Cahier des charges

Interpolaspline

Stage Applicatif L3 MIN

Année 2019-2020

Table des matières

1 Contexte et définition du problème

1.1 Motivations

Lors d'expériences scientifiques, des données sont acquises. Cependant, ces données sont très souvent discrètes ce qui ne permet pas d'établir facilement une loi continue (fonction continue de $\mathbb{R}$ dans $\mathbb{R}^n$ suivie par ces données, n étant la dimension dans laquelle l'interpolation est faite). C'est le principe de l'interpolation.

Cependant, dans la majorité des expériences, une fonction passant par toutes les données oscillerait trop. La minimisation d'une certaine quantité liée à la distance entre les données et la fonction établie permet de créer une spline la plus proche possible des données. Dans ce cas, on parle d'approximation.

Lors des expériences, si une donnée est très éloignée des autres, on peut souhaiter l'ignorer. Cela peut-être une erreur de mesure, un changement de conditions, une grosse imprécision, etc. Ces données sont dites aberrantes, et c'est ce cas qui nous intéresse dans ce projet.

Voici quelques exemples d'utilisation de l'interpolation et de l'approximation :

- Etalonnage d'instruments de mesure, en métrologie
- Découverte de lois à partir d'expériences, dans tous les domaines de la recherche
- Identification de données incohérentes : une faute de frappe, un produit défectueux

1.2 Problème

Le problème au coeur de notre projet est la création automatique d'une spline de lissage interpolant des données, en minimisant l'impact des données aberrantes sur la spline. En effet nous savons créer une spline de lissage, mais celle-ci tient compte de toutes les données de la même manière et nécessite un paramètre de lissage explicite. Notre projet se découpe donc en deux sous-problèmes : l'estimation automatique d'un paramètre de lissage en fonction des données, et la gestion des données aberrantes.

2 Objectifs

Notre objectif est de réussir à résoudre le problème évoqué précédemment, en une mais aussi en deux dimensions. Pour cela, l'idée est d'interpoler des données en tenant compte des points aberrants, c'est à dire de trouver une ou plusieurs méthodes pour les détecter et les traiter.

Pour commencer, nous interpolerons des données à une dimension en utilisant les splines naturelles cubiques C^2. Une spline naturelle cubique C^2 est une fonction définie par morceaux par des polynômes cubiques, dont la dérivée seconde est continue et dont les dérivées secondes aux extrémités de l'intervalle de définition sont nulles.

Puis, nous implémenterons une ou plusieurs méthodes permettant de créer des splines de lissage en tenant compte des données aberrantes. Les splines de lissage sont des fonctions qui minimisent une certaine quantité liée à la différence entre la courbe et les données mesurées. L'identification des valeurs aberrantes sera implémentée pour ensuite les supprimer avant de créer la spline, ou pour leur attribuer un poids faible lors de la minimisation de la quantité.

Enfin, nous adapterons ces méthodes au cas paramétrique (données à 2 dimensions).

Contenu

Le projet sera livré sous la forme d'une archive contenant :

- Plusieurs fichiers python regroupant les différentes fonctionnalités décrites à la section 4.1 :
  - Une librairie "Splines une dimension" (nommée "$Splines1D.py$"), qui regroupe les méthodes de création des splines à une dimension
  - Une librairie "Splines Paramétriques" (nommée "$SplinesParametriques.py$"), qui regroupe les méthodes de création des splines paramétriques
  - Une librairie "Données Aberrantes" (nommée "$DonneesAberrantes.py$"), qui regroupe les méthodes de gestion des points aberrants
  - Une librairie "Automatisation" (nommée "$Automatisation.py$"), qui regroupe les méthodes de configuration automatique des paramètres des splines
  - Un fichier "Interface" (nommé "$Interface.py$"), qui contient le programme principal, qui est le seul que l'utilisateur doit lancer
- Un lisez-moi consignant les dépendances entre les programmes, et les instructions d'utilisation.
- Un rapport expliquant les diverses méthodes, avec des exemples d'utilisations illustrés.

3 Limites

Voici les limites du projet :

— Le choix automatique du paramètre de lissage ne pourra pas être idéal pour tous les cas. En effet, la quantité minimisée pour les splines de lissages est composée de deux parties : une qui assure le passage de la spline à proximité de nos données, et l'autre qui assure le lissage de la spline. Nous ne pouvons pas, dans la plupart des cas, avoir une spline qui suit bien le nuage de points et qui est bien lisse. Il faut donc faire un choix de pondération, qui pourrait ne pas être le même dans deux contextes différents avec les mêmes données.

— La robustesse de la gestion des points aberrants dépendra des données. En effet, chaque méthode possède ses points faibles, et possèdera donc au moins un jeu de données pour lequel elle ne sera pas efficace : une même donnée peut être considérée comme donnée aberrante par une méthode mais pas par une autre.

4 Description fonctionnelle

4.1 Fonctionnalités

Le programme permettra de créer une approximation de données stockées dans un fichier. L'utilisateur pourra :

— Créer la spline naturelle qui interpole exactement tous les points, en une dimension ou en paramétrique

— Créer la spline de lissage associée aux données, en une dimension ou en paramétrique

— Créer la spline de lissage, en tenant compte des points aberrants

— Choisir la quantité à minimiser dans une liste de fonctions

4.2 Entrées et sorties

4.2.1 Entrées

Lors du lancement du programme, il sera demandé à l'utilisateur d'entrer le chemin vers le fichier contenant les donnnées à interpoler. Ce fichier doit contenir 2 colonnes (séparées par un espace) et un nombre inconnu de lignes. La première colonne représente les abcisses, et la deuxième les ordonnées. Le fichier peut contenir des commentaires, précédés par un dièse et sans caractères spéciaux.

Exemple :

```
# nom : fichier_de_depart.txt
0 0
1 1
2 6 # Cette valeur est etrange
3 9
4 16
```

Ce fichier donnera en abcisses $\{0, 1, 2, 3\}$, et en ordonnées $\{0, 1, 6, 9\}$.

Puis il renseigne la dimension de ses données (une dimension ou deux dimensions) : en une dimension, l'ordre des points donnés n'a pas d'importance. En paramétrique (deux dimensions), l'ordre des points forme un paramètre de plus : pour $n+1$ points $(x_i, y_i), i \in [\![0; n]\!]$, la spline sera définie par sa valeur en un paramètre $t \in [0, 1]$ tel que si $t = f(i, n)$, alors $S(t) = (x_i, y_i)$, avec $f(i, n)$ dépendant de la répartition utilisée (par exemple, la répartition uniforme donne $f(i, n) = \frac{i}{n}$). Il faut donc que les points soient rentrés dans l'ordre dans le cas paramétrique.

L'utilisateur choisit ensuite le type de sortie qu'il souhaite, entre une spline exacte et une spline de lissage.

Enfin, l'utilisateur peut demander au programme de gérer les points aberrants, avec l'une des méthodes proposées.

4.2.2 Sorties

À la fin de l'exécution, le programme affichera un graphe représentant les données fournies par l'utilisateur, ainsi que la spline interpolant ces données (spline dépendant des choix de l'utilisateur). La discrétisation de cette spline sera également fournie en sortie, dans le fichier "$nom\_du\_fichier\_de\_depart.res$".

Ce fichier sera au même format que le fichier passé en entrée, avec en première ligne la dimension de la spline ("$1D$" ou "$2D$").

Exemple :

```
# 1D
0 0
1 1
2 4
3 9
4 16
```

5 Organisation temporelle

— Du mardi 10 au mercredi 18 décembre : organisation du projet, écriture du cahier des charges
— Mercredi 18 décembre : présentation du cahier des charges et de l'organisation du projet
— Du lundi 6 au jeudi 30 avril : réalisation du projet
— Jeudi 30 avril : soutenance du projet et rendu des codes