Chapitre V
Régularisation
Cycle pluridisciplinaire d'études supérieures
Université Paris sciences et lettres

Le compromis entre biais et complexité a été présenté dans le chapitre précédent. L'ajustement de ce compromis s'effectue souvent à l'aide d'un ou de plusieurs hyperparamètres. On présente dans ce chapitre un exemple fondamental, la minimisation du risque empirique *régularisé*, où un *coefficient de régularisation* est un hyperparamètre avec lequel la complexité décroît.

I. cadre

Soit $d \geqslant 1$ un entier. On considère l'ensemble d'entrées $\mathcal{X} = \mathbb{R}^d$ (qui correspond à d variables explicatives quantitatives), et un ensemble de sorties $\mathcal{Y}$ quelconque.

On va considérer une classe de *prédicteurs linéaires*. Pour $(w, b) \in \mathbb{R}^d \times \mathbb{R}$, on rappelle qu'on note $g_{w,b} : x \mapsto \langle w, x \rangle + b$. On se donne une application $\phi : \mathbb{R} \to \mathcal{Y}$ et pour tout $(w, b) \in \mathbb{R}^d \times \mathbb{R}$, on note $f_{w,b} = \phi \circ g_{w,b}$. Enfin, on note $\mathcal{F}$ la classe de tous les prédicteurs $f_{w,b}$:

$$\mathcal{F} = \{f_{w,b}\}_{\substack{w \in \mathbb{R}^d \\ b \in \mathbb{R}}}.$$

Pour $(w, b) \in \mathbb{R}^d \times \mathbb{R}$, les coefficients $w_1, \dots, w_d$ de w ainsi que b sont appelés *paramètres* du prédicteur $f_{w,b}$.

Remarque. — *Paramètres et importance accordée aux variables explicatives.* — Soit $(w, b) \in \mathbb{R}^d \times \mathbb{R}$. Pour $1 \leqslant j \leqslant d$, le paramètre w_j peut être interprété comme l'importance accordée par le prédicteur $f_{w,b}$ à la j-ème variable explicative. En effet, si w_j est grand en valeur absolue, alors il découle de la définition de $f_{w,b}$ que la prédiction $f_{w,b}(x)$ dépendra fortement de la variable x_j. À l'inverse, si $w_j = 0$, la prédiction $f_{w,b}(x)$ ne dépendra pas de la variable x_j.

Remarque. — *Grandeur des paramètres et complexité.* — Un algorithme ayant tendance à donner des prédicteurs $f_{\hat{w},\hat{b}}$ avec des paramètres $\hat{w}_1, \dots, \hat{w}_d, \hat{b}$ grands en valeur absolue sera considéré comme ayant une grande *complexité*.

Soit $n \geqslant 1$ un entier, $S = (x_i, y_i)_{i\in[n]} \in \mathcal{S}(\mathcal{X}, \mathcal{Y})$ un échantillon, et $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ une fonction. On rappelle que l'algorithme de *minimisation du risque empirique* (ERM) utilisant la fonction de perte ℓ donne, avec S pour échantillon d'apprentissage, le prédicteur $f_{\hat{w},\hat{b}}$ où :

$$(\hat{w}, \hat{b}) = \operatorname*{arg\,min}_{\substack{w\in\mathbb{R}^d \\ b\in\mathbb{R}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_{w,b}(x_i)) \right\}.$$

Les cas particuliers importants déjà rencontrés sont la régression linéaire aux moindres carrés et la régression logistique. La minimisation du risque empirique présentée ci-après est une modification de la minimisation du risque empirique.

2. Minimisation du risque empirique régularisé

On introduit les objets suivants. Soit $\rho : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ une fonction appelée *régularisatrice* et $\lambda \geqslant 0$ un réel appelé *hyperparamètre de régularisation*.

Définition. — *Minimisation du risque empirique régularisé.* — L'algorithme de la minimisation du risque empirique régularisé avec ℓ pour fonction de perte, ρ pour fonction régularisatrice, λ pour hyperparamètre de régularisation et S pour échantillon d'apprentissage donne le prédicteur $f_{\hat{w},\hat{b}}$ où :

$$(\hat{w}, \hat{b}) = \operatorname*{arg\,min}_{\substack{w\in\mathbb{R}^d \\ b\in\mathbb{R}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_{w,b}(x_i)) + \lambda\rho(w, b) \right\}.$$

Remarque. — L'expression qui est minimisée ci-dessus comporte à présent deux termes : le risque empirique d'une part, et le terme de régularisation d'autre part. Le prédicteur obtenu $f_{\hat{w},\hat{b}}$ résulte d'un compromis entre les deux termes. Si $\lambda = 0$, il n'y a pas de régularisation et on retrouve la minimisation du risque empirique. Si λ est grand, le biais est important.

Remarque. — *Choix de la fonction régularisatice* ρ. — Typiquement, on choisit ρ une fonction croissante en les valeurs absolues des paramètres $w_1, \dots, w_d, b$. Des exemples sont présentés dans les paragraphes suivants. On a alors les conséquences suivantes. D'une part, la régularisation a tendance à diminuer les valeurs absolues des paramètres obtenus $\hat{w}_1, \dots, \hat{w}_d, \hat{b}$, ce qu'on interprète comme une diminution de la complexité. D'autre part, augmenter la valeur de λ réduit le sur-apprentissage [1]. En effet, plus les paramètres sont grands en valeur absolue, plus les prédictions sur prédicteur obtenu dépendront fortement de l'entrée x. Or un tel prédicteur pourrait donner de bonnes prédictions sur les exemples d'apprentissage, mais donner des prédictions très différentes sur des exemples pourtant proches, d'où un risque de sur-apprentissge.

Remarque. — *Choix de l'hyperparamètre de régularisation* λ. — Pour choisir une valeur de λ, on considère en pratique plusieurs valeurs pour ensuite choisir celle qui semble la meilleure par validation (éventuellement *croisée*).

3. Ridge : régularisation ℓ_2

Un exemple important de fonction régularisatrice est celle donnée par le carré de la norme ℓ_2 (ici ℓ_2 n'a aucun rapport à la fonction de perte, mais désigne la norme euclidienne), et parfois appelée régularisation de *Tikhonov* :

$$\forall (w, b) \in \mathbb{R}^d \times \mathbb{R}, \quad \rho_{\text{ridge}}(w, b) = \frac{1}{2}\left(\|w\|_2^2 + b^2\right) = \frac{1}{2}\left(\sum_{j=1}^{d} w_j^2 + b^2\right).$$

L'algorithme correspondant est appelé *Ridge* et s'écrit donc :

$$(\hat{w}, \hat{b}) = \underset{(w,b)\in\mathbb{R}^d\times\mathbb{R}}{\arg\min} \left\{ \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_{w,b}(x_i)) + \frac{\lambda}{2}\left(\|w\|_2^2 + b^2\right) \right\}.$$

1. Si toutefois $n \gg d$, le grand nombre d'exemples d'apprentissage empêche alors déjà le sur-apprentissage.

Remarque. — La fonction régularisatrice ρ_{ridge} est croissante en les valeurs absolues des paramètres $w_1, \dots, w_b, b$. Par conséquent, augmenter la valeur de l'hyperparamètre λ réduit la complexité de l'algorithme et donc le sur-apprentissage.

Remarque. — Plus la valeur de λ est grande, et plus les paramètres obtenus $\hat{w}_1, \dots, w_d, \hat{b}$ ont tendance à être faibles en valeur absolue. Pour s'en convraincre, on peut imaginer qu'on minimise seulement le terme de régularisation ; on voit que les paramètres obtenus sont alors tous nuls. Donc si on minimise le risque empirique *plus* le terme de régularisation, c'est un compromis entre la minimisation du risque empirique (sans régularisation) et le fait d'avoir des paramètres tous nuls. Les paramètres obtenus ont donc tendance à être plus faibles valeur absolue lorsqu'on ajoute le terme de régularisation.

4. LASSO : RÉGULARISATION ℓ_1

L'autre exemple fondamental de fonction régularisatrice est celle donnée par la norme ℓ_1, définie par :

$$\forall (w, b) \in \mathbb{R}^d \times \mathbb{R}, \quad \rho_{\text{lasso}}(w, b) = \|w\|_1 + |b| = \sum_{j=1}^{d} |w_j| + |b| \,.$$

L'algorithme correspondant est appelé LASSO (*Least Absolute Shrinkage and Selection Operator*) et donne donc le prédicteur $f_{\hat{w},\hat{b}}$ où :

$$(\hat{w}, \hat{b}) = \underset{(w,b)\in\mathbb{R}^d\times\mathbb{R}}{\arg\min} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_{w,b}(x_i)) + \lambda \left(\|w\|_1 + |b| \right) \right\} .$$

Remarque. — La fonction régularisatrice ρ_{lasso} est également croissante en les valeurs absolues des paramètres $w_1, \dots, w_d, b$. Par conséquent, augmenter la valeur de l'hyperparamètre λ réduit la complexité de l'algorithme et donc le sur-apprentissage.

Remarque. — *Tendance à donner peu de paramètres non nuls.* — Augmenter la valeur de λ permet non seulement d'obtenir des paramètres $\hat{w}_1, \dots, \hat{w}_d, \hat{b}$ plus petits en valeur absolue, mais permet également de rendre *nuls* un grand nombre de ces paramètres. Cela est dû a la spécificité de la régularisatrice ℓ_1, et n'est pas le cas pour régularisatrice ℓ_2. En effet, minimiser une fonction valeur absolue

$x \mapsto |x|$ ou une fonction carré $x \mapsto x^2$ donne à chaque fois 0 comme minimiseur, mais lorsqu'on s'écarte de 0, la valeur de la fonction augmente plus nettement dans le premier cas quand dans le second : la fonction $x \mapsto |x|$ *force* plus nettement le minimiseur à être 0 que la fonction $x \mapsto x^2$. Ainsi, si beaucoup de paramètres sont nuls, les prédictions du prédicteur obtenu ne dépendront que d'un petit nombre de variables explicatives (celles correspondant aux paramètres $\widehat{w}_j$ non nuls). Cette tendance est une des raisons principales pour lesquelles on a recours à l'algorithme LASSO : en effet, lorsqu'on soupçonne, ou qu'on sait, qu'un grand nombre de variables explicatives n'a pas de corrélation significative avec la variable à prédire, on peut souhaiter obtenir des prédicteurs qui ignorent un grand nombre de variables explicatives.