



Modélisation paramétrique

Exercice 1 : estimation par les moindres carrés ordinaires

À partir d'un ensemble de points dont la forme rappelle celle d'une ellipse, est-il possible de trouver l'ellipse qui passe « au plus près » de ces points ?

Ce genre de question se pose très souvent en analyse d'images. Il s'agit d'un problème d'*estimation*. Dans ce TP, vous allez tester plusieurs méthodes d'estimation des paramètres d'une ellipse.

Commencez par lancer le script `donnees.m`, qui affiche une ellipse tirée aléatoirement, ainsi que n points placés aléatoirement sur cette ellipse, dont chaque coordonnée est bruitée par un bruit additif gaussien.

L'équation cartésienne d'une conique (ellipse, parabole, hyperbole) est une équation polynomiale de degré 2 :

$$\alpha x^2 + \beta xy + \gamma y^2 + \delta x + \epsilon y + \phi = 0 \quad (1)$$

Les six paramètres $(\alpha, \beta, \gamma, \delta, \epsilon, \phi)$ ne sont pas indépendants, puisque la conique ne change pas si ces six paramètres sont multipliés par un même facteur. Cela signifie qu'une conique possède cinq « degrés de liberté ».

Les équations (1) forment un système linéaire *homogène* $AB = 0$, où $B = [\alpha, \beta, \gamma, \delta, \epsilon, \phi]^T$ et A est une matrice de taille $n \times 6$. Pour éviter la solution « triviale » $\hat{B} = 0$, on peut imposer une contrainte linéaire sur les paramètres, par exemple $\alpha + \gamma = 1$ (comme une ellipse vérifie $\beta^2 - 4\alpha\gamma < 0$, il est impossible que $\alpha + \gamma = 0$). Cette contrainte ajoute une équation linéaire *non homogène* au système des équations (1), qui n'admet donc plus la solution triviale $\hat{B} = 0$.

Complétez le script `exercice_1.m` par l'écriture de la fonction `MCO`, qui estime X par résolution approchée du nouveau système, au sens des moindres carrés ordinaires.

Remarque : on utilisera la fonction `pinv` pour la pseudo-inverse.

Nous vous fournissons le script `maximum_vraisemblance.m` correspondant à l'estimation des paramètres de l'ellipse par le maximum de vraisemblance (cf TP1 de Statistiques du premier semestre). Vous constatez que l'estimation est bien plus fiable par les moindres carrés ordinaires que par le maximum de vraisemblance.

Vous pouvez reformuler le problème en ne considérant que cinq paramètres en éliminant, soit le α , soit le γ et vérifier que la solution est la même à un facteur multiplicatif près.

Exercice 2 : estimation d'une transformation sur des images

L'objectif est de reconstruire des images préalablement modifiées par une transformation non linéaire bruitée.



Image originale

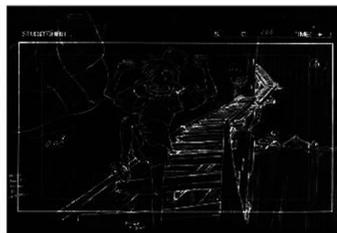


Image modifiée

Soit I une image de taille $n \times m$. Une transformation non linéaire notée T est réalisée sur chaque niveau de gris I_{ij} du pixel (i, j) de l'image I . L'image modifiée après transformation est notée J , dont les éléments sont définis par :

$$\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, m\}, \quad J_{ij} = T(\alpha, \beta, I_{ij}) = e^{-\alpha I_{ij} + \beta} \quad (2)$$

où (α, β) sont des paramètres réels.

Problème aux moindres carrés ordinaires

On dispose d'une partie de l'image modifiée par la transformation T avec un bruit additif aléatoire et de la partie de l'image originale correspondante.

1. Écrivez la formulation matricielle du problème aux moindres carrés ordinaires (MCO) permettant d'estimer les paramètres (α, β) de la fonction T définie par l'équation (2).
2. Résolvez le problème aux moindres carrés ordinaires à partir des données `Data` et `DataMod` issus respectivement des images originales et modifiées (`load SG1.mat`).
3. À partir des paramètres estimés et de l'image entière modifiée `ImMod`, reconstruisez l'image originale (issue de `ImSG1.mat`).
4. Estimez l'erreur aux moindres carrés (RMSE) entre l'image reconstruite et l'image originale.
5. Essayez avec les autres jeux de données.

Problème aux moindres carrés totaux

On propose de résoudre ce même problème en utilisant les moindres carrés totaux et la décomposition en valeurs singulières (SVD).

1. Écrivez le problème aux moindres carrés totaux.
2. Résolvez le problème et reconstruisez l'image originale.

Remarque : on utilisera la fonction `svd`.

3. Estimez l'erreur aux moindres carrés (RMSE) entre l'image reconstruite et l'image originale. Comparez vos résultats avec les MCO.
4. Imaginez des applications à ce bruitage d'images.