Approximation polynomiale de courbes

Bruno Bouzy

5 octobre 2005

Introduction

Ce document présente l'approximation polynomiale de courbes. Il est un pré-requis au chapitre « réseaux de neurones » du cours « apprentissage automatique. Son but est faire comprendre ce que signifie « biais », « variance » et « généralisation ». En effet l'apprentissage automatique est très dépendant de ces concepts.

Le problème

La figure 1 représente la fonction y = sinus(2Πx) sur l'intervalle [0,1].

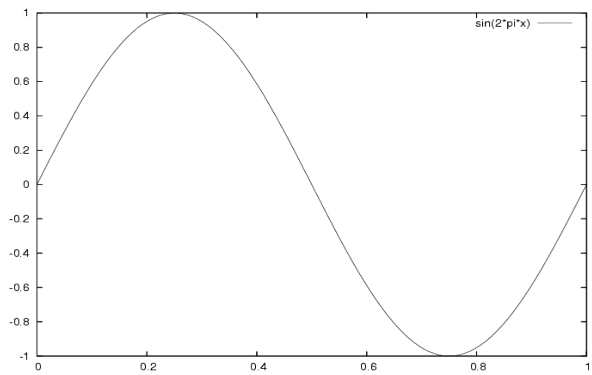

Figure 1: la fonction y = sinus(2Πx) sur l'intervalle [0,1].

Dans le cadre de l'apprentissage automatique, il faut considérer que l'on ne connaît pas cette fonction, que c'est l'objet de la recherche. Pour la trouver, nous allons disposer d'un certain nombre d'exemples et nous allons approximer cette fonction à partir des exemples que nous avons. Un exemple est un couple (xi, sinus(2Πxi)+b_i) où i est le numéro de l'exemple et b_i un bruit associé à l'exemple i. Il nous faut trouver une fonction approximante qui donne des « bonnes » valeurs de sortie pour toutes les valeurs d'entrée de l'intervalle [0,1], même si la valeur d'entrée ne correspond pas à un exemple existant. C'est cela que nous appelons connaître la fonction recherchée.

Dans le cadre de l'apprentissage automatique, la propriété qui consiste à donner une « bonne » réponse pour des valeurs d'entrée inconnues s'appelle la *généralisation*. Notre problème consiste donc à généraliser les exemples dont nous disposons.

Pour ce faire, nous supposons dans ce chapitre que la fonction approximante est un *polynôme de degré M*. Que savez-vous des polynômes de degré M ? Quelle est la propriété de base d'un polynôme de degré donné en rapport avec l'approximation de fonctions ?

Par M+1 points, passe un polynôme unique de degré M. Par exemple, par deux points passe une droite, par trois points passe un polynôme de degré 2, etc. Comment obtient-on le polynôme de degré M, connaissant les M+1 points ?

M=1, le biais

La figure 2 montre une approximation de f(x) = sinux(2Πx) connaissant les 2 exemples (0.2, f (0.2)+b1) et (0.8, f(0.8)+b2).

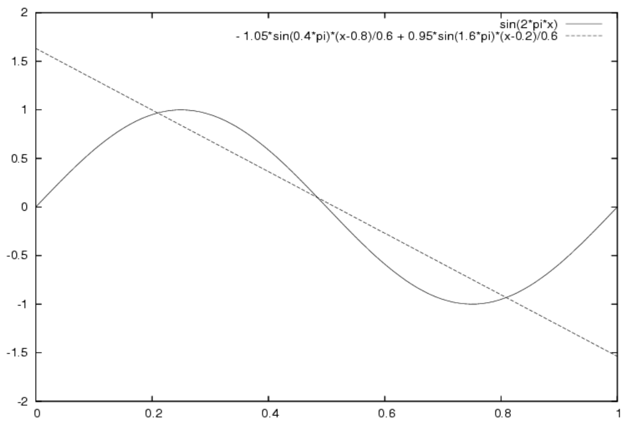

Figure 2: Approximation du sinus bruité avec 2 exemples et un polynôme de degré 1.

Si M=1, on obtient une approximation plutôt pauvre avec beaucoup de *biais.* La différence entre la fonction recherchée et son approximation est grande partout. Il faut essayer de diminuer cette différence. Nous allons donc prendre plus d'exemples et voir si cela améliore la solution actuelle.

M=3, une « bonne » approximation ?

Cette fois nous prenons 4 exemples. Il faut dont un polynôme de degré 3 pour passer par ces 4 exemples. La figure 3 montre une approximation de f(x) = sinux(2Πx) connaissant les 4 exemples pour x = 0.2, 0.4, 0.6, 0.8.

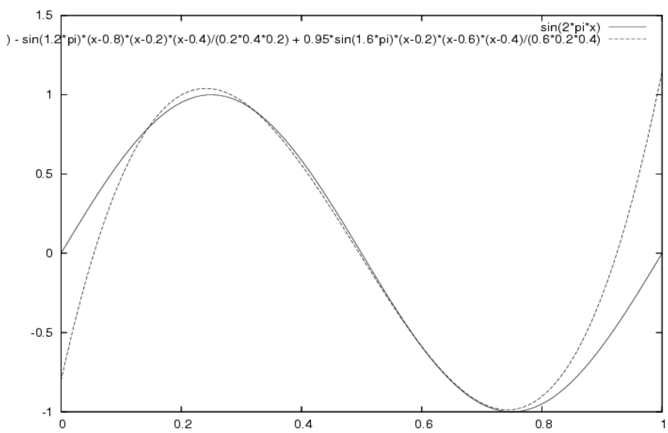

Figure 3: Approximation du sinus bruité avec 4 exemples et un polynôme de degré 3.

On observe que l’approximation est très bonne. Peut-on faire encore mieux ?

M=10, la variance

On essaie de donner 11 exemples bruités et d'approximer avec un polynôme de degré 10. La figure 4 montre une approximation de f(x) = sinux(2Πx) connaissant les 11 exemples pour x = 0.0, 0.1, 0.2, ..., 1.0.

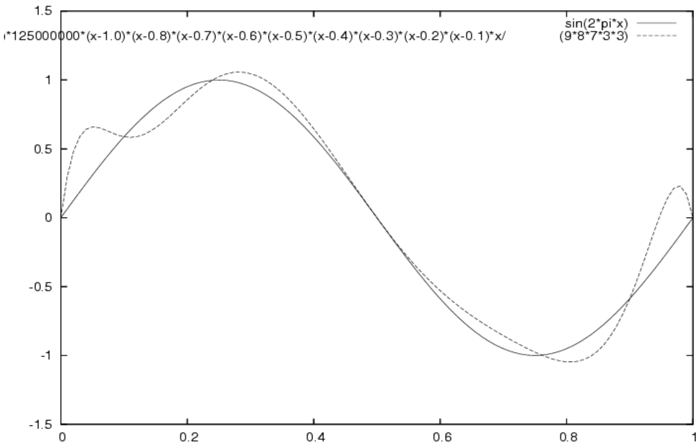

Figure 4: Approximation du sinus bruité avec 11 exemples et un polynôme de degré 10.

Si M=10, on a une approximation assez exacte sur plus de points (les 11 exemples) mais l'erreur est plus grande entre les exemples. On a beaucoup de *variance*. La fonction approximante oscille beaucoup et la généralisation est moins bonne qu'avec 4 exemples

La généralisation est meilleure si M=3 (peu de biais, peu de variance) que si M=10 (peu de biais mais beaucoup de variance) ou que si M=1 (beaucoup de biais, pas de variance).

Le point auquel la généralisation est la meilleure est déterminé par un compromis entre le biais et la variance. Il survient lorsque le nombre de degré de libertés du modèle est relativement petit comparé à la taille de l'ensemble des exemples.

On voit que les polynômes généralisent plutôt bien à condition de trouver le bon degré. Dans la suite, nous laissons de côté les polynômes et nous allons voir comment les réseaux de neurones peuvent approximer et généraliser des exemples.

Références

Christopher Bishop, "Neural Networks for Pattern Recognition", Oxford University Press, 1995