Projet FTML

Grégoire GALLY, Lucas PINOT, Joseph DURAND

June 17, 2022

1 Bayes estimator and Bayes risk

1.1 Question 1

Soit $x : \{0, 1, 2\}$ et $y : \{0, 1\}$.

$$X \rightsquigarrow U([0,1]) + B(\frac{3}{4})$$
$$Y \rightsquigarrow \begin{cases} B(\frac{2}{3}) \ si \ X = 0 \\ B(\frac{3}{5}) \ si \ X = 1 \\ B(\frac{1}{4}) \ si \ X = 2 \end{cases}$$

Avec B(p) une loi de bernoulli de paramètre p.
La loss function l est une "0-1" loss.

Le classificateur bayesien est défini par :

$$f^*(x) = \arg\min_{z \in Y} \mathbb{E}\big[l(Y, z)|X = x\big]$$

Ainsi :

$$\begin{aligned} f^*(x) &= \arg\min_{z \in y} \mathbb{E}\big[l(Y, z)|X = x\big] \\ &= \arg\min_{z \in y} \mathbb{P}\big(Y \neq z|X = x\big) \\ &= \arg\min_{z \in y} 1 - \mathbb{P}\big(Y = z|X = x\big) \\ &= \arg\max_{z \in y} \mathbb{P}\big(Y = z|X = x\big) \end{aligned}$$

$f^*(0) = 1 \ car \ \frac{1}{3} < \frac{1}{2}$
$f^*(1) = 1 \ car \ \frac{3}{5} > \frac{1}{2}$
$f^*(2) = 0 \ car \ \frac{3}{4} > \frac{1}{2}$

Le risque Bayésien est donc :

$$\begin{aligned} R^* &= \mathbb{E}[l(Y, f^*(X))] \\ &= \mathbb{E}_X[\mathbb{E}_Y(l(Y \neq f^*(X)|X)] \\ &= \mathbb{E}_X[\mathbb{P}(Y \neq f^*(X)|X)] \end{aligned}$$

Mais on a:

$$\mathbb{P}(Y \neq f^*(X)|X = x) = \mathbb{P}(Y \neq f^*(x))$$

on note $n(x) = \mathbb{P}(Y = 1|X = x)$ Donc,

- if $n(x) > \frac{1}{2}$, donc $f^*(x) = 1$, et $\mathbb{P}(Y \neq f^*(x)) = \mathbb{P}(Y = 0) = 1 - n(x)$
- if $n(x) < \frac{1}{2}$, donc $f^*(x) = 0$, et $\mathbb{P}(Y \neq f^*(x)) = \mathbb{P}(Y = 1) = n(x)$

Donc, $\mathbb{P}(Y \neq f^*(x)) = \min(n(x), 1 - n(x))$
En conclusion :

$$R^* = \mathbb{E}_X[\min(n(x), 1 - n(x))]$$

Dans ce paramètrage :

$$\begin{aligned} R^* &= \frac{1}{2} * (1 - \frac{3}{4}) * (1 - \frac{2}{3}) + \frac{1}{2} * (1 - \frac{3}{5}) + \frac{1}{2} * \frac{3}{4} * \frac{1}{4} \\ &= \frac{1}{2} * (\frac{1}{4} * \frac{1}{3} + \frac{2}{5} + \frac{3}{4} * \frac{1}{4}) \\ &= \frac{1}{2} * (\frac{20}{240} + \frac{96}{240} + \frac{45}{240}) \\ &= \frac{1}{2} * \frac{161}{240} = \frac{161}{480} \approx 0,33 \end{aligned}$$

1.2 Question 2

Soit :

$$\tilde{f} = \begin{cases} X \to Y \\ x \mapsto 1 \end{cases}$$

$$\begin{aligned} R(\tilde{f}) &= 1 - \frac{1}{2} * \frac{1}{4} * \frac{2}{3} - \frac{1}{2} * \frac{3}{5} - \frac{1}{2} * \frac{3}{4} * \frac{1}{4} \\ &= 1 - \frac{1}{2} * (\frac{1}{12} + \frac{3}{5} + \frac{3}{16}) \\ &= 1 - \frac{1}{2} * (\frac{40}{240} + \frac{144}{240} + \frac{45}{240}) \\ &= 1 - \frac{1}{2} * \frac{229}{240} = 1 - \frac{229}{480} \approx 0,52 \end{aligned}$$

$0,52 > R^*$. Ce chiffre est cohérent car le rique de Bayes est inférieur au risque réel.

2 Bayes risk with absolute loss

2.1 Question 1

Trouvons un paramétrage tel que le risque de Bayes soit différent entre la square loss et l'absolute.

Soit $X \in \mathbb{R}$, $X \rightsquigarrow N(0,1)$ et $Y \in \mathbb{R}$ tel que $X = 0,\ Y \rightsquigarrow U(-100, -20, 0, 1, 10, 30)$

Soit l1 représente la square loss et l2 l'absolute loss.
On calcule le risques de Bayes en fonction de l1 :

- Pour $z = -10$ on a $(-100+100)^2 + (-20+100)^2 + (0+100)^2 + (1+100)^2 + (10+100)^2 + (30+100)^2 = 55601$
- Pour $z = -20$ on a 10241
- Pour $z = 0$ on a 11401
- Pour $z = 1$ on a 11249
- Pour $z = 10$ on a 10421
- Pour $z = 30$ on a 12061

Donc $\underset{z \in Y}{\arg\min}\, \mathbb{E}((Y-z)^2 | X = 0) = -20$

On calcule le risques de Bayes en fonction de l2 :

- Pour $z = -10$ on a $|-100+100|+|-20+100|+|0+100|+|1+10|+|10+100|+|30+100| = 521$
- Pour $z = -20$ on a 201
- Pour $z = 0$ on a 161
- Pour $z = 1$ on a 163
- Pour $z = 10$ on a 181
- Pour $z = 30$ on a 241

Donc $\underset{z \in Y}{\arg\min}\, \mathbb{E}(|Y-z|\ |X = 0) = 0$

Ainsi $f_1^*(0) \neq f_2^*(0)$

Donc deux loss différentes peuvent avoir des risques de Bayes différents pour un même paramétrage.

2.2 Question 2

$$Soit\ :\ g(z) = \int_{y\in\mathbb{R}} |y-z|\mathbb{P}_{Y|X=x}(y)dy$$
$$= \int_z^{+\infty} (y-z)\mathbb{P}_{Y|X=x}(y)dy + \int_{-\infty}^{z} (z-y)\mathbb{P}_{Y|X=x}(y)dy$$

$$Donc\ g'(z) = -\int_z^{+\infty} \mathbb{P}(Y|X=x)(y)dy + \int_{-\infty}^{z} \mathbb{P}(Y|X=x)(y)dy$$
$$= -\mathbb{P}(Y \geq z|X=x) + \mathbb{P}(Y \leq z|X=x)$$

$g(z)$ est un minimum si $g'(z) = 0$

Or $g'(z) = 0 \Rightarrow \mathbb{P}(Y \leq z|X=x) = \mathbb{P}(Y \geq z|X=x)$
Ainsi $g(z)$ est minimal si z est la médiane
Donc $f^*(x) = z$

3 Expected value of empirical risk

3.1 Step 1

On démontre que :

$$\mathbb{E}(R_n(\hat{\theta})) = \mathbb{E}_\epsilon(\frac{1}{n}\|I_n - X(X^TX)^{-1}X^T\|^2)$$

Or
$R_n(\hat{\theta}) = \frac{1}{n}\|y - X\hat{\theta}\|^2$
$\hat{\theta} = (X^TX)^{-1}X^Ty$
$y = X\hat{\theta}^* + \epsilon$

$$Donc\ R_n(\hat{\theta}) = \frac{1}{n}\|y - X\hat{\theta}\|^2$$
$$= \frac{1}{n}\|X\hat{\theta}^* + \epsilon - X(X^TX)^{-1}X^T(X\hat{\theta}^* + \epsilon)\|^2$$
$$= \frac{1}{n}\|X\hat{\theta}^* + \epsilon - X(X^TX)^{-1}X^TX\theta^* + X(X^TX)^{-1}X^T\epsilon\|^2$$
$$= \frac{1}{n}\|\epsilon(I_n - X(X^TX)^{-1}X^T)\|^2$$

Donc $\mathbb{E}(R_n(\hat{\theta})) = \mathbb{E}_\epsilon(\frac{1}{n}\|I_n - X(X^TX)^{-1}X^T\|^2)$

3.2 Step 2

Soit $A = a_{ij}$, $B = {}^tA$ et C = A × B.
Si les coefficients sont $B = (b_{ij})$ alors par définition de la transposée on a $b_{ij} = a_{ji}$.
Par définition du produit de matrices le coefficients c_{ij} de C s'obtient par la formule :

$$\begin{aligned} c_{ij} &= \sum_{k=1}^{n} a_{ik} b_{kj} \\ &= \sum_{k=1}^{n} a_{ik} a_{jk} \ (avec\ B = {}^tA\ ) \end{aligned}$$

Pour un coefficient de la diagonale on a i = j donc :

$$c_{ii} = \sum_{k=1}^{n} a_{ik}^2$$

Donc :

$$\begin{aligned} tr(A^tA) &= tr(C) \\ &= \sum_{i=1}^{n} c_{ii} \\ &= \sum_{i=1}^{n} \sum_{k=1}^{n} a_{ik}^2 \\ &= \sum_{(i,k)\in[1,n]^2} a_{ik}^2 \end{aligned}$$

On remplace k par j est :

$$\sum_{(i,j)\in[1,n]^2} A_{ij}^2 = tr(A^tA)$$

4 Regression and Classification

Afin d'obtenir de meilleurs résultats en précision moyenne et en score R2, nous avons utilisé la méthode de Cross Validation. Cette méthode permet d'optimiser les paramètres utilisés sur un modèle afin qu'ils soient les plus adéquats avec les données utilisées. Nous avons utilisé la Cross Validation avec deux cas: un cas où la librairie python intègre la méthode au modèle (Exemple: RidgeCV) et un autre cas où la méthode est utilisée par GridSearchCV. Cette fonction permet sur un modèle d'utiliser les paramètres optimaux parmis plusieurs donnés avec la méthode de Cross Validation.

Cette partie se portera sur la résolution d'un problème de régressions avec différents algorithmes de machine learning. L'évaluation se porte sur un jeu de données de 1000 lignes et 20 colonnes.

La première méthode de régression que nous avons utilisée est RandomForestRegressor, un modèle utilisant un algorithme se basant sur l'assemblage d'arbres de décision. Le résultat obtenu est bon mais insuffisant (0.82).

Nous avons ensuite essayé de comparer les résultats sur un même modèle avec et sans GridSearchCV. Le modèle choisi est Lasso, utilisé sur python depuis la librairie sklearn. Avec un choix donné à la fonction pour le paramètre alpha (choix proche de la valeur par défaut 1) le score R2 se trouve être très légèrement supérieur avec l'utilisation de la fonction. Dans les deux cas le résultat est correct et suffisant (0.89)

Enfin pour la partie régression, nous avons décidé d'utiliser des modèles intégrant directement la méthode de Cross Validation depuis la librairie python sklearn (RidgeCV LassoCV). Les deux modèles ont été appelés avec le paramètre CV=5. Les résultats sont très corrects et très proches de ceux obtenus précédemment (0.89).
Pour la classification, nous avons gardé les méthodes Ridge et Random Forest pour voir si l'écart de performance était le même sur les deux problèmes en rajoutant la Logistic Regression pour avoir une comparaison supplémentaire. Pour le benchmark des trois méthodes, nous avons utilisé la bibliothèque scikit-learn, nous permettant de faire des tests sans avoir à tout implémenter à la main et de tuner facilement les hyperparamètres. Le dataset utilisé est un numpy array de float32 de taille 1000x20 et un numpy array représentant les labels [-1; 1] de taille 1000x1.

Ces deux datasets ont été split à l'aide de la fonction train_and_test_split avec le paramètre test_size à 0.2 afin d'obtenir un découpage aléatoire des données d'entraînement et de test pour nos benchmarks.

Le Logistic Regression Classifier avec Cross-Validation a été testé avec le paramètre cv à 5 pour que l'aglorithme Stratified K-Folds génère 5 folds. Le résultat de l'entraînement donne 0.896 pour les données de d'entraînement et 0.91 pour les données de test.
Le Ridge Classifier avec Cross-Validation a été testé sans paramètre et a des résultats presque similaires. Il donne 0.898 sur les données d'entraînement et 0.9 sur les données de test ce qui montre que la méthode Ridge s'adapte moins bien à de nouvelles données mais donne de meilleurs résultats à l'entraînement.

La méthode Random Forest donne quand à elle des résultats excellents pour la classification d'un dataset, mais est très peu tolérant à la reception de nouvelles donnée. En effet, sur les données d'apprentissage, le score est de 1, tandis que les données de tests donnent 0.85.