TP3 – Estimation robuste

Influence des données aberrantes sur l'estimation en moindres carrés

Il a été vu lors du TP2 comment estimer les paramètres de la droite de régression D_{YX} de Y en X, et ceux de la droite de régression $D_\perp$ en distance orthogonale, à partir d'un nuage de points. Or, si parmi ces points, certains constituent des *données aberrantes*, cela risque fort d'altérer l'estimation.

Lancez le script `exercice_0.m`, qui permet de visualiser l'influence de la présence de données aberrantes sur l'erreur angulaire moyenne (EAM) commise sur la direction de chaque droite de régression. Bien sûr, ces erreurs augmentent avec la proportion de données aberrantes, mais vous pouvez aussi faire les observations suivantes :

- En l'absence de données aberrantes, la droite de régression D_{YX} est quasiment aussi précise, en moyenne, que la droite de régression $D_\perp$. Au-delà de 5% de données aberrantes, l'écart se creuse.
- Si l'addition d'un bruit gaussien et le tirage de valeurs aberrantes (selon une loi uniforme) s'appliquent non seulement aux ordonnées, mais également aux abscisses des points du nuage, alors l'estimation de la droite D_{YX} se dégrade nettement. Au contraire, l'écart angulaire moyen de $D_\perp$ n'augmente que très peu.

Le but de ce TP est de vous montrer deux façons de limiter l'influence des données aberrantes sur l'estimation des droites de régression, en remettant en question l'estimation en moindres carrés vue dans le TP2.

Première parade aux données aberrantes : l'algorithme RANSAC

RANSAC (abréviation de *RANdom SAmple Consensus*) est un algorithme itératif d'*estimation robuste*, publié par Fischler et Bolles en 1981, qui consiste à effectuer une *partition des données* entre données aberrantes (*outliers*) et données conformes au modèle (*inliers*). Une caractéristique de cet algorithme est qu'il est non déterministe : le résultat n'est garanti qu'avec une certaine probabilité, qui croît avec le nombre d'itérations.

L'exemple qui illustre le mieux l'algorithme RANSAC est l'estimation robuste d'une droite de régression. Le principe de RANSAC consiste à tirer aléatoirement un sous-ensemble de données de cardinal égal au nombre minimal de données permettant d'estimer le modèle (nombre égal à 2 dans le cas d'une droite de régression). Ces données sont considérées comme des données conformes au modèle (cela reste à vérifier), puis la séquence suivante est répétée en boucle :

1. Les paramètres du modèle sont estimés à partir de ce sous-ensemble de données conformes.
2. Toutes les autres données sont testées relativement au modèle estimé, afin de détecter les données conformes (un point est jugé conforme si sa distance à la droite de régression est inférieure à un seuil).
3. Le modèle estimé est accepté si la proportion de données conformes est supérieure à un seuil.
4. Si le modèle est accepté, il est réestimé à partir de l'ensemble des données conformes.

Le modèle retenu est celui qui minimise le critère, égal à la *moyenne* des carrés des résidus des données conformes. **Attention :** comme le nombre de données conformes varie à chaque tour de boucle, il serait faux d'utiliser comme critère la *somme* des carrés des résidus.

Exercice 1 : estimation robuste par l'algorithme RANSAC

Complétez le script `exercice_1.m`, qui applique l'algorithme RANSAC à l'estimation de la droite de régression $D_\perp$, lorsque l'addition d'un bruit gaussien et le tirage de valeurs aberrantes s'appliquent à la fois aux abscisses et aux ordonnées des points du nuage. Si le bénéfice de cette nouvelle méthode d'estimation semble incontestable en termes de robustesse aux données aberrantes, elle pèche manifestement par la nécessité de régler correctement ses paramètres `k_max`, `seuil_distance` et `seuil_proportion`, et surtout par sa lenteur !

Deuxième parade aux données aberrantes : autre critère à optimiser

Dans le TP2, l'estimation des paramètres (a, b) de la droite de régression D_{YX} et celle des paramètres (θ, ρ) de la droite de régression $D_\perp$ ont été effectuées en résolvant des problèmes d'optimisation du type suivant :

$$\min_{\mathbf{p}=[p_1,\ldots,p_m]} \sum_{i=1}^{n} r_{\mathbf{p}}(P_i)^2 \tag{1}$$

où $\mathbf{p} = [p_1, \ldots, p_m]$ désigne le vecteur des paramètres à estimer (dans le cas de la droite $D_\perp$: $\mathbf{p} = [\theta, \rho]$). Or, les données aberrantes dégradent la précision de ces estimations à cause des résidus $r_{\mathbf{p}}(P_i)$ élevés, puisque ces points seront probablement éloignés de la droite de régression. Cela est encore amplifié par le fait que les résidus sont élevés au carré dans (1). Une deuxième façon de limiter l'influence des données aberrantes consiste donc à conserver toutes les données, mais à utiliser des *moindres carrés pondérés*, c'est-à-dire un nouveau critère $\mathcal{W}(\mathbf{p}) = \sum_{i=1}^{n} w_i\, r_{\mathbf{p}}(P_i)^2$, en faisant en sorte que le poids $w_i \geqslant 0$ du point P_i soit d'autant plus faible que le résidu $r_{\mathbf{p}}(P_i)$ est plus élevé. Malheureusement, il semble que cette idée soit impossible à mettre en œuvre, car on ne sait pas quels points constituent des données aberrantes (le principe de cette approche est de ne pas effectuer de partition des données, contrairement à l'algorithme RANSAC). Néanmoins, l'optimalité de ce critère s'écrit :

$$\nabla\mathcal{W}(\mathbf{p}) = 0 \quad \Longleftrightarrow \quad \sum_{i=1}^{n} w_i\, r_{\mathbf{p}}(P_i)\, \frac{\partial r_{\mathbf{p}}}{\partial p_j}(P_i) = 0\ , \quad \forall\, j \in [1, m] \tag{2}$$

D'autre part, toute fonction ϕ d'une variable réelle à valeurs dans $\mathbb{R}^+$, dérivable, paire, croissante sur $\mathbb{R}^+$, permet de définir un critère $\mathcal{H}(\mathbf{p}) = \sum_{i=1}^{n} \phi(r_{\mathbf{p}}(P_i))$ tout aussi valide que le critère utilisé dans (1), c'est-à-dire tout aussi valide que la somme des carrés des résidus. L'optimalité du critère $\mathcal{H}(\mathbf{p})$ s'écrit :

$$\nabla\mathcal{H}(\mathbf{p}) = 0 \quad \Longleftrightarrow \quad \sum_{i=1}^{n} \phi'(r_{\mathbf{p}}(P_i))\, \frac{\partial r_{\mathbf{p}}}{\partial p_j}(P_i) = 0\ , \quad \forall\, j \in [1, m] \tag{3}$$

Par identification des équations (2) et (3), on trouve $w_i = \frac{\phi'(r_{\mathbf{p}}(P_i))}{r_{\mathbf{p}}(P_i)}$. Il existe de nombreuses fonctions ϕ nulles en 0, telles que les poids w_i décroissent lorsque $r_{\mathbf{p}}(P_i)$ croît, comme par exemple $\phi_1(x) = \sqrt{x^2 + \alpha^2} - \sqrt{\alpha^2}$ ou $\phi_2(x) = \ln(x^2 + \beta^2) - \ln(\beta^2)$, où le rôle des paramètres α et β est de rendre ces fonctions dérivables en $x = 0$. Dans la pratique, il n'est pas question de résoudre les équations (3), qui n'admettent pas de solution analytique pour des fonctions telles que ϕ_1 ou ϕ_2. En revanche, il est facile de minimiser le critère $\mathcal{H}(\mathbf{p})$ par le maximum de vraisemblance, en résolvant un des deux problèmes suivants :

$$\min_{\mathbf{p}=[p_1,\ldots,p_m]} \sum_{i=1}^{n} \sqrt{r_{\mathbf{p}}(P_i)^2 + \alpha^2} \tag{4}$$

$$\min_{\mathbf{p}=[p_1,\ldots,p_m]} \sum_{i=1}^{n} \ln(r_{\mathbf{p}}(P_i)^2 + \beta^2) \tag{5}$$

La dérivabilité du critère n'étant plus requise, on peut choisir $\alpha = 0$ et $\beta = 0$. Lorsque $\alpha = 0$, le critère du problème (4) s'écrit $\sum_{i=1}^{n} |r_{\mathbf{p}}(P_i)|$, qui est la norme L_1 du vecteur des résidus. Ce critère est très souvent utilisé.

Exercice 2 : estimation robuste par résolution des problèmes (4) et (5)

Le script `estimation_robuste.m` effectue l'estimation de la droite de régression $D_\perp$ par le maximum de vraisemblance, lorsque l'addition d'un bruit gaussien et le tirage de valeurs aberrantes s'appliquent aux abscisses et aux ordonnées : d'abord en moindres carrés, puis en résolvant les problèmes (4) et (5), avec $\alpha = \beta = 1$.

Faites une copie des scripts `exercice_2.m` et `donnees_aberrantes.m` écrits lors du TP1. En vous inspirant du script `estimation_robuste.m`, modifiez le script `exercice_2.m` de manière à effectuer l'estimation *robuste*, par résolution des problèmes (4) et (5), du centre et du rayon du cercle passant « au plus près » des points affichés par le script `donnees_aberrantes.m`. Quelle estimation vous semble-t-elle la plus robuste ?