

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

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

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  $\rho : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$  une fonction appelée *régularisatrice* et  $\lambda \geq 0$  un réel appelé *hyperparamètre de régularisation*.

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

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

**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  $\lambda$  est grand, le biais est important.

**REMARQUE.** — *Choix de la fonction régularisatrice  $\rho$ .* — Typiquement, on choisit  $\rho$  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  $\lambda$  réduit le sur-apprentissage<sup>1</sup>.

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

### 3. LASSO : RÉGULARISATION $\ell_1$

Un exemple important de fonction régularisatrice est celle de la norme  $\ell_1$  (ici  $\ell_1$  n'a aucun rapport à la fonction de perte, mais désigne la norme  $\|\cdot\|_1$ ), 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 :

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

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

---

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

**REMARQUE.** — *Tendance à donner peu de paramètres non nuls.* — Augmenter la valeur de  $\lambda$  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 à la spécificité de la régularisatrice  $\ell_1$ , et ne sera pas le cas pour la régularisatrice  $\ell_2$  présentée au paragraphe suivant). Alors, les prédictions du prédicteur obtenu ne dépendront que d'un petit nombre de variables explicatives (celles correspondant aux paramètres  $\hat{w}_j$  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.

#### 4. RIDGE : RÉGULARISATION $\ell_2$

L'autre exemple fondamental de fonction régularisatrice est celle donnée par le carré de la norme  $\ell_2$ , et parfois appelée régularisation de *Tikhonov* :

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

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

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

**REMARQUE.** — La fonction régularisatrice  $\rho_{\text{ridge}}$  est également croissante en les valeurs absolues des paramètres  $w_1, \dots, w_b, b$ . Par conséquent, augmenter la valeur de l'hyperparamètre  $\lambda$  réduit la complexité de l'algorithme et donc le surapprentissage.

**REMARQUE.** — Plus la valeur de  $\lambda$  est grande, et plus les paramètres obtenus  $\hat{w}_1, \dots, w_d, \hat{b}$  sont petits en valeurs absolue, mais contrairement à l'algorithme LASSO, ils ne sont en pratique jamais nuls.

