

Optimisation
Cours PPMD
TP du 12 novembre 2024

Le but de ce TP est de vous faire pratiquer l'algorithme du simplexe modifié et la méthode RANSAC, en comparaison à la méthode des moindres carrés.

Première étape

Vous familiariser avec les deux fonctions du module Python Fonctions_TP.py. En particulier, la fonction `creer_mesures` sera utilisée pour générer des nuages de 100 points, dont les abscisses seront linéairement distribuées entre -50 et +50, et dont les ordonnées se trouveront sur la droite d'équation $y=2x+5$, avec des bruits de mesure gaussiens et une proportion d'erreurs de mesure gaussiennes à définir.

Implémenter les autres fonctions utiles pour le TP, à savoir :

- (i) une fonction simplexe modifié, utilisant la fonction interne `linprog` de la bibliothèque `scipy.optimize` avec la méthode `'highs'`¹ ;
- (ii) une fonction implémentant la méthode RANSAC.

Deuxième étape

Utiliser les fonctions précédentes sur un jeu de données généré avec un bruit $N(0,1)$. Pour chacune des trois méthodes (moindres carrés, simplexe modifié et RANSAC), afficher les valeurs absolues des différences par rapport aux valeurs vraies des paramètres, ainsi que la moyenne et l'écart-type des résidus ; afficher également le nombre d'itérations pour les méthodes itératives ; faire un graphique avec nuage et modèle estimé et un graphique avec les résidus.

Troisième étape

Générer 100 jeux de données avec un bruit $N(0,1)$. Pour chaque jeu de données, utiliser les méthodes précédemment implémentées en première étape. Pour chaque méthode, résumer les résultats obtenus avec le nombre moyen et l'écart-type des nombres d'itérations lorsque cela fait sens, les moyennes et les écarts-types des valeurs absolues des différences entre paramètres estimés et vrais, la valeur moyenne des valeurs absolues des moyennes des résidus et, enfin, la valeur médiane des écarts-types des résidus.

Quatrième étape

Faire de même qu'à la troisième étape, mais avec un bruit $N(0,1)$ et 5 % d'erreurs $N(10,1)$.

Cinquième étape

Faire de même qu'à la troisième étape, mais avec un bruit $N(0,1)$ et 10 % d'erreurs $N(10,1)$.

Sixième étape

Faire de même qu'à la troisième étape, mais sans bruit et avec 25 % d'erreurs $N(10,1)$.

Bonus

Ajouter à l'ensemble des étapes précédentes une estimation par moindres carrés avec élimination automatique des observations fausses par un test sur les résidus normalisés avec un seuil de 3.

1 Pour High performance Simplex implementation, voir <https://docs.scipy.org/doc/scipy/reference/optimize.linprog-highs.html#optimize-linprog-highs>.