemble d'échantillons déjà classés. Chaque échantillon se compose d'un vecteur p-dimensionnel, où les valeurs d'attributs ou les caractéristiques de l'échantillon sont représentées, ainsi que la classe dans laquelle elles se trouvent. $S = s_1, s_2, \ldots\ldots, s_i\ (x_{1,i}, x_{2,i}, \ldots\ldots, x_{p,i})x_j, s_i$

A chaque nœud de l'arbre, C4.5 choisit l'attribut des données qui divise le plus efficacement son ensemble d'échantillons en sous-ensembles enrichis dans l'une ou l'autre classe. Le critère de découpage est le gain d'information normalisé (différence d'entropie). L'attribut avec le gain d'informations normalisé le plus élevé est choisi pour prendre la décision. L'algorithme C4.5 revient alors sur les sous-listes partitionnées.

Cet algorithme a quelques cas de base :

- Tous les échantillons de la liste appartiennent à la même classe. Lorsque cela se produit, il crée simplement un nœud feuille pour l'arbre de décision en disant de choisir cette classe.
- Aucune des fonctionnalités ne fournit de gain d'informations. Dans ce cas, C4.5 crée un nœud de décision plus haut dans l'arbre en utilisant la valeur attendue de la classe.
- Instance de classe inédite rencontrée. Encore une fois, C4.5 crée un nœud de décision plus haut dans l'arbre en utilisant la valeur attendue

Pseudocode

En pseudocode, l'algorithme général pour construire des arbres de décision est :

1) Vérifier les cas de base ci-dessus.
2) Pour chaque attribut a , trouver le rapport de gain d'informations normalisé provenant de la division sur a .
3) Soit a_{best} l'attribut avec le gain d'informations normalisé le plus élevé.
4) Créer un nœud de décision qui se divise sur a_{best}.

5) Récurer sur les sous-listes obtenues en divisant sur a_{best}, et ajouter ces nœuds en tant qu'enfants de nœuds

Améliorations de l'algorithme ID.3

C4.5 a apporté un certain nombre d'améliorations à ID3. Certaines d'entre elles sont :

- Gestion des attributs continus et discrets - Afin de gérer les attributs continus, C4.5 crée un seuil, puis divise la liste en ceux dont la valeur d'attribut est supérieure au seuil et ceux qui lui sont inférieurs ou égaux.
- Gestion des données d'entraînement avec des valeurs d'attribut manquantes - C4.5 permet aux valeurs d'attribut d'être marquées comme ? pour disparu. Les valeurs d'attribut manquantes ne sont tout simplement pas utilisées dans les calculs de gain et d'entropie. Ceci pourrait servir notre sujet sur 'analyse de survie)
- Gestion des attributs avec des coûts différents.
- Élagage des arbres après la création - C4.5 parcourt l'arbre une fois qu'il a été créé et tente de supprimer les branches qui n'aident pas en les remplaçant par des nœuds feuilles.

CART :

Décrit dans Breiman, L. (1984). Wadsworth International Group. Arbres de classification et de régression. Belmont, Californie :

• Caractéristiques continues et discrètes

• Que des questions-tests binaires (les arbres résultants sont plus grands par rapport à ID3 et C4.5)

• Les fractionnements binaires peuvent générer de meilleurs arbres que C4.5, mais ont tendance à être plus grands et plus difficiles à interpréter. C'est-à-dire que pour k attributs, nous avons $(2^{k-1} - 1)$ façons de créer un partitionnement binaire

• Réduction de la variance dans les arbres de régression

• Utilise l'impureté de Gini dans les arbres de classification

• Effectue un élagage de la complexité des coûts

Quand un nœud interne S est coupé sur l'attribut j, seuil a_i, il donne naissance à deux descendants :

Sous-nœud gauche $S_g(p_g \approx \frac{|S_g|}{|S|})$ qui contient tous les éléments qui ont les valeurs de l'attribut $v_j < a_j$,

Sous-nœud droit $S_d(p_d \approx \frac{|S_d|}{|S|})$ qui contient tous les éléments qui ont les valeurs de l'attribut $v_j \geq a_j$.

Soit $I(S)$ une fonction qui mesure l'impureté de S par rapport à la classe cible. CART étudie le changement de l'impureté par rapport au seuil et pour tous les attributs :

- $E[I(S_{gd})] = p_g I(S_g) + p_d I(S_d)$ Ou E[·] est l'opérateur de moyenne statistique,
- $\Delta I(S) = I(S) - E[I(S_{gd}) = I(S) - p_g I(S_g) - p_d I(S_d)$.

Le problème d'optimisation est le suivant :

$$\underset{j,a_j}{argmax} \Delta I(S)$$

CART choisit donc l'attribut et le seuil qui maximisent la décroissance de l'impureté du nœud par rapport à la cible.

En classification (classement) la mesure de l'impureté utilisée est l'index (ou impureté) de Gini qui est la vraisemblance qu'un élément du nœud soit incorrectement étiqueté par un tirage aléatoire qui respecte la loi statistique de la cible estimée dans le nœud.

L'impureté (ou l'index de Gini $I_G(S)$) pour un nœud S est calculée comme suit :

- Partitionner S sur les valeurs de la cible en n groupes : $C_1,...,C_n$,
- Calculer p_i : probabilité estimée qu'un élément de S se retrouve dans $C_i (p_i \approx \frac{|C_i|}{|S|})$,
- $I_G(S) = \sum_{i=1}^{m} p_i(1-p_i) = \sum_{i=1}^{m}(p_i - {p_i}^2) = 1 - \sum_{i=1}^{m} {p_i}^2$
- $I_G(S) = \sum_{i \neq j} p_i p_j$ Index de Gini,
- $I_G(S) = 0$ Si S est homogène (tous les éléments sont dans la même classe, donc impureté du groupe nulle).

Toujours en classification on peut utiliser d'autres types de mesures d'impureté :

- $H(p) = -\sum_i p_i log_2(p_i)$ (Entropie),

$E(S) = 1 - {}^{max}_{i}(p_i)$ (Erreur de classification).

Arbres de décision optimaux :

Optimal Trees est une nouvelle approche pour la construction d'arbres de décision qui surpasse considérablement les méthodes d'arbre de décision existantes (Bertsimas et Dunn, 2019). Elle formule le problème de construction d'arbre de décision du point de vue de l'optimalité globale en utilisant l'optimisation en nombres entiers mixtes (MIO), et résout le problème de gourmandises des approches précédentes avec une descente de coordonnées pour trouver des solutions optimales ou quasi optimales dans des temps d'exécution pratiques. Ces arbres optimaux sont souvent aussi puissants que les méthodes de pointe comme les forêts aléatoires ou les arbres boostés, mais ils ne sont qu'un arbre de décision unique et sont donc facilement interprétables. Cela évite d'avoir à faire un compromis entre interprétabilité et précision de pointe lors du choix d'une méthode prédictive.

Les arbres optimaux entraînent de manière flexible et efficace les arbres de décision selon une fonction de perte de la forme :

$$min_T(erreur(T, D) + \alpha.complexité(T))$$

Avec T l'arbre à optimiser, D les données d'entrainement, et erreur(T,D) est une fonction mesurant à quel point T s'ajuste aux données D, et complexité(T) une fonction pénalisant la complexité de l'arbre T. et α est le paramètre de complexité qui contrôle le compromis entre la qualité de l'ajustement et la taille de l'arbre.

L'algorithme des arbres optimaux est capable de s'adapter à de grands ensembles de données (n en millions, p en milliers) en utilisant la descente de coordonnées pour entraîner les arbres de décision vers l'optimalité globale. Lors de la formation d'un arbre, les divisions de l'arbre sont optimisées à plusieurs reprises une par une, en trouvant des changements qui minimise la fonction de perte. Pour donner un aperçu de haut niveau, les nœuds de l'arbre sont visités dans un ordre aléatoire et à chaque nœud, nous considérons les modifications suivantes :

- Si le nœud n'est pas une feuille, supprimer la division de ce nœud.
- Si le nœud n'est pas une feuille, trouver la division optimale à utiliser à ce nœud et mettre à jour la division actuelle.
- Si le nœud est une feuille, créez une nouvelle division de ce nœud

Pour chacun des changements, nous calculons la valeur objective de l'arbre modifié par rapport à la minimisation de la fonction de perte. Si l'un de ces changements entraîne une amélioration de la valeur de l'objectif, la modification est alors acceptée. Lorsqu'une modification est acceptée ou que toutes les modifications potentielles ont été rejetées, l'algorithme procède à la visite d'autres nœuds de l'arbre dans un ordre aléatoire jusqu'à ce qu'aucune autre amélioration ne soit trouvée, ce qui signifie que cet arbre est localement optimal par rapport à la fonction de perte. Ce problème n'est pas convexe, nous répétons donc le processus de descente de coordonnées à partir de divers arbres de décision de départ générés aléatoirement, avant de sélectionner l'arbre final localement optimal avec la valeur d'objectif globale la plus basse comme meilleure solution

Arbres de survie :

Les méthodes d'arbres de décision ont reçu beaucoup d'attention dans la littérature, la méthode la plus importante étant l'algorithme d'arbre de classification et de régression (CART). Cependant, les algorithmes d'arbre traditionnels nécessitent des observations complètes de la variable dépendante dans les données d'apprentissage, ce qui les rend inadaptés aux données censurées.

Les algorithmes d'arborescence incorporent une règle de fractionnement qui sélectionne les partitions à ajouter à l'arborescence, et une règle d'élagage détermine quand arrêter d'ajouter d'autres partitions. Les règles de fractionnement dans les arbres de survie sont généralement basées soit sur :

- Des mesures de distance des nœuds qui cherchent à maximiser la déférence entre les observations dans des nœuds séparés.
- Soit des mesures de pureté des nœuds qui cherchent à regrouper des observations similaires dans un seul nœud.

Les algorithmes basés sur les mesures de distance des nœuds comparent les deux nœuds enfants adjacents qui sont générés lorsqu'un nœud parent est divisé, en retenant la division qui produit la plus grande déférence dans les nœuds enfants. Les mesures proposées de la distance des nœuds comprennent le test du logrank à deux échantillons, la statistique du rapport de vraisemblance et les tests de permutation d'inférence conditionnelle. Nous notons que la fonction de score utilisée dans les modèles de régression de Cox appartient également à la classe des mesures de distance des nœuds, car la statistique de vraisemblance partielle est basée sur une comparaison du coefficient de risque relatif prédit pour chaque observation.

Les règles de fractionnement basées sur la dissimilitude ne conviennent pas à certaines applications (telles que l'algorithme des arbres optimaux) car elles ne permettent pas l'évaluation d'un seul nœud isolément. Nous allons donc nous concentrer sur les règles de séparation de la pureté des nœuds pour développer l'algorithme OST (Optimal Survival Tree).

Il existe des algorithmes d'arbre de survie avec une règle de séparation de la pureté des nœuds basée sur les estimations de Kaplan-Meier. D'autres utilisent une règle de partage basée sur la log-vraisemblance négative d'un modèle exponentiel, ou les résidus de martingale comme estimation de l'erreur de nœud. D'autres algorithmes ont proposé de comparer la log-vraisemblance d'un modèle saturé à la première étape d'une procédure d'estimation de vraisemblance complète pour le modèle à risques proportionnels et ont montré que la vraisemblance complète et les résidus de martingale peuvent être calculés efficacement à partir de l'estimateur de risque cumulatif de Nelson-Aalen. Plus récemment, Molinaro et al. (2004) ont proposé une nouvelle approche pour ajuster les fonctions de perte pour les données non censurées basées sur la probabilité inverse des poids de censure (IPCW).

La plupart des algorithmes d'arbre de survie utilisent l'élagage par coût de complexité pour déterminer la taille correcte de l'arbre, en particulier lorsque les nœuds sont fractionnés basé sur leurs. Ces méthodes sélectionnent un arbre qui minimise une combinaison pondérée de l'erreur totale de l'arbre (c'est-à-dire la somme de chaque erreur de nœud feuille) et de la complexité de l'arbre (le nombre de nœuds feuilles), avec des poids relatifs déterminés par validation croisée. Une méthode similaire de fractionnement de nœuds basé sur élagage par complexité divisée a été suggérée par LeBlanc et Crowley (1993) pour les mesures de distance des nœuds, en utilisant la somme des tests statistiques de division et le nombre de divisions dans l'arbre. D'autres propositions incluent l'utilisation du critère d'information d'Akaike (AIC) (Ciampi et al., 1986) ou l'utilisation d'un critère d'arrêt de la p-valeur pour arrêter la croissance de l'arbre lorsqu'aucune autre division significative n'est trouvée (Hothorn et al., 2006).

Algorithme d'arbre de survie :

Le modèle d'arbre optimal a été adapté pour l'analyse de survie avec le modèle Arbre de Survie Optimal OST (Optimal Survival Tree). Comme d'autres algorithmes d'arbre, le modèle OST nécessite une fonction cible qui détermine les divisions à ajouter à l'arbre. L'efficacité de calcul est un facteur important dans le choix de la fonction cible, puisqu'elle doit être réévaluée pour chaque changement

potentiel de l'arbre pendant les procédures d'optimisation. Une exigence clé pour la fonction cible est que « l'ajustement » ou l'erreur de chaque nœud doit être évalué indépendamment du reste de l'arbre. Dans ce cas, la modification d'une division particulière dans l'arbre ne nécessitera que la réévaluation du sous-arbre directement en dessous de cette division, plutôt que l'arbre entier. Cette exigence limite le choix de la fonction cible aux approches de pureté de nœud.

La règle de partage implémentée dans l'algorithme OST est basée sur la méthode de vraisemblance. Cette règle de répartition est dérivée d'un modèle à risques proportionnels qui suppose que la distribution de survie sous-jacente pour chaque observation est donnée par

$$P(S_i \leq t) = 1 - e^{-\theta_i \Lambda(t)},$$

Avec $\Lambda(t)$, la fonction de risque cumulé de référence et les coefficients θ_i sont les ajustements du risque cumulé de référence pour chaque observation

Dans le modèle d'arbre de survie, nous remplaçons $\Lambda(t)$ par une estimation empirique de la probabilité cumulée de décès à chacun des moments d'observation en utilisant l'estimateur de Nelson-Aalen.

$$\widehat{\Lambda}(t) = \sum_{i:t_i \leq t} \frac{\delta_i}{\sum_{j:t_j \geq t} 1}$$

Avec ce risque de base, l'objectif du modèle d'arbre de survie est d'optimiser les coefficients de risque θ_i. Nous imposons que le modèle d'arbre utilise le même coefficient pour toutes les observations contenues dans un nœud feuille donné de l'arbre, c'est-à-dire $\theta_i = \hat{\theta}_{T(i)}$. Ces coefficients sont déterminés en maximisant la vraisemblance de l'échantillon intra-feuille

$$L = \prod_{i=1}^{n} (\theta_i \frac{d}{dt} \Lambda(t_i))^{\delta_i} e^{-\theta_i \Lambda(t_i)},$$

Pour obtenir les coefficients du nœud,

$$\hat{\theta}_i = \frac{\sum_i \delta_i I_{\{T_i = k\}}}{\sum_i \widehat{\Lambda}(t_i) I_{\{T_i = k\}}},$$

Pour évaluer dans quelle mesure différents découpages s'adaptent aux données disponibles, nous comparons le modèle d'arbre actuel à un arbre avec un seul coefficient pour chaque observation. Nous appellerons cela un arbre entièrement saturé, car il a un paramètre unique pour chaque observation. Les estimations du maximum de vraisemblance pour ces coefficients de modèle saturés sont :

$$\hat{\theta}_i^{sat} = \frac{\delta_i}{\widehat{\Lambda}(t_i)}, \qquad i = 1, \dots, n.$$

Nous calculons l'erreur de prédiction à chaque nœud comme la déférence entre la log-vraisemblance pour le coefficient du nœud ajusté et les coefficients du modèle saturé à ce nœud :

$$erreur_k = \sum_{i:T(i)=k} (\delta_i \log (\frac{\delta_i}{\widehat{\Lambda}(t_i)}) - \delta_i \log(\hat{\theta}_k) - \delta_i + \widehat{\Lambda}(t_i)\hat{\theta}_k).$$

La fonction d'erreur globale utilisée pour optimiser l'arbre est simplement la somme des erreurs sur les nœuds feuilles de l'arbre T compte tenu des données d'apprentissage D :

$$erreur(T, D) = \sum_{k \in feuilles(T)} erreur_k(D).$$

Nous pouvons ensuite appliquer l'approche des arbres optimaux pour former un arbre selon cette fonction d'erreur en substituant cette expression dans la fonction de perte globale. À chaque étape du processus de descente de coordonnées, nous déterminons de nouvelles estimations de $\hat{\theta}_k$ pour chaque nœud feuille k dans l'arbre. Nous calculons et additionnons ensuite les erreurs à chaque nœud pour obtenir l'erreur totale de la solution actuelle, qui est utilisée pour guider la descente des coordonnées et générer des arbres qui minimisent l'erreur.

Métriques de précision de l'arbre de survie :

Des mesures de précision ont été introduites afin d'évaluer les performances de l'algorithme OST pour les modèles d'arbre de survie. Nous utiliserons la notation T pour représenter un modèle d'arbre, où $T_i = T(X_i)$ est la classification du nœud feuille de l'observation i avec les covariables X_i dans l'arbre T. Nous utiliserons la notation T^0 pour représenter un modèle nul (un arbre avec pas de divisions et un seul nœud).

La statistique de concordance :

L'application d'une approche de classement à l'analyse de la survie est un moyen efficace de traiter les distributions asymétriques des temps de survie ainsi que la censure des données. La statistique de concordance, qui est la plus connue de la régression logistique, est une autre métrique populaire qui a été adaptée pour mesurer la qualité de l'ajustement dans les modèles de survie (Harrell et al., 1982). L'indice de concordance est défini comme la proportion de toutes les paires d'observations comparables dans lesquelles les prédictions du modèle sont concordantes avec les résultats observés. Deux observations sont comparables si l'on sait avec certitude qu'un individu est mort avant l'autre. Cela se produit lorsque l'heure réelle du décès est observée pour les deux individus (aucun n'est censuré) ou lorsque le décès d'un individu est observé avant que l'autre ne soit censuré. Une paire comparable est concordante si le risque prédit est plus élevé pour l'individu qui est décédé en premier, et la paire est discordante si le risque prédit est plus faible pour l'individu qui est décédé en premier. Ainsi, le nombre de paires concordantes dans un échantillon est donné par :

$$CC = \sum_{i,j} \mathbb{1}(t_i < t_j)\mathbb{1}(\theta_i > \theta_j)\delta_i$$

et le nombre de paires discordantes est :

$$DC = \sum_{i,j} \mathbb{1}(t_i < t_j)\mathbb{1}(\theta_i < \theta_j)\delta_i$$

Où les indices i et j font référence à des paires d'observations dans l'échantillon. La multiplication par le facteur δ_i élimine les paires d'observations qui ne sont pas comparables parce que le plus petit temps de survie est censuré, c'est-à-dire δ_i= 0. Ces définitions n'incluent pas les paires comparables avec des prédictions de risque liées, nous notons donc ces paires comme :

$$TR = \sum_{i,j} \mathbb{1}(t_i < t_j)\mathbb{1}(\theta_i = \theta_j)\delta_i$$

Le nombre de paires concordantes et discordantes est généralement résumé à l'aide de l'indice C de Harrell (Harrell et al., 1982),

$$H_C = \frac{CC + 0.5 \times TR}{CC + DC + TR}$$

L'indice C de Harrell prend des valeurs comprises entre 0 et 1, des valeurs plus élevées indiquant un meilleur ajustement.

Uno et al. (2011) ont introduit une statistique C non paramétrique,

$$U_{C_\tau} = \frac{\sum_{i,j}(\hat{G}(t_i))^{-2}\mathbb{1}(t_i < t_j, t_i < \tau)\mathbb{1}(\theta_i > \theta_j)\delta_i}{\sum_{i,j}(\hat{G}(t_i)^{-2}(\mathbb{1}(t_i < t_j, t_i < \tau)(\theta_i > \theta_j)\delta_i + \mathbb{1}(t_i < t_j, t_i < \tau)(\theta_i \leq \theta_j)\delta_i)}$$

Où $\hat{G}(\cdot)$ est l'estimation de Kaplan-Meier pour la distribution de censure. Grâce à ces coefficients, U_C converge vers une quantité indépendante de la distribution de censure. U_C prend des valeurs comprises entre 0 et 1, les valeurs les plus élevées indiquant un meilleur ajustement.

Il est important de noter que les métriques décrites ci-dessus ne sont pas spécifiquement conçues pour les arbres de survie et présentent donc certaines limites lorsqu'elles sont appliquées dans ce contexte. Le score de vraisemblance partielle de Cox et la C-statistique deviennent moins informatifs lorsqu'un grand nombre d'observations ont le même coefficient de risque prédit, ce qui est généralement le cas dans les modèles d'arbres. L'augmentation du nombre de nœuds dans l'arbre peut gonfler ces scores même si la qualité globale du modèle ne s'améliore pas.

Score Brier intégré

La métrique du score de Brier est couramment utilisée pour évaluer les arbres de classification (Brier, 1950). Il a été développé à l'origine pour vérifier l'exactitude d'une prévision de probabilité, principalement destinée aux prévisions météorologiques. La formule la plus courante calcule l'erreur quadratique moyenne de prédiction :

$$B = \frac{1}{n}\sum_{i}^{n}(\hat{P}(y_i) - y_i)^2$$

Où n est la taille de l'échantillon, $y_i \in \{0,1\}$ est le résultat de l'observation i et $\hat{p}(y_i)$ est la probabilité de prévision de ce résultat observé. Dans le contexte de l'analyse de survie, le score de Brier peut être utilisé pour évaluer l'exactitude des prédictions de survie à un moment donné par rapport aux décès observés à ce moment-là. Nous appellerons cela le Brier Point Score :

$$BP_\tau = \frac{1}{|\mathfrak{T}_\tau|}\sum_{i\epsilon\mathfrak{T}_\tau}(\hat{S}_i(\tau) - \mathbb{1}(t_i > \tau))^2$$
$$o\grave{u}\ , \mathfrak{T}_\tau = \{i\epsilon\{1, \ldots., n\}, |t_i \geq \tau\ ou\ \delta_i = 1\}$$

$\hat{S}_i(\tau)$ est la probabilité de survie prédite pour l'observation i au temps τ et $\mathfrak{T}_\tau$ est l'ensemble des observations connues pour être vivantes/mortes au temps τ. Les observations censurées avant l'heure τ sont exclues de ce score, car leur statut de survie est inconnu.

L'application de cette version du score Brier peut être utile dans les applications où le principal résultat d'intérêt est la survie à un moment donné, comme les taux de survie à un an après le début d'une maladie. On peut évaluer, le score de Brier ponctuel au moment d'observation médian dans chaque ensemble de données. Et pour faciliter l'interprétation, les scores rapportés sont normalisés par rapport au score d'un modèle nul, c'est-à-dire :

$$BPR_\tau = 1 - \frac{BP_\tau(T)}{BP_\tau(T^\circ)}$$

Le score Brier Point présente deux inconvénients importants dans l'analyse de survie. Tout d'abord, il évalue la précision prédictive des modèles de survie à un seul point dans le temps plutôt que sur toute la période d'observation, ce qui n'est pas bien adapté aux applications où les distributions de survie sont le

résultat d'intérêt. Deuxièmement, il devient moins informatif à mesure que le nombre d'observations censurées augmente, car un plus grand nombre d'observations sont rejetées lors du calcul du score.

Graf et al. (1999) ont relevé ces défis en proposant une version ajustée du Brier Score pour les ensembles de données de survie avec des résultats censurés. Plutôt que de mesurer la précision des prédictions de survie en un seul point, cette mesure agrège le score de Brier sur tout l'intervalle de temps observé dans les données. Cette mesure modifiée est couramment utilisée dans la littérature sur la survie et a été indifféremment appelée Brier Score ou Integrated Brier Score par divers auteurs (Reddy et Kronek, 2008). Ce score de Brier intégré (IB), défini comme :

$$IB = \frac{1}{n}\frac{1}{t_{max}}\sum_{i=1}^{n}\int_0^{t_i}\frac{(1-\hat{S}_i(t))^2}{\hat{G}_i(t)}dt + \delta_i\int_{t_i}^{t_{max}}\frac{(\hat{S}_i(t))^2}{\hat{G}_i(t_i)}dt.$$

Le score IB utilise des estimations de Kaplan-Meier pour la distribution de survie, $\hat{S}$(t), et la distribution de censure, $\hat{G}$(t). Dans un modèle d'arbre de survie, ces estimations sont obtenues en regroupant les observations dans chaque nœud de l'arbre, c'est-à-dire, $\hat{S}(t) = \hat{S}_{T_i}(t)$. Le score IB est une version pondérée du score Brier original, les pondérations étant $1/\hat{G}_i(t_i)$ si un événement se produit avant le temps t_i, et $1/\hat{G}_i(t)$ si l'événement se produit après le temps t.

Nous rapportons le ratio des scores Brier intégrés (IBR), qui compare la somme des scores Brier intégrés dans un arbre donné aux scores Brier intégrés correspondants dans un arbre nul :

$$IBR = 1 - \frac{IB(T)}{IB(T^{\circ})}$$

Nous notons que toutes les mesures ci-dessus ont certaines limites et ne fournissent pas de preuve définitive qu'un modèle est meilleur qu'un autre. En pratique, ces métriques fournissent souvent des évaluations contradictoires lors de la comparaison de déférents modèles d'arbres.

Ces limitations rendent difficile l'obtention d'une comparaison sans ambiguïté entre les performances des déférents algorithmes d'arbre de survie. Pour relever ce défi, nous allons maintenant introduire une procédure de simulation et des mesures de précision associées spécialement conçues pour évaluer les modèles d'arbres de survie.

Métriques de précision de la classification :

Soit i = 1,...,n un ensemble d'observations avec des covariables indépendantes et identiquement distribuées $X_i = (X_{ij})_{j=1}^{m}$. Soit C un modèle d'arbre qui partitionne les observations en fonction de ces covariables de sorte que $C_i = C(X_i)$ est l'indice du nœud feuille de C qui contient l'individu i. Soit S_i une variable aléatoire représentant le temps de survie de l'observation i, de distribution $S_i \sim F_{C_i}(t)$. La distribution de survie de chaque individu est entièrement déterminée par son emplacement dans l'arbre C, et nous nous référons donc à C comme le « vrai » modèle d'arbre.

Nous mesurons la précision de la classification d'un modèle d'arbre empirique (T) par rapport au véritable arbre (C) à l'aide des métriques suivantes :

1. **Homogénéité des nœuds**
   La statistique d'homogénéité des nœuds mesure la proportion d'observations dans chaque nœud $k \epsilon T$ qui ont la même vraie classe dans C. Soit $p_{k,l}$ la proportion d'observations dans le nœud $k \epsilon T$ qui proviennent de la classe $l \epsilon C$ et soit $n_{k,l}$ le nombre total d'observations au nœud $k \epsilon T$ de la classe lϵC. Alors,

$$NH = \frac{1}{n}\sum_{k \epsilon T}\sum_{l \epsilon C} n_{k,l}\, p_{k,l}$$

Un score de NH = 1 indique que chaque nœud du nouveau modèle d'arbre contient des observations d'une seule classe dans C. Cela ne signifie pas nécessairement que la structure de T est identique à C. Par exemple, un arbre saturé avec une seule observation dans chaque nœud aurait un score d'homogénéité de nœud parfait. La métrique d'homogénéité des nœuds est donc biaisée vers des modèles d'arbres plus grands avec peu d'observations dans chaque nœud.

2. **Récupération de classe**

   La récupération de classe est une mesure de la capacité d'un nouveau modèle d'arbre à conserver des observations similaires ensemble dans le même nœud, évitant ainsi des fractionnements inutiles. La récupération de classe est calculée en comptant la proportion d'observations d'une vraie classe l ϵ C qui sont placées dans le même nœud de T. Soit $q_{k,l}$ la proportion d'observations de la classe l ϵ C qui sont classées dans le nœud k ϵ T et soit $n_{k,l}$ le nombre total d'observations au nœud k ϵ T de la classe l ϵ C. Alors :

$$CR = \frac{1}{n}\sum_{l \epsilon c}\sum_{k \epsilon T} n_{k,l}\, q_{k,l}$$

   Cette métrique est biaisée en faveur des arbres plus petits, car un arbre nul avec un seul nœud aurait un score de récupération de classe parfait. Il est donc utile de considérer simultanément les scores de récupération de classe et d'homogénéité des nœuds afin d'évaluer les performances d'un modèle d'arbre (voir la figure 2 pour des exemples). Lorsqu'elles sont utilisées ensemble, ces métriques indiquent dans quelle mesure le modèle T reflète la structure du vrai modèle C.

Les scores d'homogénéité des nœuds et de récupération de classe peuvent également être utilisés pour comparer deux modèles d'arbre, T1 et T2. Dans ce cas, ces métriques doivent être interprétées comme une mesure de la similarité structurelle entre les deux modèles d'arbre. Il faut noter que lorsque T1 et T2 sont appliqués au même jeu de données, l'homogénéité des nœuds pour le modèle T1 par rapport à T2 est équivalente à la récupération de classe pour T2 par rapport à T1, et vice versa. Le score moyen d'homogénéité des nœuds pour T1 et T2 est donc égal au score moyen de récupération de classe pour T1 et T2. Ce score est appelé score de similarité pour les modèles T1 et T2.

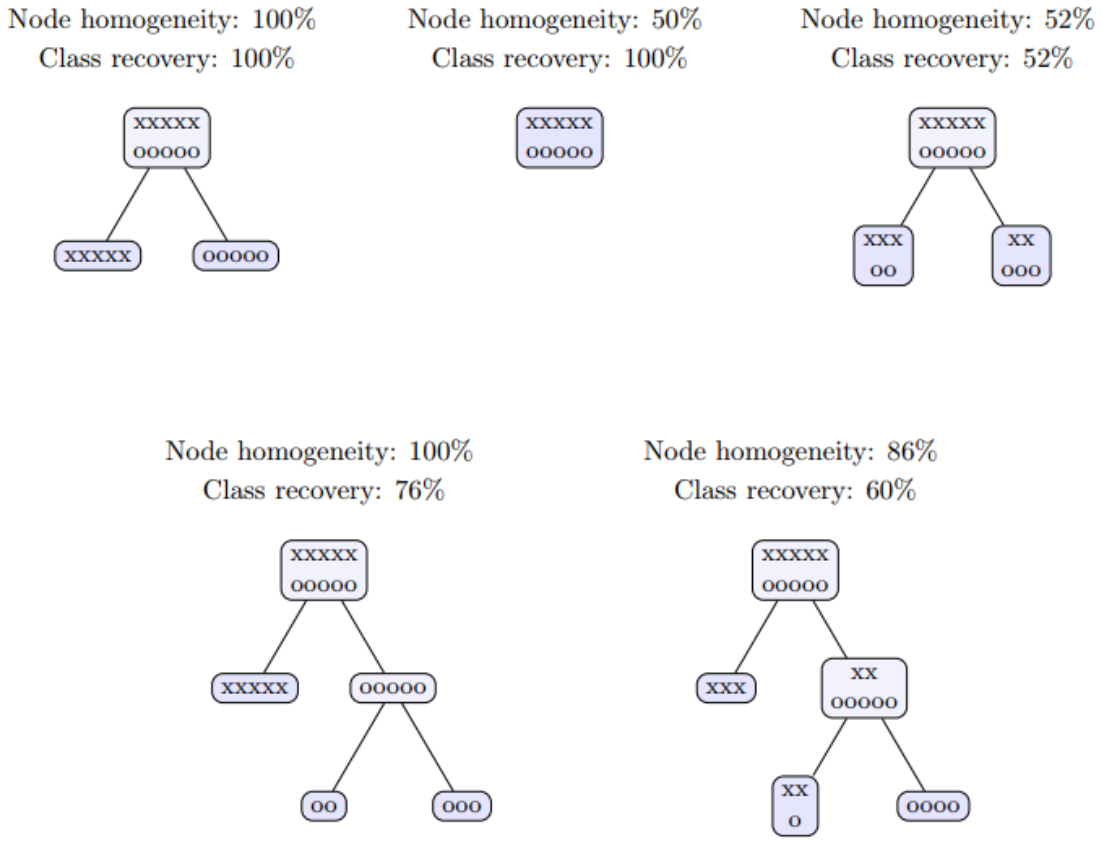

Figure 14 : homogénéité et score de similarité pour les modèles

Métrique de précision des prédictions :

La métrique de précision de prédiction mesure à quel point les courbes de Kaplan-Meier non paramétriques à chaque feuille de T estiment correctement la distribution de survie de chaque observation.

Aire entre les courbes (ABC) :

Pour une observation i avec une vraie distribution de survie $F_{C_i}(t)$, supposons que $\hat{S}_{T_i}(t)$ est l'estimation de Kaplan-Meier au nœud correspondant dans l'arbre T (voir Figure 3). L'aire entre la vraie courbe de survie et l'estimation de l'arbre est donnée par :

$$ABC_i^T = \frac{1}{T_{max}} \int_0^{t_{max}} \left|1 - F_{C_i}(t) - \hat{S}_{T_i}(t)\right| dt.$$

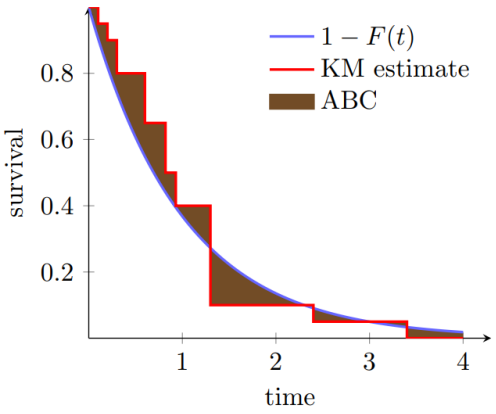

Figure 15 : Surface entre les courbes.

Pour faciliter l'interprétation de cette métrique, on compare l'aire entre les courbes d'un arbre donné au score d'un arbre nul avec un seul nœud (T_0). Le rapport de surface (AR) est donné par :

$$AR = 1 - \frac{\sum_i ABC_i^T}{\sum_i ABC_i^{T^\circ}}$$

Semblable à la métrique R^2 populaire pour les modèles de régression, l'AR indique le degré de précision obtenu en utilisant les estimations de Kaplan-Meier générées par l'arbre par rapport à la précision de base obtenue en utilisant une seule estimation pour l'ensemble de la population.