Bagging et Boosting

Considérons un ensemble de données $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, avec pour tout $i \in \{1, \ldots, N\}$, $\mathbf{x}_i = (x_{i,j})_{1 \leq j \leq D} \in \mathbb{R}^D$ un vecteur de valeurs, et un ensemble de valeurs cibles $\mathbf{t} = (t_1, \ldots, t_N) \in \mathbb{R}^N$. Rappelons que le problème de régression est le suivant : étant donné un nouveau vecteur d'entrée x, inférer quelle est la valeur cible t correspondante la plus probable.

Exercice 1 (Décomposition Biais-Variance)**.** On considère dans cet exercice un algorithme d'apprentissage qui renvoie une fonction $y(\mathbf{x})$ choisissant une valeur cible t pour chaque entrée $\mathbf{x}$. Par ailleurs, on suppose qu'il existe une distribution de probabilité de densité $p(\mathbf{x}, t)$ sur l'ensemble $\mathbb{R}^{D+1}$ des paires composées des vecteurs d'entrée et de la cible correspondante.

1. Exprimer l'erreur quadratique moyenne (sur l'espace $\mathbb{R}^{D+1}$) en terme d'intégrales.
2. Montrer que cette erreur est la somme de deux termes, l'un fonction de y, l'autre non. En déduire l'expression de $y(\mathbf{x})$ minimisant l'erreur moyenne quadratique. Dans la suite, on notera $h(\mathbf{x})$ cette fonction *idéale*.
3. En réalité, on ne connaît pas la densité p. Supposons cependant qu'on possède un nombre important d'ensembles de données $\mathcal{D}$ de taille N, tirés aléatoirement en suivant la densité $p(\mathbf{x}, t)$. Pour chaque ensemble $\mathcal{D}$, l'algorithme d'apprentissage renvoie donc une fonction de prédiction $y(\mathbf{x}; \mathcal{D})$ (dépendante de $\mathcal{D}$). On s'intéresse cette fois-ci à l'erreur quadratique (moyennée sur les ensembles $\mathcal{D}$), définie comme l'espérance (prise sur les ensembles $\mathcal{D}$) de la fonction quadratique $\big(y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\big)^2$. Exprimer cette erreur comme la somme de deux termes : reconnaître un terme de biais élevé au carré, et un terme de variance.
4. En déduire l'erreur quadratique moyenne (sur l'espace $\mathbb{R}^{D+1}$ et sur les ensembles de données $\mathcal{D}$) entre la fonction $y(\mathbf{x}; \mathcal{D})$ et la fonction idéale.

Exercice 2 (Bagging)**.** Afin d'améliorer un algorithme d'apprentissage, comme considéré dans l'exercice précédent, on va donc l'appliquer plusieurs fois sur différents ensembles de données tirés aléatoirement et indépendamment, puis les moyenner.

1. Étudier et commenter l'exemple fourni dans la figure jointe. Lorsqu'on moyenne ainsi les fonctions, quel terme de l'erreur quadratique moyenne tend à s'annuler ?
2. En pratique, on a un unique ensemble de données $\mathcal{D}$. Il est donc impossible d'appliquer une telle idée a priori. Une approche possible est d'introduire de la variabilité dans l'ensemble $\mathcal{D}$ considéré par des méthodes de *bootstrap* : on génère à partir de $\mathcal{D}$, M ensembles de taille N, en tirant uniformément avec remise des points de l'ensemble $\mathcal{D}$. Ainsi, certains points sont dupliqués dans les ensembles et d'autres n'apparaissent pas. Quelle est l'erreur quadratique commise en moyennant les fonctions $y_m(\mathbf{x})$ ($m \in \{1, \ldots, M\}$) obtenues en appliquant indépendamment l'algorithme d'apprentissage sur chaque ensemble de bootstrap. Notons E_{Bagg} cette erreur (rappelons que *bagging* vient de la contraction de *bootstrap* et d'*aggregating*).
3. Afin d'évaluer cette erreur, la comparer à l'erreur moyenne E_{Moy} définie par

$$E_{\text{Moy}} = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{i=1}^{N} \big(y_m(\mathbf{x}_i) - t_i\big)^2 .$$

4. Sous quelles conditions sur les séries statistiques $(y_m(\mathbf{x}_i) - t_i)_{1\leq i\leq N}$, a-t-on $E_{\text{Bagg}} = \frac{1}{M}E_{\text{Moy}}$? Ces conditions sont-elles réalistes ?
5. Considérer un algorithme de bagging dans lequel on pondère de manière non uniforme les poids de chaque ensemble de bootstrap. On considère ainsi que la prédiction prend la forme $y(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})$. Donner une condition nécessaire et suffisante sur $(\alpha_m)_{1\leq m\leq M}$ pour que

$$\forall \mathbf{x} \quad \min_m y_m(\mathbf{x}) \leq y(\mathbf{x}) \leq \max_m y_m(\mathbf{x})\,.$$

Exercice 3 (Boosting)**.** Afin d'améliorer les performances de l'algorithme de *bagging*, on introduit des méthodes de *boosting*. Plutôt que de considérer les M ensembles de bootstrap de manière parallèle, on les considère de manière séquentielle. On rappelle ci-dessous le fonctionnement de l'algorithme AdaBoost. On se place dans le cas d'un problème de classification à deux classes, où $t_n \in \{-1, 1\}$. On suppose qu'on possède un algorithme d'apprentissage $\mathcal{A}$ qui fournit une fonction $y(\mathbf{x})$ à valeur dans $\{-1, 1\}$ (régression logistique, arbres de décision...).

1. Initialiser les coefficients $\{w_n\}$ avec $w_n^{(1)} = 1/N$ pour tout $n \in \{1, \ldots, N\}$.
2. Pour $m = 1, \ldots, M$:
   (a) Apprendre une fonction $y_m(\mathbf{x})$ minimisant l'erreur $J_m = \sum_{n=1}^{N} w_n^{(m)} \chi(y_m(\mathbf{x}_n) \neq t_n)\,.$
   (b) Évaluer les quantités

$$\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} \chi(y_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

   et les utiliser pour évaluer $\alpha_m = \ln\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right)$.
   (c) Mettre à jour les coefficients $\{w_n\}$: $w_n^{(m+1)} = w_n^{(m)} \exp(\alpha_m \chi(y_m(\mathbf{x}_n) \neq t_n))\,.$
3. Faire des prédictions en utilisant le modèle final donné par

$$Y_M(\mathbf{x}) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})\right)\,.$$

1. On considère la fonction d'erreur exponentielle $E = \sum_{n=1}^{N} \exp(-t_n f_m(\mathbf{x}_n))$ avec $f_m(\mathbf{x})$ un classificateur défini comme une combinaison linéaire des $y_\ell(\mathbf{x})$ par

$$f_m(\mathbf{x}) = \frac{1}{2}\sum_{\ell=1}^{m} \alpha_\ell y_\ell(\mathbf{x})\,.$$

   Montrer que minimiser E en fonction de $y_m(\mathbf{x})$, à $y_1(\mathbf{x}), \ldots, y_{m-1}(\mathbf{x})$ et $\alpha_1, \ldots, \alpha_{m-1}$ fixés, équivaut à minimiser J_m.
2. Montrer que minimiser E en fonction de α_m, à $y_1(\mathbf{x}), \ldots, y_m(\mathbf{x})$ et $\alpha_1, \ldots, \alpha_{m-1}$ fixés, fournit les valeurs α_m de l'algorithme AdaBoost.
3. Prouver la correction de l'étape (c) de l'algorithme AdaBoost.
4. Interpréter finalement la formule donnée pour faire des prédictions.

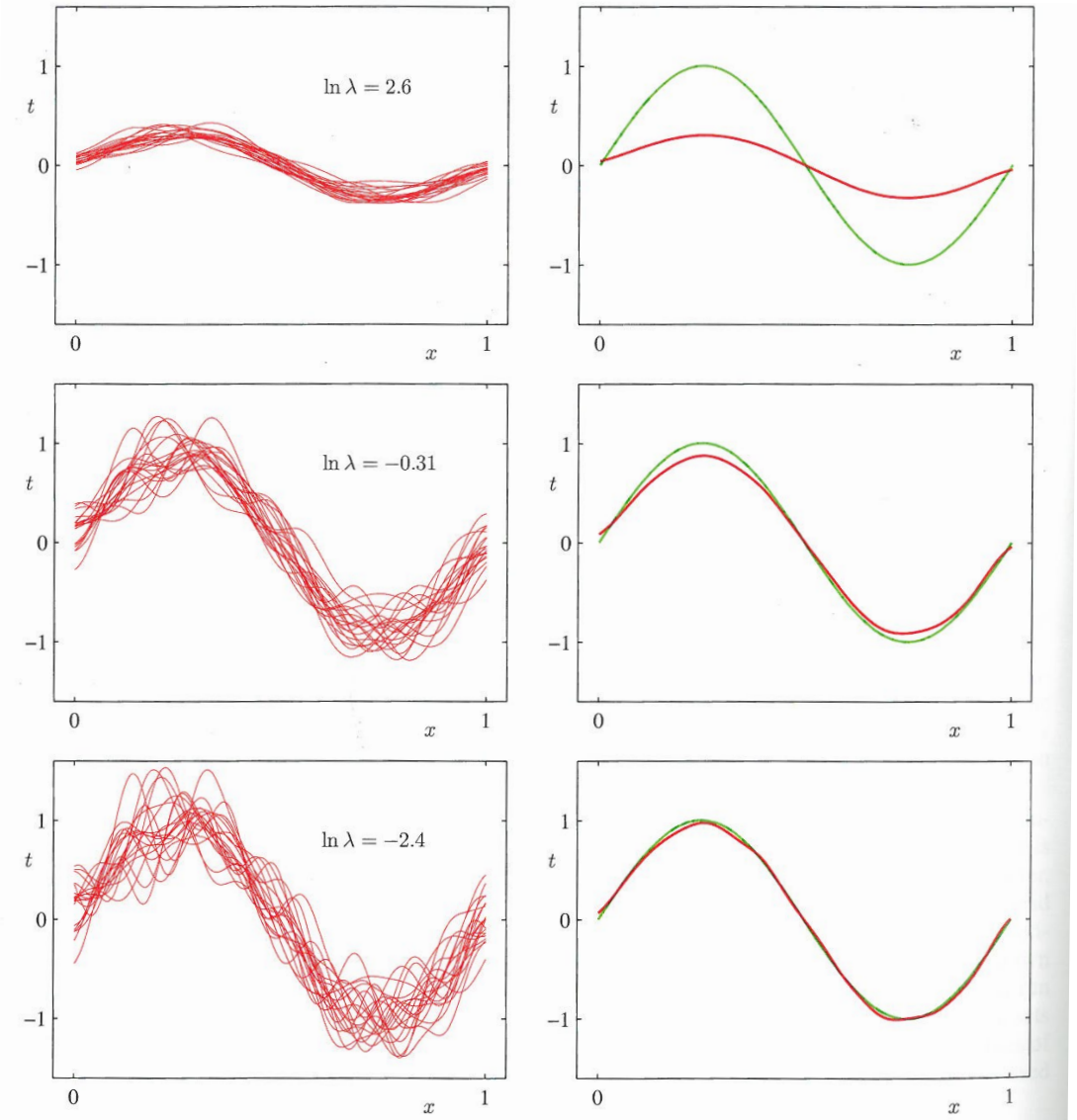

Figure 3.5 Illustration of the dependence of bias and variance on model complexity, governed by a regularization parameter λ, using the sinusoidal data set from Chapter 1. There are $L = 100$ data sets, each having $N = 25$ data points, and there are 24 Gaussian basis functions in the model so that the total number of parameters is $M = 25$ including the bias parameter. The left column shows the result of fitting the model to the data sets for various values of $\ln \lambda$ (for clarity, only 20 of the 100 fits are shown). The right column shows the corresponding average of the 100 fits (red) along with the sinusoidal function from which the data sets were generated (green).

Plot of squared bias and variance, together with their sum, corresponding to the results shown in Figure 3.5. Also shown is the average test set error for a test data set size of 1000 points. The minimum value of $(\text{bias})^2 + \text{variance}$ occurs around $\ln \lambda = -0.31$, which is close to the value that gives the minimum error on the test data.