Thème 1

Introduction à la prédiction supervisée

Vincent Charvillat
Septembre 2014

Ce premier thème introduit la notion de prédiction utilisée en apprentissage artificiel ou analyse de données. Quelques classifieurs élémentaires sont également présentés.

1.1 Terminologie illustrée

En analyse de données ou en apprentissage articificiel, on s'intéresse à des *mécanismes de prédiction* dans des contextes où les données manipulées sont *aléatoires*. Ces dernières étant vues comme des réalisations de variables aléatoires, on parle donc aussi d'*apprentissage statistique*.

L'exemple introductif classique est celui de la reconnaissance automatique de l'écriture manuscrite. La figure 1.1 montre des réalisations manuscrites aléatoires des chiffres. Un algorithme d'apprentissage artificiel permet de mettre au point un mécanisme de prédiction du nombre écrit à partir d'une donnée d'entrée (par exemple une imagette numérique de 64×64 pixels). En d'autres mots, on veut prédire la classe d'appartenance d'une imagette quelconque ($\in \mathbb{R}^{64\times 64}$) parmi les dix classes possibles associées aux chiffres $\{0, 1, \ldots, 9\}$ possibles. Un tel problème de prédiction d'une classe d'appartenance est aussi appelé un problème de *classification*.

Plus formellement, on considèrera pour la classification, un espace d'entrée $\mathcal{X}$ ($\mathbb{R}^{64\times 64}$ dans l'exemple ci-dessus) et un espace discret de sortie à k classes $\mathcal{Y} = \{0, \ldots k-1\}$ ($k = 10$ dans l'exemple). Un *prédicteur* (ou *classifieur*) est une fonction $h : \mathcal{X} \to \mathcal{Y}$ qui prédit pour une donnée d'entrée $x_i \in \mathcal{X}$ (une imagette dans l'exemple), sa classe : $\widehat{y_i} = h(x_i) \in \mathcal{Y}$.

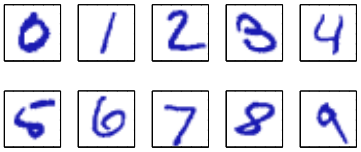


Figure 1.1: Réalisations manuscrites de chiffres.

Pour savoir si cette prédiction est correcte, on peut utiliser un ensemble de données supervisées. Une paire $z_i = (x_i, y_i)$ forme une donnée supervisée, si un expert a étiqueté la donnée d'entrée x_i avec sa classe correcte y_i. Plus formellement, un ensemble de n données supervisées est, par exemple, noté $D = \{z_i\}_{i=1...n}$ avec $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$. Pour savoir si une prédiction $\widehat{y_i} = h(x_i) \in \mathcal{Y}$ d'un classifieur h est bonne, on la compare alors à l'étiquette y_i correcte (donnée par l'expert). Une fonction non-négative $e : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ dite de *perte* exprime l'erreur de prédiction (ou, entre d'autres mots, l'ampleur de l'écart entre y_i et la prédiction $\widehat{y_i}$). Naturellement, on prend souvent une perte nulle (resp. strictement positive) si la prédiction est correcte (resp. fausse) : $e(\widehat{y_i}, y_i) = 0$ si $\widehat{y_i} = y_i$ (resp. $e(\widehat{y_i}, y_i) > 0$ si $\widehat{y_i} \neq y_i$).

Un bon classifieur devra conduire à des pertes minimales sur l'ensemble des données (ou exemples) qu'il aura à classer (c'est-à-dire potentiellement tout $\mathcal{X} \times \mathcal{Y}$). Mais, toutes les paires $z_i = (x_i, y_i)$ ne sont pas équiprobables. Chaque paire est la réalisation d'une v.a. $Z = (X, Y)$. Les distributions de probabilités associées aux variables aléatoires X, Y, Z et leurs dépendances [1] sont des ingrédients clés en apprentissage statistique où l'on cherchera à minimiser l'espérance des pertes.

1.2 Une multitude de prédicteurs possibles

Ce principe général de réduction de l'espérance des pertes induites par de mauvaises prédictions sera formalisé dans la suite. Il faut toutefois noter qu'il permet de comparer une multitude de prédicteurs de natures parfois très différentes.

Les prédicteurs peuvent être définis mathématiquement (un prédicteur peut être une fonction dont on connaît la forme analytique) ou de manière procédurale (un prédicteur peut être associé au parcours d'un arbre de décision ou de régression). Les prédicteurs peuvent être déterministes (une entrée x conduit invariablement à la même prédiction) ou probabiliste (sachant l'entrée, une prédiction est émise avec une certaine probabilité). Pour un même type de prédicteur (qu'il s'agisse, par exemple, d'une fonction polynomiale ou d'un arbre), *les prédicteurs peuvent être de complexité variable* : le degré du polynôme peut croître, la profondeur de l'arbre peut varier. Si les prédicteurs sont paramétriques, la complexité croît généralement avec le nombre de paramètres (ou degré de liberté) considérés.

Dans la suite, nous illustrons cette multiplicité de prédicteurs possibles en introduisant plusieurs classifieurs différents : un classifieur binaire linéaire, un classifieur quadratique, un arbre de décision...

[1] Y doit dépendre de X si on espère faire une prédiction à partir d'une réalisation de X

1.2.1 Classifieurs binaires et quadratiques

Un classifieur binaire linéaire $h^{\theta}_{\mathbf{w}}$ est défini, de manière paramétrique, par :

- un vecteur $\mathbf{w} = (w_1, .., w_p)^T$ de p poids,
- un seuil de décision θ.

Les espaces d'entrée et de sortie sont $\mathcal{X} \subset \mathbb{R}^p$, $\mathcal{Y} = \{+1, -1\}$
Une prédiction s'exprime alors selon :

$$\widehat{y} = h^{\theta}_{\mathbf{w}}(x) = \text{sgn}(\mathbf{w}^T x - \theta) \tag{1.1}$$

où $\text{sgn}(s) = +1$ lorsque $s > 0$ et -1 lorsque $s \leq 0$

On dit que ce classifieur binaire (cas à deux classes) est linéaire car il combine linéairement les composantes du vecteur d'entrée x avec les poids rangés dans le vecteur de paramètres $\mathbf{w}$. Un algorithme d'apprentissage consiste à ajuster les paramètres $\mathbf{w}$ et θ pour qu'ils soient cohérents avec un ensemble de données supervisées ou *échantillon d'apprentissage* $D = \{(x_i, y_i)\}_{i=1...n} \in (\mathcal{X} \times \mathcal{Y})^n$.

Par extension avec le classifieur linéaire précédent, on peut introduire un classifieur quadratique dont la prédiction est donnée par :

$$\widehat{y} = h_{\omega}(x) = \text{sgn}\left(\omega_0 + \sum_i \omega_i x_i + \sum_{i,j} \omega_{i,j} x_i x_j\right) \tag{1.2}$$

Il est clair que les classifieurs quadratiques sont plus généraux que les classifieurs linéaires. Ils conduisent à des régions et frontières de décisions plus complexes que celles issues d'une séparation linéaire. Si on nomme $\mathcal{H}_1$ (resp. $\mathcal{H}_2$, resp. $\mathcal{H}_k$), la classe des prédicteurs linéaires (resp. quadratiques, resp. d'ordre k), on a une hiérarchie de classes telle que :

$$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_k \tag{1.3}$$

La croissance de la *complexité des classifieurs* va avec celle du nombre de paramètres à estimer. On peut donc d'emblée distinguer deux problèmes complémentaires :

- Etant donnée une classe de fonction $\mathcal{H}$, quel est le meilleur prédicteur $h \in \mathcal{H}$ possible pour un problème de prédiction donné ? Il s'agit, par exemple, d'estimer les meilleurs paramètres ($\mathbf{w}$ et θ) d'un classifieur linéaire connaissant un échantillon d'apprentissage D. C'est un problème d'estimation (au sens des statistiques) ou d'optimisation (au sens de l'analyse).
- Etant donnée une hiérarchie de modèles de prédiction possibles ($\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_k$), quelle est la meilleure classe possible ? Il s'agit par exemple de choisir entre un classifieur linéaire ou quadratique. En apprentissage, effectuer ce choix revient à résoudre un problème de *sélection de modèle* sur lequel nous reviendrons plus en détail.

1.2.2 Arbre de décision

Les séparateurs précédents sont définis analytiquement. On peut aussi utiliser des approches algorithmiques ou procédurales pour définir des prédicteurs dans le domaine de la classification ou de la régression. L'exemple des arbres de décision est révélateur de cette possibilité. La figure 1.2 présente la structure arborescente d'un classifieur binaire qui opère sur un espace d'entrée $\mathcal{X} = [0.0, 1.0]^2 \subset \mathbb{R}^2$ et peut prédire les deux valeurs qualitatives de $\mathcal{Y} = \{+1, -1\}$. Cet arbre de décision est un arbre de discrimination binaire qui consiste à classer une donnée x de $\mathcal{X}$ à l'issue du parcours d'une séquence de *noeuds*.

Un noeud est défini par le choix conjoint d'une variable (parmi les composantes de la variable d'entrée) et d'un prédicat qui induit une *division* des données en deux classes. A titre d'exemple, la racine de l'arbre de la figure 1.2 est un noeud qui sépare les données d'entrée $x \subset \mathbb{R}^2$ selon que la variable associée à la première composante (notée $X1$) de x est supérieure ou non à un seuil (0.5). Si les variables ne sont pas quantitatives, l'utilisation d'un seuil doit être remplacée par une autre fonction logique. Il est aisé à partir de cet arbre de décision de déterminer les régions et frontières de décisions du classifieur binaire ainsi obtenu.

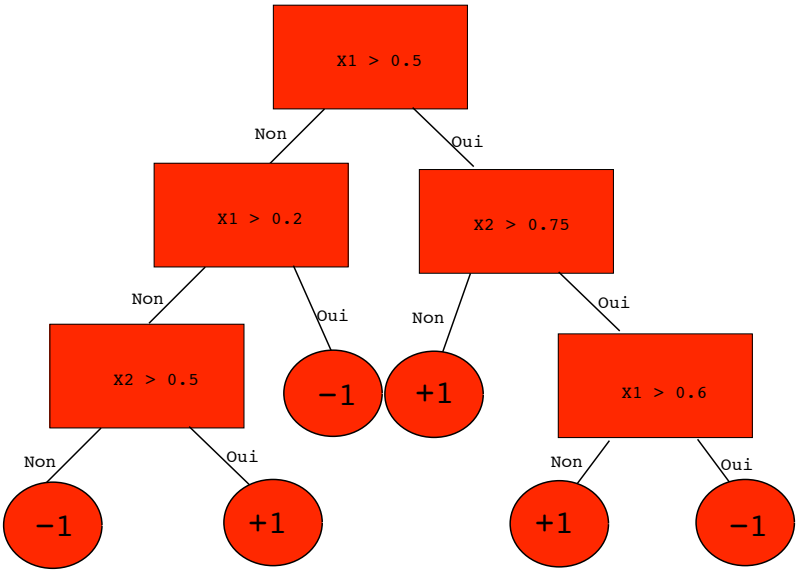


Figure 1.2: Un arbre de décision

Il reste trois points délicats à aborder pour construire (apprendre) automatiquement un tel classifieur :

i) Parmi toutes les *divisions* admissibles à un instant donné lors de la création d'un tel arbre, il faut pouvoir identifier la meilleure division possible. Ce critère de sélection est à bien choisir.

ii) Il faut également choisir une règle permettant de décider si un noeud est terminal ou pas.

iii) Enfin, les noeuds terminaux (c'est-à-dire les feuilles de l'arbre), doivent être étiquetés par les valeurs de l'espace de sortie $\mathcal{Y}$ selon un principe à mettre au point.

Le second point ii), qui vise à interrompre le développement en profondeur de l'arbre, revient à gérer la *complexité* du classifieur obtenu au sens exactement équivalent à ce qui a été vu plus haut. Un arbre trop développé, c'est-à-dire trop compliqué, peut être obtenu si on cherche à expliquer «coûte que coûte» un ensemble de données supervisées. L'arbre maximal, le plus développé, conduit à des régions (et frontières) de décision très complexes. Ce classifieur peut se révéler défaillant pour la prédiction de nouvelles données comme nous l'expliquerons formellement plus loin. L'élagage d'un arbre maximal revient alors à simplifier le modèle pour atteindre le meilleur compromis entre fidélité aux données d'apprentissage et bonne capacité de prédiction sur de nouvelles données.

1.2.3 Prédiction supervisée

Parmi différentes formes d'apprentissage artificiel possibles, l'**apprentissage supervisé** est celui que nous avons tacitement utilisé dans les illustrations précédentes. Il s'agit du scénario d'apprentissage pour lequel un expert fournit d'emblée des exemples de ce qu'il faut apprendre. Exemples à partir desquels un prédicteur est bâti. On s'intéresse typiquement à un prédicteur h qui prédit une sortie $\widehat{y}$ à partir d'un stimulus $x \in \mathcal{X}$ selon le schéma suivant :

$$x \in \mathcal{X} \rightarrow \boxed{\text{h ?}} \rightarrow \widehat{y} = h(x) \in \mathcal{Y}$$

Comme nous l'avons déjà dit, un ensemble de n données supervisées, appelé également «échantillon d'apprentissage», est donné par l'expert et usuellement noté $D = \{(x_i, y_i)\}_{i=1...n}$. Il regroupe les dires de l'expert qui a indiqué qu'à la donnée d'entrée x_i devait correspondre la valeur de sortie y_i. Le prédicteur h peut alors être appris automatiquement en minimisant, pour toutes les données disponibles dans D, les écarts entre ses prédictions $\widehat{y_i} = h(x_i)$ et les sorties correctes y_i. Une fonction de *perte* à bien choisir mesure numériquement ces écarts (les erreurs de prédiction). Retenons que la solution au problème de prédiction posé est connue, grâce aux instructions fournies par l'expert, pour un ensemble de données d'entrées $x_i, i = 1, .., n$.

Thème 2

Généralisation

Vincent Charvillat
Septembre 2014

Dans ce second temps, on formalise la notion de généralisation qui est centrale en prédiction supervisée. On souligne aussi la similitude et les différences entre régression et classification.

2.1 Complexité optimale d'un classifieur

A titre d'exemple introductif, on considère un classifieur dont l'espace d'entrée est $\mathcal{X}$ et l'espace discret de sortie à k classes est $\mathcal{Y} = \{0, \dots k-1\}$. On considère un ensemble de données supervisées $D = \{z_i\}_{i=1\dots n}$. On décompose classiquement les données selon $z_i = (x_i, y_i)$ avec $x_i \in \mathcal{X}$ et $y_i \in \mathcal{Y}$. Chaque donnée z_i est la réalisation d'une v.a. $Z = (X, Y)$. On notera $P_Z(z) = P_{X,Y}(x, y)$ (resp. $P_X(x)$, $P_{Y|X}(y)$), les distributions jointes, (resp. d'entrée, conditionnelle) associées.

Nous allons décrire trois choix importants à effectuer en vue de l'apprentissage d'un tel classifieur :

- choix de la fonction perte,
- choix des classes de prédicteurs considérées (un ou plusieurs modèles de complexités distinctes)
- apprentissage, au sein de chaque classe, des meilleurs classifieurs par minimisation du risque empirique.

2.1.1 Fonctions de perte pour la classification

Comme nous l'avons vu précédemment, les instructions données par l'expert ($\forall i$, «classer la donnée d'entrée x_i avec l'étiquette y_i») permettent d'apprendre un classifieur à partir de l'ensemble d'apprentissage D en minimisant les écarts entre ses

prédictions et les vraies valeurs. En classification, la fonction de perte «non informative» (notée e_z) est la plus simple à utiliser pour mesurer cet écart. Elle est définie par :

$$e_z(\widehat{y}, y) = \begin{cases} 0 \ si \ y = \widehat{y} \\ 1 \ si \ y \neq \widehat{y} \end{cases} \tag{2.1}$$

On dit que la perte e_z est non-informative car elle se contente de dire, de manière binaire, si la prédiction $\widehat{y}$ est correcte (resp. fausse) et a donc un coût nul (resp. unitaire). On parle aussi de perte 0-1 pour e_z. D'autres pertes peuvent être imaginées pour quantifier de manière plus fine l'écart entre une prédiction imparfaite et la classe correcte. Dans certaines applications, il est clair que toutes les erreurs de classification ne sont pas de gravité équivalente et doivent donc avoir des coûts différenciés. Une fonction perte générale e doit donc simplement vérifier :

$$e(\widehat{y}, y) = \begin{cases} 0 \ si \ y = \widehat{y} \\ > 0 \ si \ y \neq \widehat{y} \end{cases} \tag{2.2}$$

2.1.2 Hiérarchie de modèles de complexité croissante

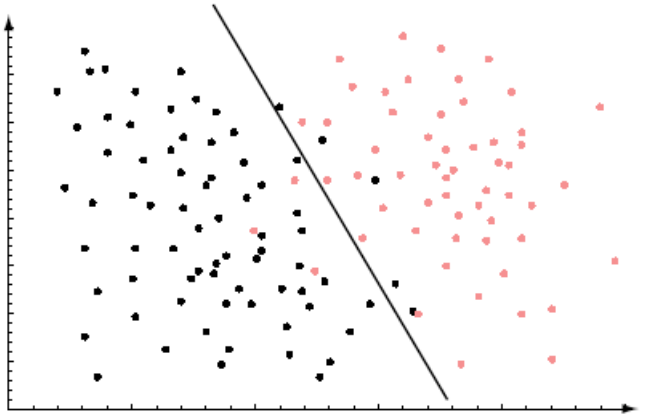


Figure 2.1: Un classifieur «trop simple»

Dans notre exemple, on choisit trois modèles de complexité croissante :

- les classifieurs linéaires (classe $\mathcal{H}_1$),
- les classifieurs quadratiques (classe $\mathcal{H}_2$ avec $\mathcal{H}_1 \subset \mathcal{H}_2$),
- des classifieurs beaucoup plus compliqués (classe $\mathcal{H}_c$).

Les formes possibles des frontières de décision associées à cette hiérarchie de modèles sont illustrées par la figure 2.1 pour un classifieur linéaire, la figure 2.2 pour un classifieur quadratique et la figure 2.3 pour un classifieur de complexité la plus forte. Ces figures présentent un exemple de classification binaire pour lequel les données des deux classes présentes dans D sont colorées en noir et rouge.

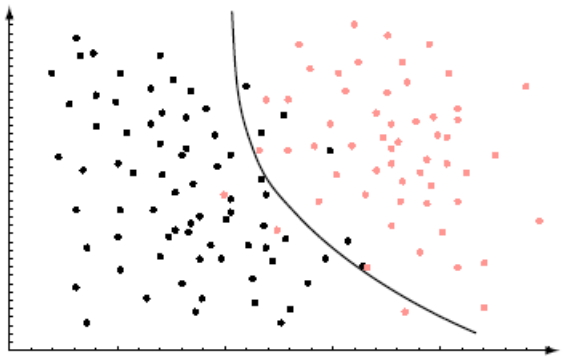


Figure 2.2: Un classifieur de complexité «moyenne»

2.1.3 Minimisation du risque empirique

Les classifieurs précédents sont obtenus à l'issue de la minimisation du risque empirique. On appelle, pour un classifieur candidat h (pris dans une classe de fonctions $\mathcal{H}$), un ensemble de données supervisées $D = \{(x_i, y_i)\}_{i=1\ldots n}$ et une perte e, «risque empirique» la quantité :

$$R_D(h) = \frac{1}{n} \sum_{i=1}^{n} e(h(x_i), y_i) \tag{2.3}$$

Le classifieur dont la frontière de décision est montrée sur la figure 2.1 est donc le classifieur h_1^* solution de :

$$h_1^* = \underset{h \in \mathcal{H}_1}{\operatorname{argmin}}\ R_D(h) \tag{2.4}$$

Apprendre le meilleur classifieur d'une classe donnée à partir de D revient donc à résoudre un problème d'optimisation. On appelle aussi «erreur d'apprentissage» le risque empirique minimal obtenu pour la solution h_1^* (après apprentissage) :

$$\texttt{err}(h_1^*) = R_D(h_1^*) \tag{2.5}$$

Cette erreur évalue le cumul des pertes engendrées par les mauvaises prédictions du classifieur sur l'échantillon d'apprentissage D. De même on notera h_2^* (resp. h_c^*), les meilleurs classifieurs obtenus par minimisation du risque empirique sur les classes $\mathcal{H}_2$ (resp. $\mathcal{H}_c$). On remarquera, qu'en supposant que la perte e_z est celle utilisée pour apprendre le classifieur linéaire de la figure 2.1, l'erreur d'apprentissage est alors facile à déterminer grâce à la figure.

Lorsque les modèles choisis pour les classifieurs deviennent de plus en plus complexes, c'est-à-dire qu'ils disposent de plus en plus de degrés de liberté, les régions de décisions peuvent plus facilement s'ajuster aux données d'apprentissage présentes dans D. On observe ainsi sur la figure 2.3 que l'erreur d'apprentissage $\texttt{err}(h_c^*)$ est nulle : $\texttt{err}(h_c^*) = R_D(h_c^*) = 0$. Grâce à la flexibilité du classifieur, toutes les prédictions sont correctes et la frontière de décision sépare les données sans commettre d'erreurs parmi les données d'apprentissage de D.

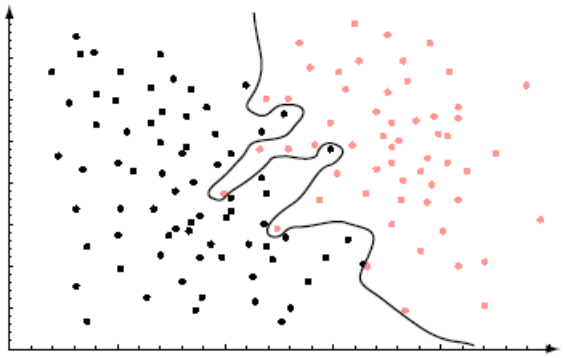


Figure 2.3: Un classifieur «trop complexe»

Un classifieur qui réussit à ramener une erreur d'apprentissage à zéro est-il pour autant le meilleur ? La réponse est négative. La seule «minimisation du risque empirique» ne suffit pas pour obtenir un «apprentissage optimal».

2.2 Risque espéré et généralisation

Le meilleur classifieur est celui dont la performance ne se limite pas à correctement classifier les données appartenant à l'ensemble d'apprentissage : on souhaite aussi qu'il prédise correctement la classe de *nouvelles données* pour lesquelles l'expert ne s'est pas prononcé. On dira que le meilleur classifieur est celui qui possède la plus forte capacité de *généralisation*. En guise d'illustration, le classifieur de la figure 2.3 possède des frontières qui satisfont parfaitement l'ensemble d'apprentissage ($\texttt{err}(h_c^*) = R_D(h_c^*) = 0$) mais dont on pressent qu'elles sont «trop complexes» pour correctement prédire de nouveaux exemples situés à la frontière des deux classes.

Formellement, on appelle "Expected Prediction Error (EPE)" ou risque espéré (R) d'un classifieur $h : \mathcal{X} \to \mathcal{Y}$ pour une perte non négative e, la quantité :

$$R(h) = EPE(h) = E_{(X,Y)}[e(h(X), Y)] \tag{2.6}$$

Cette quantité mesure l'espérance des pertes dues aux prédictions de h sur l'ensemble des données susceptibles d'être générées par l'application (et non pas uniquement sur les données supervisées). Chaque nouvelle donnée (ou exemple) se réalise selon une mesure d'entrée x (réalisation de la variable aléatoire X) et correspond à une classe y inconnue (réalisation de la variable aléatoire Y). Avec l'EPE ou le risque espéré, on mesure donc une espérance des écarts de prédiction lorsque le couple de variables aléatoires X, Y varie selon une distribution $P_{X,Y}(x, y)$. Le meilleur classifieur est celui qui rend l'EPE minimale. En apprentissage statistique, la difficulté principale vient du fait que la loi des données n'est généralement pas connue. Seul l'échantillon d'apprentissage nous informe «empiriquement» sur la distribution des données. La taille de l'échantillon est donc un élément fondamental. Plus on dispose de données supervisées, plus notre échantillon d'apprentissage est significatif et révélateur de la loi générale et inconnue des données. Dans le cas d'un

échantillon limité, on va donc chercher à contourner cette difficulté en introduisant des connaissances ou hypothèses supplémentaires. On tentera ainsi d'approcher (mathématiquement ou empiriquement) l'EPE en vue d'identifier les meilleurs prédicteurs possibles.

Dans le cas particulier de la classification, la loi conditionnelle de Y sachant X est discrète. On peut donc développer l'EPE sous la forme suivante :

$$R(h) = EPE(h) = E_{(X,Y)}[e(h(X), Y)] = \int_{\mathcal{X}} P_X(x) \left[\sum_{y \in \mathcal{Y}} e(h(x), y) P_{Y|X=x}(y) \right] dx \tag{2.7}$$

On utilise dans cette dernière expression le fait que les valeurs à prédire pour la classification sont prises dans un ensemble discret et fini. L'ensemble des idées précédentes peut être repris lorsque l'on veut prédire des valeurs continues, réelles. Dans ce dernier cas, le problème abordé au sens de l'apprentissage supervisé est celui, bien connu, de la régression.

2.3 Prédiction et régression

Un problème de régression fait appel aux mêmes ingrédients qu'un problème de classification supervisée. On s'intéresse aussi à un prédicteur h qui prédit une sortie $\widehat{t}$ à partir d'un stimulus $x \in \mathcal{X}$ selon le schéma suivant :

$$x \in \mathcal{X} = \mathbb{R}^q \rightarrow \boxed{\text{h ?}} \rightarrow \widehat{t} = h(x) \in \mathcal{T} = \mathbb{R}^m$$

On dispose également d'un ensemble de données supervisées qui nous sert d'échantillon d'apprentissage : $D = \{(x_i, t_i)\}_{i=1\ldots n}$. La différence fondamentale est que la variable de sortie (celle qu'il faut prédire) est une variable continue de $\mathcal{T} = \mathbb{R}^m$. On parle aussi de variable de sortie quantitative pour la régression par opposition avec les variables qualitatives utilisées en classification. L'espace d'entrée est le plus souvent de même nature que celui de sortie ($\mathcal{X} = \mathbb{R}^q$) mais de dimension éventuellement distincte ($q \neq m$).

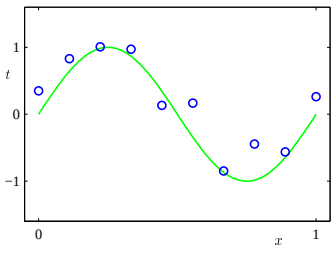


Figure 2.4: Régression et apprentissage.

La figure 2.4 illustre cette nouvelle situation et présente, grâce aux cercles bleus, un ensemble de données d'apprentissage $D = \{(x_i, t_i)\}_{i=1\ldots n}$ avec $x_i \in \mathbb{R}^1$ (en abscisse) et $t_i \in \mathbb{R}^1$ (en ordonnée). Ces points du plan se répartissent autour du

graphe d'une fonction (en vert) inconnue. Très souvent on considèrera que les données d'apprentissage sont des observations bruitées de points situés sur le graphe de cette fonction inconnue f que l'on va chercher à modéliser grâce à une fonction de prédiction (un prédicteur). On considèrera par exemple que les sorties t_i mises en jeu dans les données d'apprentissage de D s'expliquent selon $t_i = f(x_i) + \epsilon_i$ avec ϵ_i une réalisation d'un bruit additif aléatoire. On parlera de données fonctionnelles bruitées additivement.

La figure 2.5 suivante illustre le principe d'une fonction de prédiction h dont le graphe apparaît en rouge. Le prédicteur h qui joue un rôle similaire aux classifieurs vus précédemment prédit des sorties $\widehat{t_i} = h(x_i)$ pour chaque donnée d'entrée x_i. Les prédictions dans ce cas simple sont tout simplement les ordonnées sur le graphe du prédicteur h prises aux abscisses d'entrées. Intuitivement encore apprendre le meilleur prédicteur revient à trouver celui qui minimise l'écart entre prédictions $\widehat{t_i}$ et données disponibles t_i. Dans le cas de données fonctionnelles avec bruit blanc additif (et quelques autres détails), on retrouvera formellement plus loin le résultat intuitif suivant : le meilleur prédicteur h est la fonction inconnue f elle-même.

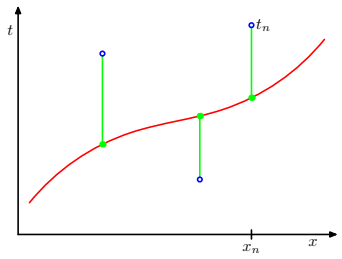


Figure 2.5: Prédiction en régression.

On peut alors reprendre la même démarche que celle décrite précédemment pour l'apprentissage de classifieurs. Les trois choix importants à effectuer sont strictement similaires :

- choix de la fonction perte,
- choix des classes de prédicteurs considérées (un ou plusieurs modèles de complexités distinctes)
- apprentissage, au sein de chaque classe, des meilleurs classifieurs par minimisation du risque empirique.

2.3.1 Fonctions de perte pour la régression

La différence fondamentale entre classification et régression se situe dans la dimension de l'espace de sortie : ici les sorties sont continues et appartiennent à $\mathcal{T} = \mathbb{R}^m$. Les normes L_p sont des choix naturels pour mesurer l'écart entre une sortie prédite $\widehat{t}$ et le vecteur correct t.

$$e(\widehat{t}, t) = ||\widehat{t} - t||_p^p \tag{2.8}$$

En pratique, on utilise très souvent la norme 2 ($p = 2$). Elle est liée aux méthodes des moindres carrés et à leurs interprétations statistiques au sens du maximum de vraisemblance dans le cas de bruits gaussiens [1].

La norme 1 a des propriétés de robustesse (aux données aberrantes) qui la rendent parfois intéressante. D'autres fonctions de pertes robustes sont aussi utilisables. Elles sont de forme générale $c(||\widehat{t} - t||)$ avec une fonction de pondération c choisie de sorte que les très grands écarts entre prédictions et données correctes soient pénalisés de manière (asymptotiquement) constante.

2.3.2 Hiérarchie de modèles de complexité croissante

Comme pour les classifieurs, une hiérarchie de prédicteurs de complexité croissante peut être considérée dans le cas de modèles de prédiction polynomiaux. Il s'agit de considérer un prédicteur paramétrique de type :

$$h_\omega(x) = \omega_0 + \omega_1 x + \omega_2 x^2 + \cdots + \omega_M x^M = \sum_{j=0}^{M} \omega_j x^j \tag{2.9}$$

Le degré M du prédicteur h_ω défini par le vecteur de paramètres $\omega = (\omega_0, \ldots \omega_M)^T$ donne une idée immédiate de sa complexité. De manière cohérente avec les notations précédentes, on dispose d'une hiérarchie de prédicteurs de complexité croissante :

- les prédicteurs constants de degré $M = 0$ (classe $\mathcal{H}_0$),
- les prédicteurs affines $M = 1$ (classe $\mathcal{H}_1$ avec $\mathcal{H}_0 \subset \mathcal{H}_1$),
- les prédicteurs quadratiques (classe $\mathcal{H}_2$ avec $\mathcal{H}_0 \subset \mathcal{H}_1 \subset \mathcal{H}_2$),
- des prédicteurs arbitrairement complexes avec le degré M (classe $\mathcal{H}_M$ avec $\mathcal{H}_0 \subset \mathcal{H}_1 \cdots \subset \mathcal{H}_{M-1} \subset \mathcal{H}_M$).

Dans les illustrations suivantes (figure 2.6) nous retrouvons le besoin de correctement *sélectionner le modèle* le plus approprié. Le prédicteur de degré nul (cas $M = 0$ et graphe rouge de la figure 2.6 (a)) prédit la même sortie pour toutes les entrées x. Il est trop «simple» vis-à-vis de la fonction autour de laquelle les données se structurent (cf. la courbe en vert reprenant la fonction de la figure 2.4). Les erreurs de prédictions sont importantes si les sorties attendues sont autour de 1 ou -1. Les commentaires sont les mêmes pour $M = 1$, (figure 2.6 (b)). Ces prédicteurs n'ont pas assez de degrés de liberté pour s'ajuster correctement aux données d'apprentissage et approcher correctement la fonction inconnue qui les explique (qui est visiblement «plus qu'affine»).

La figure 2.6 (d) présente une situation inverse pour laquelle $M = 9$. Ici le graphe du prédicteur est suffisamment compliqué pour que chaque donnée de l'ensemble d'apprentissage soit expliquée sans erreur. On a pour un vecteur $\omega^* \in \mathbb{R}^{10}$ bien choisi : $\forall i, \widehat{t_i} = h_{\omega^*}(x_i) = t_i$. Cette situation fait écho à celle rencontrée pour la classification et illustrée par la figure 2.3. Dans les deux cas, on n'observe aucune

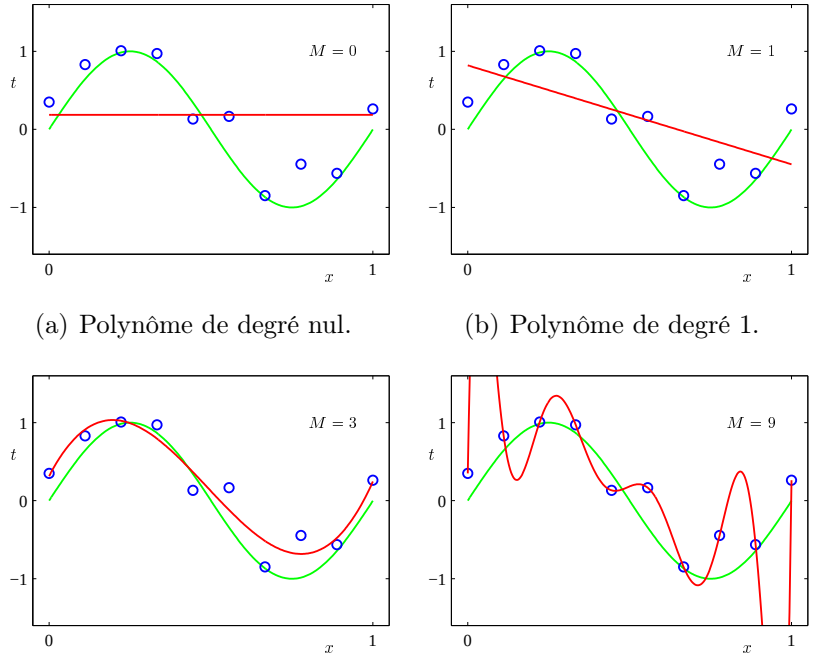
[1] Des rappels sur ce sujet sont proposés plus loin.

erreur/écart de prédiction sur l'ensemble d'apprentissage D. Le prédicteur est suffisamment complexe pour expliquer toutes les données d'apprentissage. Par contre, il est vraisemblablement «trop complexe» pour expliquer de nouvelles données prises hors de D. En particulier, dans la figure 2.6 (d), il apparaît clairement que les prédictions pour de nouvelles données proches du graphe vert, pour les abscisses immédiatement supérieures (resp. inférieures) à 0 (resp. 1), seront totalement erronées. On parle de «sur-apprentissage» (ou «overfitting» en anglais) pour expliquer que de tels prédicteurs, trop complexes, tentent d'expliquer (ici d'interpoler) toutes les données bruitées au lieu d'approcher la fonction f.

Ni trop simple, ni trop compliqué, le prédicteur présenté en rouge dans la figure 2.6 (c) semble être un prédicteur plus approprié pour expliquer à la fois :

- les données d'apprentissage (en bleu) et
- de nouvelles données susceptibles d'être générées comme celles de D (par exemple selon $t'_i = f(x'_i)+\epsilon'_i$ avec la même f mais de nouvelles réalisations pour le bruit ϵ'_i et/ou de nouvelles abscisses distinctes de celles des données présentes dans D).

On retrouve le point central en apprentissage statistique : la capacité de *généralisation* dans des termes exactement similaires à ceux de la section 2.2 précédente.



(a) Polynôme de degré nul.

(b) Polynôme de degré 1.

(c) Polynôme de degré 3.

(d) Polynôme de degré 9.

Figure 2.6: Hiérarchie de modèles pour la régression.

2.3.3 Minimisation du risque empirique pour la régression

L'apprentissage de chaque prédicteur de la figure 2.6 précédente revient à minimiser un risque empirique défini par l'équation (2.3) et déjà introduit pour la classification. La transposition entre classification et régression est immédiate. Seule la fonction de perte change.

Pour apprendre un prédicteur de degré M, à partir d'un ensemble de données supervisées $D = \{(x_i, y_i)\}_{i=1\dots n}$[2] on minimise pour $h \in \mathcal{H}_M$ le risque empirique $R_D(h)$ avec une perte e adaptée à la régression :

$$h^*_M = \underset{h\in\mathcal{H}_M}{\operatorname{argmin}} \left\{ R_D(h) = \frac{1}{n}\sum_{i=1}^{n} e(h(x_i), y_i) \right\} \tag{2.10}$$

Pour une perte quadratique (en norme L_2), apprendre le meilleur prédicteur au sein de $\mathcal{H}_M$ revient à résoudre un problème aux moindres carrés :

$$h^*_M = \underset{h\in\mathcal{H}_M}{\operatorname{argmin}} \left\{ \frac{1}{n}\sum_{i=1}^{n} ||y_i - h(x_i)||_2^2 \right\} \tag{2.11}$$

Dans le cas particulier précédent, on regroupe les coefficients polynomiaux dans le vecteur de paramètre $\omega = (\omega_0, \dots \omega_M)^T$. Les prédictions $h_\omega(x_i) = \sum_{j=0}^{M} \omega_j x_i^j$ sont linéaires en ω. Ce qui conduit à un problème aux moindres carrés linéaires :

$$\omega^*_M = \underset{\omega\in\mathbb{R}^M}{\operatorname{argmin}} \left\{ \frac{1}{n}\sum_{i=1}^{n} ||y_i - (1, x_i, x_i^2, \dots x_i^j, \dots x_i^M)\omega||_2^2 = \frac{1}{n}||\mathsf{Y} - \mathsf{A}\omega||_2^2 \right\} \tag{2.12}$$

où le vecteur Y s'écrit $\mathsf{Y} = (y_1 \dots y_n)^T$ et la matrice A est de taille $n \times M + 1$:

$$\mathsf{A} = \begin{pmatrix} 1 & \cdots & x_1^j & \cdots & x_1^M \\ \vdots & & \vdots & & \vdots \\ 1 & \cdots & x_i^j & \cdots & x_i^M \\ \vdots & & \vdots & & \vdots \\ 1 & \cdots & x_n^j & \cdots & x_n^M \end{pmatrix}$$

Des solutions analytiques pour le problème (2.12) sont connues lorsque le problème est bien posé. Pour un degré M suffisamment élevé, on peut ainsi trouver des polynômes qui interpolent les données d'apprentissage. Dans la figure 2.6 (d), on a $\texttt{err}(h^*_9) = R_D(h^*_9) = 0$. C'est, du point de vue de l'apprentissage et pour la régression, la situation similaire à celle présentée dans la figure 2.3 dans le cas de la classification. L'erreur d'apprentissage (risque empirique obtenu pour la solution h^*_9 ou ω^*_9) est nulle. Pour autant, le meilleur prédicteur n'est pas dans notre exemple celui de degré le plus élévé. Augmenter la complexité (ici le degré) des prédicteurs permet de minimiser l'erreur d'apprentissage mais ne permet pas de minimiser le risque espéré, c'est-à-dire l'EPE. Dans le cas de la régression, l'EPE a une forme similaire à celle rencontrée pour la classification (équation 2.7) mais les données de sortie et la loi conditionnelle associée sont continues :

[2]Dans la suite, les ordonnées à prédire seront généralement notées y_i.

$$EPE(h) = E_{(X,Y)}[e(h(X), Y)] = \int_{\mathcal{X}} P_X(x) \left[\int_{\mathcal{Y}} e(h(x), y) P_{Y|X=x}(y) dy \right] dx \quad (2.13)$$

2.4 Bilan autour du vrai risque : l'EPE

Le cadre de la prédiction supervisée ou de l'apprentissage statistique permet d'aborder de manière unifiée les problèmes de régression et de classification. Dans les deux cas, il s'agit de trouver des prédicteurs dont la capacité de généralisation est la meilleure possible. Il ne faut pas confondre le risque empirique (ou erreur d'apprentissage) utilisé pour ajuster un prédicteur à un échantillon d'apprentissage et le «vrai risque espéré» (ou erreur de généralisation).

Pour un échantillon d'apprentissage donné, le risque empirique décroît progressivement en fonction de la complexité de la classe au sein de laquelle on apprend un prédicteur. Dans la figure 2.7, l'erreur d'apprentissage devient même nulle pour une classe $\mathcal{H}_k$ suffisamment complexe.

Le vrai risque $R(h)$ est quant à lui défini par l'EPE ou Expected Prediction Error pour un prédicteur h et une perte e. Il dépend de la loi inconnue des données :

$$R(h) = EPE(h) = E_{(X,Y)}[e(h(X), Y)] \quad (2.14)$$

Cette quantité mesure l'espérance des pertes dues aux prédictions de h sur l'intégralité des données susceptibles d'être générées (et non pas uniquement sur un ensemble d'apprentissage). Ce «vrai risque» présente un minimum pour la classe de prédicteurs qui a la meilleure capacité de généralisation. Parmi la hiérarchie de classes de prédicteurs de complexités croissantes illustrée par la figure 2.7, l'EPE est minimale pour une classe ni trop simple ni trop complexe.

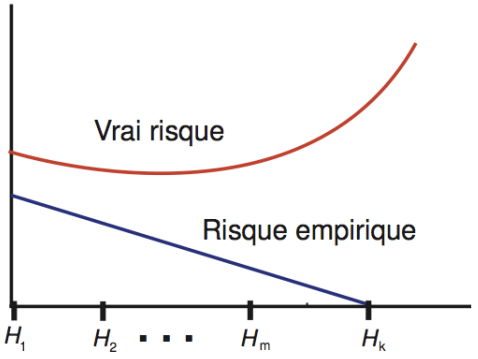


Figure 2.7: Risque empirique et vrai risque espéré.

Dans la suite du cours, nous présenterons les solutions théoriques de la minimisation de l'EPE et leurs traductions pratiques si la loi des données est inconnue.

Thème 3

Approximations de l'EPE

Vincent Charvillat
Décembre 2014

Dans ce thème, on s'intéresse à l'évaluation de l'EPE dont la forme générale, pour un prédicteur ($h \in \mathcal{H}$) et une perte e, est la suivante :

$$R(h) = EPE(h) = E_{(X,Y)}[e(h(X), Y)] = \int_{\mathcal{X}} \int_{\mathcal{Y}} e(h(x), y) P_{X,Y}(x, y) dxdy \qquad (3.1)$$

Pour évaluer l'EPE, la difficulté vient du fait que la loi des données est inconnue. On présente donc d'abord des estimateurs basés sur le risque empirique. Ces estimateurs permettent d'approcher l'EPE, de comparer différents prédicteurs et de sélectionner celui dont la capacité de généralisation est la meilleure. La validation croisée est à utiliser en pratique lorsqu'on dispose d'une quantité limitée de données supervisées.

3.1 Estimateur empirique de l'EPE

3.1.1 Cadre

Soit h^* un prédicteur minimisant le risque empirique associé à un échantillon d'apprentissage $D = \{x_i, y_i\}_{i=1...n}$:

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}}\ R_D(h) = \frac{1}{n} \sum_{i=1}^{n} e(h(x_i), y_i) \qquad (3.2)$$

Nous avons déjà vu que, pour des classes de prédicteurs suffisamment complexes, l'erreur d'apprentissage $\texttt{err}(h^*) = R_D(h^*)$, c'est-à-dire le risque empirique à la solution h^*, est une estimation excessivement optimiste du vrai risque $R(h)$. La figure

2.7 résume ce phénomène dit de «sur-apprentissage» pour des classes de prédicteurs $\mathcal{H}_j$ de complexité croissante avec j. Evaluer la capacité de généralisation d'un prédicteur à partir de l'échantillon D qui a permis de le construire n'est pas satisfaisant.

Pour estimer correctement la capacité de généralisation de h^*, c'est-à-dire $R(h^*) = EPE(h^*)$, nous pouvons utiliser un autre ensemble de données supervisées , **distinct de** D. On appelle cet ensemble de $m \neq n$ nouvelles données, un échantillon de test et on le note généralement T. Si on note à nouveau[1] $z_i = (x_i, y_i)$ les données supervisées contenues dans $T \neq D$, on a simplement m nouvelles réalisations z_i du vecteur aléatoire $Z = (X, Y)$ de loi $P_Z(z) = P_{X,Y}(x, y)$ qui a déjà permis de réaliser (indépendamment) les données de D.

Un estimateur empirique de l'EPE pour h^* est déduit du risque empirique calculé sur T selon :

$$R_T(h^*) = \frac{1}{m} \sum_{i=1}^{m} e(h^*(x_i), y_i) \tag{3.3}$$

Ce risque empirique mesure effectivement une capacité de généralisation via la moyenne empirique des écarts entre les prédictions $h^*(x_i)$ et les sorties attendues y_i sur les nouvelles données de $T = \{x_i, y_i\}_{i=1\ldots m} \neq D$.

3.1.2 Propriétés de l'estimateur du risque empirique

Plus formellement, si on note Z^m la variable aléatoire associée à T : $Z^m = (Z_1, Z_2, \ldots, Z_m)$. Les composantes $Z_i = (X_i, Y_i)$ sont iid ($\forall i, Z_i$ est distribuée selon P_Z). L'estimateur de $R(h) = EPE(h)$ considéré est le suivant :

$$R_{Z^m}(h) = \frac{1}{m} \sum_{i=1}^{m} e(h(X_i), Y_i) \tag{3.4}$$

Cet estimateur a de bonnes propriétés :

- Il est non biaisé : $E_{Z^m}\left[R_{Z^m}(h)\right] = R(h)$
- Sa variance diminue avec le nombre m d'exemples dans l'échantillon de test (propriété de convergence) :

$$\mathrm{Var}_{Z^m}\left[R_{Z^m}(h)\right] = \frac{\mathrm{Var}_{Z=(X,Y)}\left[e(h(X), Y)\right]}{m} \tag{3.5}$$

L'absence de biais est immédiate à démontrer et la réduction de variance découle du théorème suivant : Soit $\{V_i\}$ n v.a. iid tq $\forall i, E[V_i] = \mu$ et $Var[V_i] = \sigma^2$ alors $Var\left[\frac{1}{n}\sum_{i=1}^{n} V_i\right] = \frac{\sigma^2}{n}$. De sorte que le risque empirique $R_{Z^m}(h)$ est un bon estimateur de l'EPE lorsque m est elevé et pour autant que $\mathrm{Var}_{Z=(X,Y)}\left[e(h(X), Y)\right]$ soit bornée.

Ces bonnes propriétés encouragent l'utilisation du risque empirique comme estimateur du vrai risque. En pratique, si l'on dispose d'énormément de données supervisées, il est possible de scinder les données en, d'une part un échantillon D

[1] On pourrait distinguer, sans intérêt réel, $D = \{x_i, y_i\}_{i=1\ldots n}$ et $T = \{x'_i, y'_i\}_{i=1\ldots m}$.

d'apprentissage et, d'autre part, un échantillons de test T. Pour respectivement apprendre des prédicteurs pris dans différentes classes et tester celui qui, parmi eux, généralise le mieux.

3.1.3 Du bon usage du risque empirique

Les bonnes propriétés asymptotiques du risque empirique justifient son utilisation lorsque la taille des échantillons est importante. A titre d'exemple, le sur-apprentissage d'un polynôme de degré 9 illustré par la figure 2.6(d) intervient en raison d'un ensemble d'apprentissage de taille limitée. Dans la figure 3.1, la taille de l'échantillon d'apprentissage augmente considérablement. Cette figure montre qu'apprendre un polynôme assez complexe (degré 9) en minimisant le risque empirique sur un échantillon important retrouve du sens.

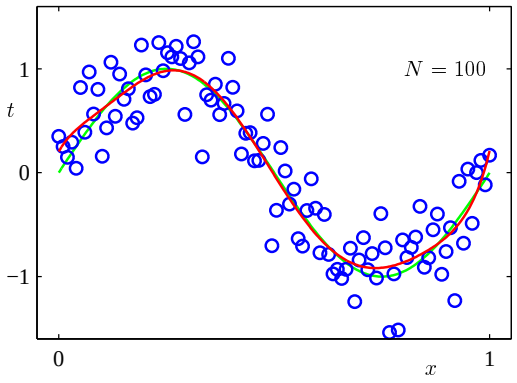


Figure 3.1: Effet positif associé à un grand ensemble d'apprentissage.

En fait, l'approximation du vrai risque par un risque empirique calculé sur un ensemble $S = \{x_i, y_i\}_{i=1...m}$ consiste à estimer empiriquement la loi jointe $P_{X,Y}(x, y)$ par :

$$\widehat{P}_{X,Y}(x, y) = \frac{1}{m} \sum_{i=1}^{m} \delta(x - x_i)\delta(y - y_i) \tag{3.6}$$

Ce qui conduit effectivement à :

$$R(h) = EPE(h) \approx \int_{\mathcal{X}} \int_{\mathcal{Y}} e(h(x), y)\widehat{P}_{X,Y}(x, y)dxdy = R_S(h) \tag{3.7}$$

Pour m suffisamment grand, $R_S(h)$ nous donne une bonne estimation de $R(h)$ ce qui motive l'utilisation de la minimisation du risque empirique pour l'apprentissage et/ou le test d'un prédicteur. Bien entendu, en pratique, le nombre de données supervisées est toujours le facteur limitant.

3.2 Validation croisée

Les techniques précédentes ne s'appliquent donc pas lorsque l'on dispose d'un (unique) ensemble d'apprentissage de taille $n = |D|$ limitée. Séparer $D = \{x_i, y_i\}_{i=1\ldots n}$ pour en extraire un échantillon de test de taille suffisante n'est pas possible.

3.2.1 Validation croisée, "K-fold"

Pour approcher l'EPE lorsqu'on manque de données, la méthodologie de la validation croisée permet d'utiliser :

- toutes les données pour tester un prédicteur h,
- presque toutes les données pour construire h.

L'idée est d'itérer l'estimation de l'erreur de généralisation sur plusieurs échantillons de validation extraits de D puis d'en calculer la moyenne. Moyenner permet de réduire la variance et ainsi d'améliorer la précision de l'approximation associée de l'EPE si la taille de D est limitée.

L'algorithme de la validation croisée (**K-fold Cross-Validation**) est décrit ci-après :

o Découper aléatoirement l'échantillon D en K ensembles disjoints $\{T_1, T_2 \ldots T_K\}$ (approximativement) de même taille $|T_i|$ et réalisant une partition de D.

o **Pour** $i = 1, 2, \ldots K$ **Faire**

1. Apprendre $h^*_{D-T_i}$ par minimisation de $R_{D-T_i}(h)$
2. Evaluer ce prédicteur : $R_i = R_{T_i}(h^*_{D-T_i})$

Fin Pour

o Calculer $CV_{Kfold} = R^K_{CV} = \frac{1}{K}\sum_{i=1}^{K} R_i$

3.2.2 Validation croisée, "Leave-One-Out"

Lorsque $K = |D|$, la validation croisée consiste à exclure successivement chaque donnée de D pour tester le modèle appris en privant D d'une seule donnée. On parle alors de validation croisée **Leave-One-Out (LOO)**.

Dans la séance de travaux pratiques (TP) consacrée à la validation croisée, on montre que le calcul de l'approximation $CV_{LOO} = R^{K=1}_{CV}$ peut parfois s'effectuer rapidement. Si l'on sait facilement passer du prédicteur appris à partir de tout D à celui appris sur D privé d'une seule donnée, on peut éviter les n apprentissages (étape 1.) de la boucle **Pour** ci-dessus.

La notion de **Validation Croisée Généralisée** (GCV en anglais) peut alors être introduite et reliée aux critères de sélection de modèles comme cela est expliqué dans le TP. Le lecteur est encouragé à compléter ces éléments de cours sur la validation croisée avec les explications, compléments et interprétations vues en TP.

3.2.3 Variance du score de validation croisée

Très souvent, les résultats de validation croisée sont présentés sous la forme d'un intervalle :

$$R_{CV}^K \pm \sigma_{CV}^K$$

avec

$$\sigma_{CV}^K \approx \sqrt{\frac{1}{K(K-1)} \sum_{i=1}^{K} (R_i - R_{CV}^K)^2}$$

Cette approximation est partiellement justifiée par les deux points suivants :

(i) la variance σ^2 des R_i vues comme des v.a. peut être estimée empiriquement par la moyenne des écarts quadratiques à la moyenne R_{CV}^K :

$$\sigma^2 \approx \frac{1}{(K-1)} \sum_{i=1}^{K} (R_i - R_{CV}^K)^2$$

(ii) l'estimateur R_{CV}^K est une moyenne de K v.a. de variance σ^2 d'où $\sigma_{CV}^K \approx \sigma/\sqrt{K}$.

La justification n'est que partielle car on a (ii) si les R_i sont indépendantes. Ce n'est pas tout à fait le cas puisque les échantillons d'apprentissage se recouvrent pour $K > 2$. Un tel intervalle visant à estimer l'incertitude liée au score de la validation croisée est, sous cette forme, assez grossier.

Thème 4

Solutions de l'EPE et méthodes dérivées

Vincent Charvillat
Décembre 2014

Dans ce thème, on s'intéresse à la minimisation de l'EPE dont la forme générale, pour un prédicteur ($h \in \mathcal{H}$) et une perte e, est la suivante :

$$R(h) = EPE(h) = E_{(X,Y)}[e(h(X), Y)] = \int_{\mathcal{X}} \int_{\mathcal{Y}} e(h(x), y) P_{X,Y}(x, y) dx dy \qquad (4.1)$$

On peut résoudre formellement le problème de la minimisation de l'EPE. Nous donnons les solutions théoriques pour la régression et la classification. Ces solutions ne sont que «théoriques» puisqu'elles dépendent de la loi inconnue des données $P_{X,Y}(x, y) = P_{Y|X=x}(y) P_X(x)$. Toutefois, ces solutions justifient le bien-fondé (et les limites) de techniques très utilisées en pratique : prédicteurs non-paramétriques aux plus proches voisins, classifieur Bayésien, etc.

4.1 Solution de l'EPE pour la régression

4.1.1 Espérance conditionnelle

On souhaite minimiser l'EPE pour la régression mise sous la forme suivante :

$$R(h) = EPE(h) = \int_{\mathcal{X}} \left[\int_{\mathcal{Y}} (y - h(x))^2 p(y|x) dy \right] p(x) dx \qquad (4.2)$$

Les sorties sont scalaires ($\mathcal{Y} \subset \mathbb{R}$), la perte choisie est $e(y, \widehat{y}) = (y - \widehat{y})^2$ et les lois (continues) sont à densité.

On définit la régression (ou espérance conditionnelle) comme la fonction :

$$r(x) = E[Y|X = x] = \int_{\mathcal{Y}} yp(y|x)dy \tag{4.3}$$

Théorème La régression $r(x)$ est la fonction qui minimise l'EPE (eq. 4.2)

La démonstration est un exercice simple si l'on observe que la minimisation de l'EPE peut se faire indépendamment pour chaque x fixé ou bien si on vérifie que le théorème est vrai en développant :

$$R(h) = \int_{\mathcal{X}} \int_{\mathcal{Y}} (y - r(x) + r(x) - h(x))^2 p(x, y)dxdy \tag{4.4}$$

La connaissance de la loi des sorties conditionnelle à l'entrée $p(y|x) = p(x, y)/p(x)$ rend simple le problème de la minimisation de l'EPE. Nous retrouvons donc ici qu'en supposant l'existence d'une dépendance statistique de Y sur X selon :

$$Y = f(X) + \epsilon \tag{4.5}$$

la solution de la minimisation de l'EPE est évidemment la fonction $x \to f(x)$ qui est la régression $f(x) = E[Y|X = x]$ puique ϵ est un bruit d'espérance nulle, $E[\epsilon] = 0$. Ainsi en régression, la meilleure prédiction de Y pour une entrée $X = x$ donnée est intuitivement l'espérance (c'est-à-dire une moyenne) des sorties pouvant être générées. Bien entendu, si l'on change la perte, la solution change. Un exercice donné en fin de chapitre traite en particulier des pertes L_p pour $p \neq 2$.

4.1.2 Méthode des k plus proches voisins (kppv)

Les méthodes des k plus proches voisins (kppv) sont justifiées par l'intuition précédente (prédire en moyennant les sorties possibles pour une entrée donnée). Si on considère un ensemble d'apprentissage $D = \{x_i, y_i\}_{i=1..n}$, il est improbable d'avoir au sein de D plusieurs entrées égales à un x_0 fixé et donc plusieurs sorties à «moyenner». Le prédicteur aux kppv calcule donc une moyenne empirique des sorties en utilisant les plus proches voisins de x_0 :

$$\widehat{f_{kppv}}(x_0) = \text{moyenne}\left(y_{\sigma(i)}|x_{\sigma(i)} \in V_k(x_0)\right) = \frac{1}{k}\sum_{i=1}^{k} y_{\sigma(i)} \tag{4.6}$$

Dans cette formule, les k plus proches voisins de x_0 appartiennent au voisinage $V_k(x_0)$. Le choix d'une distance permettant une définition formelle de $V_k(x_0)$ est à discuter. Deux approximations sont utilisées vis-à-vis de l'équation (4.3) :

a) l'espérance est approchée par une moyenne empirique,

b) le conditionnement à $X = x_0$ est relâché en prenant des entrées voisines de x_0.

Pour de grands échantillons d'apprentissage (n grand), les points de $V_k(x_0)$ ont des chances d'être proches de x_0 et si k devient grand, la moyenne deviendra suffisamment stable pour obtenir un prédicteur régulier. Il faut en effet bien comprendre que le prédicteur de l'équation (4.6) est par construction constant par morceaux : pour un point très proche de x_0, disons $x_0 + \epsilon$, les k voisins ne changent pas...

En fait, sous des hypothèses de régularité suffisantes pour la loi jointe $p(x, y)$, on peut montrer que :

- Si $n \to \infty$, $k \to \infty$ tq. $k/n \to 0$
- Alors $\widehat{f_{kppv}}(x) \to E[Y|X = x]$

Ces propriétés sont très prometteuses. Est-il alors raisonnable, pour construire un prédicteur, de postuler des dépendances élémentaires entres entrées et sorties, de postuler un modèle hypothétiquement linéaire alors que le prédicteur empirique de l'équation (4.6) tend vers la solution ? En d'autres termes, tenons-nous le prédicteur idéal ? Malheureusement la réponse est négative. Le fléau de la dimension (de l'espace d'entrée $\mathcal{X}$) va limiter l'intérêt de cette approche.

4.1.3 Fléau de Bellman ("Curse of dimensionality")

Le prédicteur aux kppv de l'équation (4.6) est une méthode de prédiction «locale» opérant au voisinage $V_k(x_0)$ d'un x_0 donné. Insistons sur le fait que nous *souhaitons* qu'elle soit locale pour que l'approximation b) du paragraphe précédent soit la plus raisonnable possible. Dans de nombreux problèmes pratiques où on veut mettre au point un mécanisme de prédiction à partir de mesures (par exemple physiques, biologiques etc.), les (mesures) d'entrée sont nombreuses. Ce qui revient souvent à $x_0 \in \mathcal{X} \subset \mathbb{R}^p$ avec p grand. Le problème du fléau de la dimension est qu'une méthode par voisinage dans $\mathbb{R}^p$ avec p grand n'est plus du tout «locale».

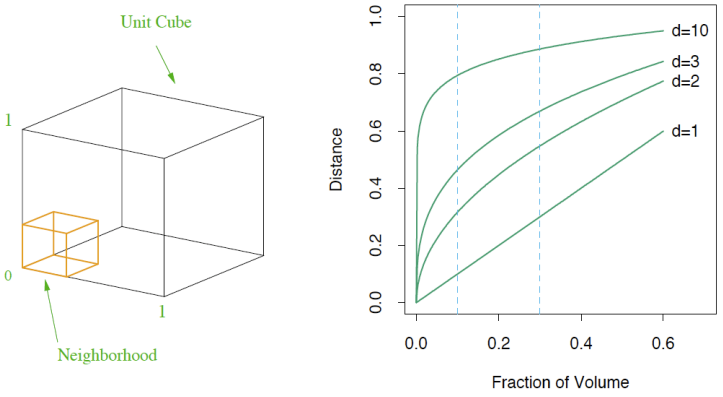


Figure 4.1: Illustration du "Curse of dimensionality", d'après Hastie et al.

Pour comprendre le problème, considérons une procédure aux plus proches voisins pour des entrées x_i uniformément réparties dans un (hyper)cube unité de $\mathbb{R}^p$. La

figure 4.1 donne, à gauche, une représentation de ce cube unité en dimension p. Considérons un voisinage V autour d'un point d'intérêt x_0 qui représente une fraction r des obsservations disponibles : on veut par exemple que le voisinage V intègre 5 % ou 10 % des données. Le voisinage représente une fraction r du volume du cube unité. En dimension p, un tel voisinage a un volume équivalent à un cube dont les côtés ont pour longueur $l_p(r) = r^{1/p}$: un tel cube de voisinage apparaît en rouge dans le cube unité et vérifie :

$$\text{Volume}(V) = \underbrace{r^{1/p} \times \cdots \times r^{1/p}}_{p \text{ fois}} = r$$

La partie droite de la figure 4.1 donne les graphes de $r \to l_d(r) = r^{1/d}$ avec la dimension d variant de 1 à 10 et r, en abscisse, variant de 0.0 à 0.6. On peut voir sur ce type de figure que pour capturer 1% du volume en dimension 10, le côté de V mesure 0.63 : $l_{10}(0.01) = 0.63$. Dès lors V n'est plus un voisinage local ce qui rend la prédiction aux kppv inappropriée. La réduction de r n'est pas une bonne solution puisqu'elle conduira, comme nous le verrons dans le thème suivant à une augmentation de variance dans la prédiction.

De surcroît, il est évident que pour échantillonner l'espace $\mathcal{X}$ avec une densité constante, le fléau de la dimension intervient : si nous échantillonnons $[0, 1]$ avec $N_1 = 100$ exemples, il faut échantillonner $[0, 1]^{10}$ avec $N_{10} = 100^{10}$ exemples pour obtenir la même densité. En d'autres termes, la densité des échantillons d'entrée (les x_i) en haute dimension est toujours faible. Cela limite également l'intérêt des méthodes de type kppv en grande dimension.

Disons pour conclure cette discussion, que l'utilisation des prétraitements visant une réduction de dimension (cf. les prétraitements vus au début du cours) prend donc tout son sens en amont d'une prédiction aux plus proches voisins.

4.2 Solution de l'EPE pour la classification

On considère un classifieur $h : \mathcal{X} \to \mathcal{Y}$ dont :

- les données d'entrée sont dans $\mathcal{X}$ et sont distribuées selon la loi de densité p,
- l'espace discret de sortie à K classes est $\mathcal{Y} = \{\omega_1, \ldots, \omega_K\}$ avec une loi conditionnelle discrète sachant l'entrée $X = x$ qui est définie par les K probabilités $P_{Y|X=x}(\omega_i) = Pr(Y = \omega_i|X = x) = Pr(\omega_i|X = x)$

L'EPE que l'on veut minimiser prend la forme suivante :

$$EPE(h) = \int_{\mathcal{X}} p(x) \left[\sum_{y \in \{\omega_1, \ldots, \omega_K\}} e(h(x), y) Pr(Y = y|X = x) \right] dx \tag{4.7}$$

Ou de manière plus compacte encore :

$$EPE(h) = E_X \left[\sum_{i=1}^{K} e(h(X), \omega_i) Pr(\omega_i|X) \right] \tag{4.8}$$

4.2.1 Solution pour une perte générale

Comme vu précédemment pour la régression, il faut observer que la minimisation de l'EPE (pour des pertes positives) peut s'effectuer ponctuellement pour chaque x. Ainsi le classifieur optimal en x_0 est :

$$\widehat{h}(x_0) = \omega_{i_0} = \underset{g \in \{\omega_1, \ldots, \omega_K\}}{\operatorname{argmin}} \sum_{i=1}^{K} e(g, \omega_i) Pr(\omega_i | X = x_0)$$

Les interprétations suivantes sont possibles :

- le terme $e(g, \omega_i)$ représente le coût de décider la classe g alors que la classe correcte est ω_i,
- le terme $\sum_{i=1}^{K} e(g, \omega_i) Pr(\omega_i | X = x_0)$ représente le risque (coût) de prédire g observant x_0 en entrée

Lorsque l'on modélise un problème de classification, le choix des $K \times K$ valeurs $e(\omega_j, \omega_i)$ avec $i, j \in \{1, \ldots K\}^2$ est donc une étape fondamentale. On remarquera en particulier qu'il est parfois pertinent de prendre : $e(\omega_j, \omega_i) \neq e(\omega_i, \omega_j)$ avec $i \neq j$. On pourra aussi observer que le choix de chaque perte permet naturellement de considérer une $K + 1^{\text{ième}}$ classe, dite *classe de rejet*, pour laquelle aucune décision de classification n'est prise. Classe additionnelle pour laquelle des pertes adaptées au problème et à l'application doivent être adroitement choisies.

4.2.2 Solution pour une perte non-informative

On dit que la perte 0-1 notée e_z est non-informative par opposition au cas précédent où chaque erreur de classification a un coût particulier. Avec la perte 0-1, c'est du tout ou rien : ou bien on a raison (on classe/prédit bien) ou bien on a tort :

$$e_z(\widehat{y}, y) = \begin{cases} 0 \ si \ y = \widehat{y} \\ 1 \ si \ y \neq \widehat{y} \end{cases} \tag{4.9}$$

En choisissant la perte e_z, il est immédiat de montrer que le classifieur optimal devient :

$$Opt(x_0) = \omega_{i_0} = \underset{g \in \{\omega_1, \ldots, \omega_K\}}{\operatorname{argmax}} Pr(g | X = x_0)$$

Certains appellent ce classifieur, le classifieur Bayésien, car il maximise la probabilité a posteriori de la classe (c'est-à-dire sachant la mesure d'entrée).

Grâce au résultat précédent, $Opt(x) = \text{argmax}_g Pr(g|X = x)$, on peut aussi bâtir un classifieur aux k plus proches voisins en approchant empiriquement la probabilité $Pr(g|X = x)$. Il s'agit de prédire pour une entrée x la classe la plus probable ω_0 compte-tenu d'un ensemble d'apprentissage $D = \{x_i, y_i\}$ au sein duquel certains x_i sont voisins de x et majoritairement associés à la classe ω_0. Ce classifieur fait l'objet d'un exercice ci-après.

Thème 5

Compromis Biais-Variance

Vincent Charvillat
Décembre 2014

Dans ce thème, on aborde les conditions défavorables rencontrées en pratique lorsqu'on met en œuvre un mécanisme de prédiction. Par conditions défavorables, on entend une quantité limitée de données supervisées et la difficulté à sélectionner des prédicteurs de complexité satisfaisante. En décomposant l'EPE, c'est-à-dire le risque espéré, on montre clairement que des prédicteurs trop complexes fournissent des prédictions trop dépendantes des ensembles d'apprentissage considérés et, qu'inversement, des prédicteurs trop simples peuvent conduire à des erreurs systématiques de prédiction. Le compromis sous-jacent est appelé «compromis biais-variance». On peut établir un compromis acceptable en sélectionnant un modèle avec les méthodes empiriques vues précédemment (par validation croisée par exemple) ou en utilisant les mécanismes de contrôle de la complexité présentés dans ce thème : la sélection de modèle ou la régularisation.

5.1 Décomposition de l'EPE

5.1.1 Cadre

On se place dans le cadre de la régression, vue comme un problème d'apprentissage supervisé. On cherche des prédicteurs dans une classe $\mathcal{H}$ fixée. Pour un ensemble d'apprentissage D et grâce à la minimisation d'un risque empirique, on peut donc apprendre un prédicteur $h_D \in \mathcal{H}$ que l'on évalue par :

$$EPE(h_D) = \int_{\mathcal{X}} \int_{\mathcal{Y}} (y - h_D(x))^2 P_{X,Y}(x,y) dxdy \tag{5.1}$$

La dépendance du prédicteur appris vis-à-vis de D (notée h_D) est centrale dans ce qui suit. En particulier, il est clair que le prédicteur n'est pas forcément le

meilleur atteignable dans $\mathcal{H}$. Comme D est de taille limitée, on a vraisemblablement $h_D \neq h^*_{\mathcal{H}}$ avec $h^*_{\mathcal{H}} = \underset{h \in \mathcal{H}}{\operatorname{argmin}}\, EPE(h)$. Nous allons étudier la sensibilité de h_D au choix de l'échantillon D ou, en d'autres termes, comment se comportent les prédictions lorsque D varie.

Pour avancer, on suppose l'existence d'une dépendance fonctionnelle de Y sur X selon :

$$Y = f(X) + \epsilon \tag{5.2}$$

où :

- f est une fonction réelle inconnue $f : \mathbb{R} \to \mathbb{R}$,
- ϵ est un bruit (variable aléatoire réelle) d'espérance nulle : $E(\epsilon) = 0$, indépendant de X,
- on suppose en outre $\mathrm{Var}(\epsilon) = \sigma^2$,
- Y est une variable aléatoire réelle (v.a comme fonction de v.a.).

5.1.2 Décompositions bruit-biais-variance

Nous allons décomposer l'EPE. Dans un premier temps, grâce à nos hypothèses, il est assez aisé de décomposer l'EPE sous la forme suivante :

$$EPE(h_D) = \sigma^2 + \int_{\mathcal{X}} (f(x) - h_D(x))^2 P_X(x) dx \tag{5.3}$$

On retrouve bien alors que $f(x) = E(Y|X = x)$ est la solution optimale de la minimisation de l'EPE (qui n'est pas forcément dans $\mathcal{H}$). La même décomposition opérée ponctuellement en x_0 conduit à :

$$EPE(x_0, h_D) = E_{Y|X}\left[(y - h_D(x))^2 | x = x_0\right] = \sigma^2 + (f(x_0) - h_D(x_0))^2 \tag{5.4}$$

Lorsque D varie, h_D varie. De même, ponctuellement, lorsque D varie, $h_D(x_0)$ varie autour de $E_D\left[h_D(x_0)\right]$. L'introduction de l'espérance des prédictions permet d'obtenir la décomposition bruit-biais-variance suivante :

$$EPE(x_0) = E_D\left[EPE(x_0, h_D)\right] = \sigma^2 + (f(x_0) - E_D\{h_D(x_0)\})^2 + Var_D\{h_D(x_0)\} \tag{5.5}$$

On écrit usuellement :

$$\text{risque espéré} = \text{bruit} + (\text{biais})^2 + \text{variance} \tag{5.6}$$

L'erreur de généralisation est décomposée en les trois termes suivants (voir aussi la figure 5.1) :

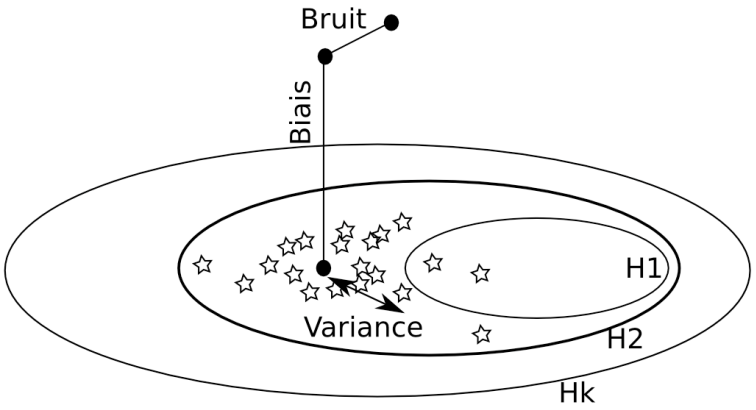


Figure 5.1: Illustration de la décomposition bruit-biais-variance.

- l'erreur due au fait que la classe de prédicteurs choisie ne contient pas nécessairement la solution optimale au problème de minimisation de l'EPE. On appelle cette partie de l'erreur le biais car l'espérance des prédictions, lorsque les ensembles d'apprentissage varient, peut ne pas être égale à la valeur optimale.

- le terme de variance qui mesure l'écart entre les prédictions issues de chaque ensemble d'apprentissage et l'espérance des prédictions. Il s'agit en d'autres termes de mesurer la sensibilité de la prédiction au choix d'un ensemble d'apprentissage particulier.

- le terme de bruit est quant à lui irréductible. Il s'agit de la variance du bruit additif qui contamine intrinsèquement les données de sortie. Ce bruit ne peut pas être prédit.

Le terme de bruit minore donc l'EPE : si la classe des prédicteurs choisie contenait la solution, si l'ensemble d'apprentissage était de taille illimitée (et si nos capacités de calcul l'étaient aussi) on pourrait espérer réduire l'EPE à ce seul terme et retrouver la solution optimale $f(x) = E[Y|x]$.

5.1.3 Illustration pratique

La décomposition précédente permet de bien comprendre les faibles capacités de généralisation de prédicteurs trop simples ou trop compliqués. Un prédicteur trop simple présente une faible variance mais la paye au prix d'un biais important. Inversement, un prédicteur trop compliqué conduira à un faible biais mais de fortes variances de prédiction. Les figures 5.2 et 5.3 illustrent ce phénomène. La solution $f(x)$ polynomiale de degré 6 au problème de régression est donnée par la courbe de Bézier en bleu. Deux ensembles d'apprentissage D_1 et D_2 sont utilisés dans chacune des figures avec, chaque fois, deux modèles de prédiction : l'un trop simple de degré 2 (< 6) et l'autre trop complexe de degré 11 (> 6). Pour des abscisses bien choisies, il est très facile de visualiser les termes en $(f(x_0) - E_D\{h_D(x_0)\})^2$ et $Var_D\{h_D(x_0)\}$.

La figure 5.1 suggère la même chose : elle montre des classes imbriquées de prédicteurs $\mathcal{H}_1 \subset \mathcal{H}_2 \cdots \subset \mathcal{H}_k$. La classe $\mathcal{H}_1$ des prédicteurs les plus simples implique un biais important. La taille plus importante de $\mathcal{H}_k$, regroupant des prédicteurs complexes, implique une variance significative lorsque l'ensemble d'apprentissage varie.

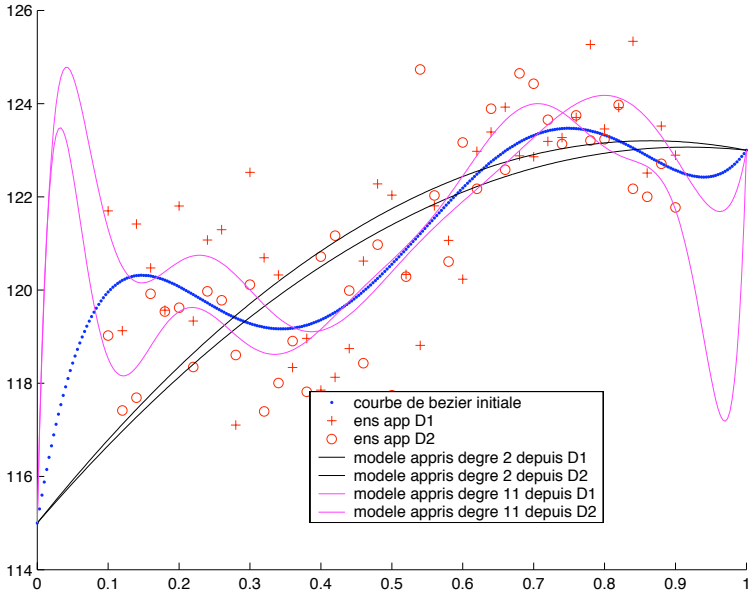


Figure 5.2: Illustration de la décomposition bruit-biais-variance.

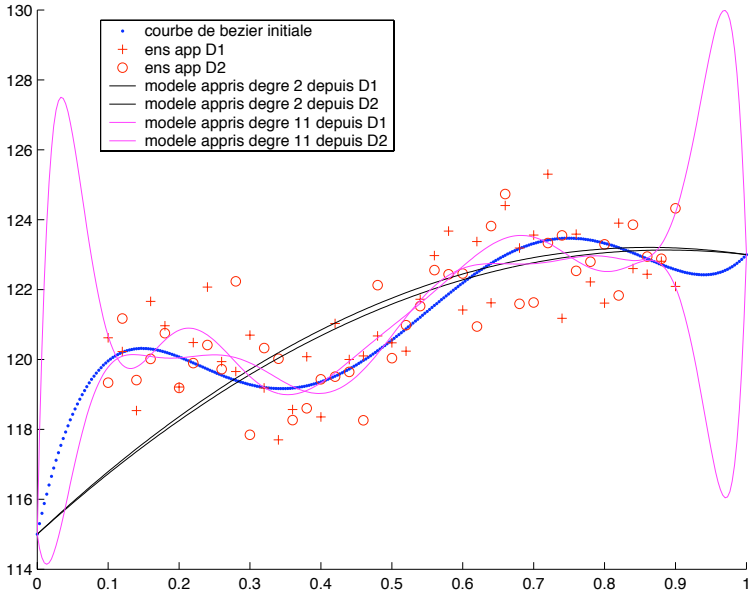


Figure 5.3: Illustration de la décomposition bruit-biais-variance.

5.1.4 Cas du prédicteur aux kppv

On veut décomposer l'EPE pour les prédicteurs aux kppv. On reprend les hypothèses (cf. section 5.1.1) qui nous ont conduit à la décomposition suivante :

$$EPE(x_0) = \sigma^2 + \left(f(x_0) - E_D\{h_D^{\text{kppv}}(x_0)\}\right)^2 + Var_D\left\{h_D^{\text{kppv}}(x_0)\right\} \tag{5.7}$$

Dans l'équation précédente h_D^{kppv} est le prédicteur aux k plus proches voisins déduit d'un ensemble d'apprentissage $D = \{(x_i, y_i)\}_{i=1\ldots n}$:

$$h_D^{\text{kppv}}(x_0) = \text{moyenne}\left(y_{\sigma(i)} | x_{\sigma(i)} \in V_k(x_0), (x_{\sigma(i)}, y_{\sigma(i)}) \in D\right) = \frac{1}{k}\sum_{i=1}^{k} y_{\sigma(i)} \tag{5.8}$$

On considère des ensembles d'apprentissage comme ceux montrés dans les figures précédentes. Pour deux ensembles d'apprentissage $D_1 = \{(x_i, y_i)\}_{i=1\ldots n}$, $D_2 = \{(x_i, y_i')\}_{i=1\ldots n}$, les abscisses $\{x_i\}_{i=1\ldots n}$ sont constantes et seules les n réalisations indépendantes et identiquement distribuées des bruits additifs diffèrent : $y_i = f(x_i) + \epsilon_i$, $y_i' = f(x_i) + \epsilon_i'$. Lorsque D varie, seul le vecteur des bruits iid $E = (\epsilon_1, \ldots, \epsilon_i, \ldots, \epsilon_n)^T$ varie. On aura donc :

$$E_D\{.\} = E_X E_{E|X}\{.\} \to E_E\{.\} \tag{5.9}$$

On peut alors montrer pour $x_{\sigma(i)} \in V_k(x_0)$:

$$EPE(x_0) = \sigma^2 + \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_\sigma(i))\right)^2 + \frac{\sigma^2}{k} \tag{5.10}$$

On en déduit que le paramètre k des «k plus proches voisins» permet de réaliser un compromis biais-variance en contrôlant la régularité du prédicteur.

5.2 Régularisation

Dans l'exemple précédent, le contrôle de complexité peut donc s'opérer via un (hyper)paramètre qui permet de régulariser le prédicteur en limitant sa variance au prix d'un biais potentiellement plus élevé.

Le mécanisme général sous-jacent est appelé *régularisation*. L'idée principale consiste à chercher un prédicteur dans une classe la plus générale possible mais de contraindre ces prédicteurs (potentiellement trop complexes) à être suffisamment réguliers pour éviter l'écueil du sur-apprentissage et l'excès de variance associé. En revenant au cas illustré par la figure 5.3, il s'agirait d'utiliser les prédicteurs polynomiaux les plus complexes (par exemple de degré 11) pour modéliser les données disponibles tout en contraignant ces modèles à être suffisamment réguliers pour éviter les phénomènes d'«oscillations» excessives du prédicteur lorsqu'il cherche à interpoler les données d'apprentissage.

Nous présentons trois approches classiques de la régularisation dans ce paragraphe :

- les splines de lissage,
- l'introduction de connaissances a priori via l'estimation MAP,
- la *ridge regression*, que nous pourrions traduire par «régression écrêtée».

5.2.1 Splines de lissage

Nous nous intéressons ici à la régression à partir d'un ensemble d'apprentissage $D = \{(x_i, y_i)\}, i = 1 \dots n$ tel que $\forall i \ (x_i, y_i) \in \mathbb{R} \times \mathbb{R}$. Nous considérons des prédicteurs généraux qui sont les fonctions de classe C_2 (deux fois continûment dérivables). On cherche alors à apprendre le meilleur prédicteur h^* en minimisant pour $h \in C_2$ une somme pénalisée de résidus aux moindres carrés [1]:

$$PRSS_\lambda(h) = \sum_{i=1}^{n} (y_i - h(x_i))^2 + \lambda \int \{h''(t)\}^2 dt \tag{5.11}$$

On dit que l'(hyper)paramètre λ est un paramètre de lissage à bien choisir. Le premier terme du critère $PRSS$ est un terme d'attache aux données d'apprentissage. Le second terme pénalise des courbures excessives pour le prédicteur qui est ainsi régularisé[2]. L'hyperparamètre λ établit un compromis entre les deux termes. On a en particulier :

- $\lambda = 0$ revient à accepter que le prédicteur interpole les données de D,
- $\lambda = \infty$ revient à chercher une droite de régression (aucune courbure n'est tolérée).

Entre ces deux situations extrêmes allant du modèle affine le plus simple au modèle d'interpolation le plus compliqué, on espère trouver le paramètre λ qui revient à travailler dans la classe de fonctions de complexité optimale. Cette optimalité est à comprendre au sens de l'apprentissage, elle consiste à trouver le meilleur λ^* associé à la capacité de généralisation la plus grande. En pratique, il est possible de trouver ce λ^* en utilisant des techniques vues précédemment comme la validation croisée. On parle aussi de «degré de liberté effectif» pour λ car il joue le rôle d'un nombre «continu» de degrés de liberté.

La solution h^* au problème de minimisation du critère précédent est connue sous le nom de «spline cubique naturelle». Le développement de la preuve est hors sujet ici. Disons simplement qu'une telle «spline naturelle» se met finalement sous la forme simple suivante :

$$h^*(x) = \sum_{j=1}^{n} \beta_j N_j(x) \tag{5.12}$$

Sous cette forme, les $N_j(x)$ forment pour $j \in 1 \dots n$ une base de fonctions pour représenter cette famille de splines. Le critère (équation 5.11) se ramène alors au cas (linéaire) de la *ridge regression* discuté ci-après.

[1] Le critère $PRSS$ est mis pour *Penalized Residual Sum of Squares*.

[2] En toute rigueur un espace de Sobolev doit être considéré avec ce terme.

5.2.2 Régularisation via l'estimateur MAP

En régression, la minimisation du risque empirique avec une perte quadratique et des données fonctionnelles (contaminées par un bruit centré additif) revient à résoudre un problème aux moindres carrés ordinaires qui coïncide également avec une estimation au sens du Maximum de Vraisemblance (MV). Dans ce contexte, l'apprentissage d'un prédicteur paramétrique $x \rightarrow h(\beta_D, x) = h_{\beta_D}(x)$ pris dans une classe $\mathcal{H}$ revient à trouver les paramètres β_D^{MV} qui expliquent au mieux un ensemble d'apprentissage $D = \{(x_i, y_i)\}, i = 1 \dots n$ selon :

$$\beta_D^{MV} = \underset{\beta}{\text{argmax}}\ L(D|\beta) = p(D|\beta) \Leftrightarrow \beta_D^{MV} = \underset{\beta}{\text{argmin}}\ R_D(h_\beta) \tag{5.13}$$

Equations pour lesquelles $L(D|\beta)$ ou $p(D|\beta)$ représentent la vraisemblance des données d'apprentissage D sachant β. On sait qu'une maximisation de L (ou de la «log-vraisemblance» lnL) revient à minimiser le risque empirique R_D c'est-à-dire un critère aux moindres carrés.

L'estimateur du Maximum A Posteriori (MAP) de β consiste à adopter une posture Bayésienne en introduisant une connaissance a priori sur β ou plus précisèment sur la *loi de* β. C'est cette connaissance qui va être l'élément régularisant recherché. Le paramètre β est donc considéré comme une variable aléatoire dont la loi a priori est donnée et notée $L(\beta)$. La démarche Bayésienne consiste finalement, pour l'estimateur MAP, à maximiser la loi a posteriori de β sachant D selon

$$\beta_D^{MAP} = \underset{\beta}{\text{argmax}}\ L(\beta|D) \propto L(D|\beta)L(\beta) \tag{5.14}$$

Un exemple simple, convaincra aisément le lecteur que le critère $L(D|\beta)L(\beta)$ conduit comme précédemment à pénaliser le terme d'attache aux données (la vraisemblance $L(D|\beta)$) avec un terme $L(\beta)$ qui pénalise un écart potentiel entre les paramètres appris et ceux attendus. L'exemple le plus simple est celui de la droite de régression pour laquelle on considère un ensemble d'apprentissage $D = \{(x_i, y_i)\}, i = 1 \dots n$ bâti selon :

$$y_i = a^* x_i + b^* + \epsilon_i \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2) \quad \text{iid} \tag{5.15}$$

Les paramètres du modèle exact $\beta^* = (a^*, b^*)^T$ étant inconnus, on veut les estimer en introduisant des connaissances a priori sur la loi de $\beta = (a, b)^T$ vu comme un vecteur aléatoire. On peut par exemple supposer que :

- la pente a suit une loi $\mathcal{N}(0, \sigma_a^2)$. Ce qui indique que la droite de régression recherchée est plutôt horizontale ou, plus précisément, de pente statistiquement d'autant plus faible que la variance σ_a^2 est petite,
- l'ordonnée à l'origine b est uniformément répartie dans l'intervalle $[0, 1]$: $b \sim \mathcal{U}[0, 1]$.

Sous ces hypothèses, on peut immédiatement vérifier que :

$$L(\beta|D) \propto \exp -\frac{1}{2}\left(\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - ax_i - b)^2 + \frac{1}{\sigma_a^2}a^2\right) \tag{5.16}$$

On retrouve exactement la même idée que précédemment selon laquelle l'hyper paramètre $\frac{1}{\sigma_a^2}$ pondère un terme aux moindres carrés pour l'attache aux données et un terme de régularisation (en a^2) qui limite ici l'amplitude de la pente. Plus la variance σ_a^2 est faible, plus le poids du terme régularisant est fort dans le compromis recherché. Cette modélisation conduit à l'estimation MAP suivante pour la pente :

$$\widehat{a}_{MAP} = \frac{\frac{1}{n}\sum_i x_i y_i - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2 + \frac{\sigma^2}{n\sigma_a^2}} \tag{5.17}$$

Ce qui nous permet de vérifier que l'effet de régularisation attendu est bien présent : $\sigma_a^2 \to 0 \Rightarrow \widehat{a}_{MAP} \to 0$. Cet exemple souligne la généralité d'une méthode Bayésienne de régularisation via l'introduction de connaissances/lois a priori. En pratique, on s'éfforce simplement d'introduire des «lois a priori conjuguées» pour que les modèles régularisés conduisent, autant que possible, à des solutions analytiques.

5.2.3 Ridge regression

L'approche de la régularisation par *ridge regression* est une généralisation de l'exemple précédent pour lequel on limitait l'amplitude d'un paramètre à apprendre (la pente a de la droite de régression). En *ridge regression*, on souhaite écrêter ou «rétrécir» un ou plusieurs paramètres[3] pour régulariser le prédicteur h associé. On pénalise donc de trop fortes amplitudes pour le ou les paramètres concernés selon :

$$\widehat{\beta}_{\text{ridge}} = \underset{\beta=(\beta_1 \ldots \beta_p)^T \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - h(\beta, x_i))^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{5.18}$$

Les notations sont les mêmes que précédemment pour les données (les x_i, y_i de D). L'hyperparamètre ($\lambda \geq 0$) contrôle ici aussi la complexité du prédicteur (pour lequel le choix d'une paramétrisation compatible avec cette approche est un problème clé). Plus λ est grand plus le retrécissement des composantes β_j s'imposera vis-à-vis du terme d'attache aux données d'apprentissage. De manière équivalente, on peut aussi présenter la *ridge regression* comme la solution du problème d'optimisation avec contrainte suivant :

$$\begin{aligned} \widehat{\beta}_{\text{ridge}} = \underset{\beta=(\beta_1 \ldots \beta_p)^T \in \mathbb{R}^p}{\operatorname{argmin}} & \sum_{i=1}^{n} (y_i - h(\beta, x_i))^2 \\ \text{s.c.} & \sum_{j=1}^{p} \beta_j^2 \leq s \end{aligned} \tag{5.19}$$

On considère de manière privilégiée le cas d'un prédicteur linéaire (ou affine) en β pour lequel on peut reformuler vectoriellement le critère régularisé. Pour $h(\beta, x) = C(x)^T\beta$ on a trivialement un vecteur Y et une matrice X similaires à ceux utilisés dans le modèle de régression linéaire simple qui ramènent le critère de la *ridge regression* à l'expression suivante :

$$PRSS_\lambda(\beta) = (Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta \tag{5.20}$$

Il est alors aisé de montrer que :

[3] On parle donc aussi en anglais de *shrinkage method*.

$$\widehat{\beta}_{\text{ridge}} = \underset{\beta=(\beta_1 \ldots \beta_p)^T \in \mathbb{R}^p}{\operatorname{argmin}} PRSS_\lambda(\beta) \Rightarrow \widehat{\beta}_{\text{ridge}} = (X^T X + \lambda \text{Id}_{p\times p})^{-1} X^T Y \quad (5.21)$$

On peut naturellement donner de nombreuses interprétations de cette dernière équation par comparaison avec celle de la pseudo-inverse usuelle. Une des plus simples est d'observer que la matrice de lissage S_λ remplace la matrice chapeau $H = X(X^T X)^{-1} X^T$ selon :

$$\widehat{Y} = X(X^T X + \lambda \text{Id}_{p\times p})^{-1} X^T Y = S_\lambda Y \quad (5.22)$$

Là où la trace de H était égale au nombre p de paramètres, les degrés effectifs de liberté (valeurs continues) sont désormais définis par :

$$df(\lambda) = Trace(S_\lambda) \quad (5.23)$$

Le nombre effectif de degrés de liberté est donc d'autant plus faible que λ est grand. Il tend vers 0 si $\lambda \to \infty$ et, inversement, $df(\lambda) \to Trace(H) = p$ si $\lambda \to 0$ (absence de régularisation).

Nous dirons pour conclure ce paragraphe sur la régularisation par *ridge regression*, que des pénalisations de l'amplitude des paramètres β_j peuvent aussi s'entendre au sens de la norme L_1 (et non au sens de L_2 comme présenté ici). La pénalité dite de «Lasso» s'exprime simplement par $\sum_{j=1}^{p} |\beta_j| \leq t$. C'est une bonne manière, si t diminue, de forcer certains des β_j à s'annuler est de sélectionner les composantes grâce auxquelles on veut prédire !

5.3 Sélection de modèle

Une approche duale de la régularisation pour réaliser un compromis biais-variance consiste à pénaliser les modèles de prédiction trop complexes. L'idée force est d'observer que l'erreur d'apprentissage sous-estime le risque espéré (l'EPE) pour un prédicteur trop complexe. On dit que l'écart (le biais) entre l'EPE et l'erreur d'apprentissage est un terme d'optimisme[4]. Pour un prédicteur h, ce terme d'opti misme pourrait être défini comme :

$$\texttt{op} = EPE(h) - \texttt{err}(h) \quad (5.24)$$

Ce terme d'optimisme est d'autant plus grand que le modèle est complexe. Le principe de la sélection de modèle consiste à utiliser une estimation $\widehat{\texttt{op}_p}$ de ce terme d'optimisme pour un prédicteur de complexité p (p est typiquement le nombre de paramètres ou ddl considéré). Cette estimation permet alors de pénaliser un terme d'attache aux données d'apprentissage $D = \{(x_i, y_i)\}, i = 1 \ldots n$. On apprend alors en minimisant un critère de sélection de modèle C_{sm} pour une classe de prédicteurs h_p de complexité p :

[4] On parle d'optimisme puisque la capacité de généralisation est surévaluée en utilisant les données ayant servi à l'apprentissage comme données de test.

$$C_{sm}(h_p) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \underbrace{h_p(x_i)}_{\widehat{y_i}})^2 + \widehat{\mathtt{op}_p} \tag{5.25}$$

5.3.1 Critère AIC d'Akaïke

Dans le cadre de la régression avec modèle gaussien et utilisation du risque quadratique, le critère d'information d'Akaïke (critère AIC mis pour *Akaïke Information Criterion*) coïncide avec le premier critère de pénalisation apparu dans la litérature (critère C_p de Mallows). Le critère AIC prend la forme suivante :

$$AIC_p = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 + 2\frac{p}{n}\widehat{\sigma^2} \tag{5.26}$$

Dans ce critère, $\widehat{\sigma^2}$ est une estimation de la variance de l'erreur par un modèle de faible biais. La qualité de cette estimation est un préalable fondamental pour que la sélection du bon modèle s'opère. Mais il s'agit d'un problème de poules et d'œufs...Si le modèle le plus général n'est pas le vrai modèle (ou du moins si son biais n'est pas limité), l'estimation correcte de la variance nécessite de connaître la complexité optimale qui, à son tour, ne peut être identifiée par AIC sans connaître la variance...

Plusieurs justifications théoriques d'AIC existent. Nous présentons ci-après celle de Hastie qui définit le terme d'optimisme de l'équation 5.25 comme l'écart entre une approximation de l'EPE et une espérance (comparable) de l'erreur d'apprentissage. En moyenne, lorsque les sorties $Y = (y_1, \dots y_i \dots y_n)^T$ varient, l'erreur d'apprentissage moyenne est :

$$E_Y\left[\mathtt{err}(\widehat{h})\right] = E_Y\left[\frac{1}{n}\sum_{i=1}^{n} e(y_i, \widehat{h}(x_i))\right] \tag{5.27}$$

Pour approcher raisonnablement l'EPE, on peut évaluer la capacité de généralisation d'un prédicteur en remplaçant les sorties utilisées pour l'apprentissage, par de nouvelles sorties notées Y^{new}. L'écart moyen de prédiction pour la i^{ieme} donnée s'exprime alors en $E_{Y^{new}}\left[e(y_i^{new}, \widehat{h}(x_i))\right]$. Ce qui nous conduit à comparer l'espérance donnée par l'équation 5.27 avec :

$$\mathrm{Err}_{in} = E_Y\left[\frac{1}{n}\sum_{i=1}^{n} E_{Y^{new}}\left[e(y_i^{new}, \widehat{h}(x_i))\right]\right] \tag{5.28}$$

Le terme d'optimisme est alors défini par :

$$\widehat{\mathtt{op}_p} = \mathrm{Err}_{in} - E_Y\left[\mathtt{err}(\widehat{h})\right] \tag{5.29}$$

On montre pour la perte quadratique que le terme d'optimisme ainsi défini revient à :

$$\widehat{\mathtt{op}_p} = \frac{2}{n}\sum_{i=1}^{n} Cov(\widehat{y_i}, y_i) \tag{5.30}$$

Ainsi, l'optimisme de l'erreur d'apprentissage dépend de combien une prédiction $\widehat{y_i}$ est dépendante de la sortie y_i (ou, autrement dit, de l'influence de la sortie y_i sur sa propre prédiction $\widehat{y_i}$). Il devient clair alors que l'optimisme est d'autant plus fort que le modèle est complexe puisqu'il possède alors suffisamment de degrés de libertés pour s'ajuster à chaque sortie y_i. Dans le cas inverse, un modèle plus simple est trop rigide pour subir l'influence de chaque sortie y_i.

Il reste à montrer qu'avec un modèle de régression linéaire simple et des bruits iid additifs gaussiens centrés de variance σ^2 sur les sorties, on a :

$$\sum_{i=1}^{n} Cov(\widehat{y_i}, y_i) = Trace(H)\sigma^2 = p\sigma^2 \tag{5.31}$$

Ce qui justifie entièrement le critère de l'équation (5.26).

5.3.2 Autres Critères

Il existe de nombreux autres critères de sélection de modèles issus de différentes développements théoriques (ou approximations). Une version affinée de AIC qui est dédiée au cas gaussien et qui est adaptée aux petits échantillons est connue sous le nom d'AIC corrigée ou AIC_c :

$$AIC_{c_p} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 + \frac{n+p}{n-p-2}\widehat{\sigma^2} \tag{5.32}$$

Une démarche Bayésienne conduit quant à elle au critère BIC (*Bayesian Information Criterion*) qui cherche asymptotiquement le modèle associé à la plus grande probabilité a posteriori. Il revient dans notre cadre à :

$$BIC_p = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 + log(n)\frac{p}{n}\widehat{\sigma^2} \tag{5.33}$$

Dès que n est suffisamment grand, BIC tend à pénaliser les modèles complexes plus fortement que AIC. On peut montrer théoriquement que la probabilité de choisir, via BIC, le bon modèle tend vers 1 avec n tendant vers l'infini. Ce n'est pas vrai pour AIC qui tend plutôt à choisir des modèles trop complexes. Néanmoins, pour de petites tailles d'échantillon, BIC risque de se limiter à des modèles trop simples. En pratique, une comparaison empirique entre les différentes pénalisations disponibles est à opérer avant le choix du meilleur mécanisme de contrôle de la complexité pour un problème donné.