Application de méthodes de statistique robuste à l'analyse de mesures de bruit de roulement

Marie-Paule Ehrhart

Application de méthodes de statistique robuste à l'analyse de mesures de bruit de roulement

Rapport de stage

Laboratoire Régional des Ponts et Chaussées de Strasbourg

Groupes « Acoustique » et « Méthodes Physiques »
Sous la direction de Guillaume Dutilleux et Pierre Charbonnier

Marie-Paule Ehrhart
M1 Statistique
Année 2010-2011

Résumé

Les mesures de bruit de roulement sont effectuées pour deux catégories de véhicules, véhicules légers et poids lourds. Il s'agit de déterminer, pour chaque catégorie, les paramètres d'une droite représentant le niveau de bruit (en dB) en fonction de la variable log10(vitesse) (vitesse en km/h). Durant le stage, plusieurs méthodes d'estimation robuste ont été étudiées afin de trouver une solution alternative à l'élimination manuelle des points aberrants par l'opérateur chargé des mesures : les M-estimateurs qui calculent le résidu sur le niveau sonore seul, l'ACP (et l'ACP robuste) qui calcule le résidu sur les deux axes. Face à l'observation que les valeurs aberrantes d'une classe semblent être des valeurs valides de l'autre classe (et réciproquement), une nouvelle classification des deux catégories par algorithme EM est effectuée. Afin d'enlever l'erreur venant des valeurs aberrantes, le *bootstrap* non paramétrique a aussi été essayé. Celui-ci n'a rien apporté de neuf quant aux estimations des paramètres, mais il fournit des intervalles de confiance qu'il peut être intéressant d'exploiter. Une première tentative a été faite de sélectionner une méthode d'estimation parmi toutes celles testées, en calculant le niveau sonore équivalent. Toutefois, l'approche développée n'est pas suffisamment discriminante.

Mots-clés

Mesure de bruit de roulement, valeurs aberrantes, régression, M-estimation, ACP robuste, algorithme EM, niveau de pression sonore équivalent

Remerciements

Je tiens à remercier M. Georges Kuntz, directeur du Laboratoire Régional de Strasbourg, de m'avoir accueillie dans son laboratoire, et de m'avoir ainsi permis d'effectuer mon stage.

Merci à Guillaume Dutilleux et Pierre Charbonnier, mes maîtres de stage, qui m'ont guidée et conseillée tout au long de ce stage et qui m'ont ainsi permis de mener à bien le projet qui m'a été confié.

Je remercie enfin les membres de l'équipe « Acoustique » et « Méthodes Physiques » qui m'ont accueillie dans leurs groupes et m'ont permis de découvrir les différents travaux effectués dans ce laboratoire. En particulier, je remercie Loïc Toussaint et David Ecotière pour leurs conseils concernant le stage.

Table des matières

Chapitre 1

Présentation et contexte du stage

1.1 Organisme d'accueil [1]

Mon stage s'est déroulé au Laboratoire Régional des Ponts et Chaussées de Strasbourg, qui fait partie du CETE de l'Est et se trouve au 11, rue Jean Mentelin à Koenigshoffen. Il a été effectué sous la direction de MM. Guillaume Dutilleux et Pierre Charbonnier. Le stage a eu lieu du 6 juin au 13 juillet et du 1er au 19 août 2011.

1.1.1 Les CETE : Centres d'Etudes Techniques de l'Equipement

Les CETE, pour « Centre d'Etudes Techniques de l'Equipement » sont au nombre de huit répartis dans toute la France. Ces organismes sont des bureaux d'études d'ingénierie publique et font partie du Ministère de l'Ecologie, du Développement Durable, des Transports et du Logement. Le CETE de l'Est répartit ses compétences sur trois sites : direction et département d'études à Metz, et deux laboratoires régionaux, à Nancy et Strasbourg.

La zone d'action du CETE de l'Est couvre les régions Alsace, Lorraine et Champagne-Ardenne, soit au total 10 départements. Comme le CETE est un organisme public, il agit pour le compte de l'Etat, mais intervient également pour les communes, les collectivités territoriales, ou pour d'autres organismes publics ou privés.

Plusieurs missions sont confiées au CETE, concernant notamment l'aménagement du territoire. Son travail consiste à faire des études, de la recherche, de l'expertise, du conseil ou encore du contrôle (sur site ou en laboratoire) dans de nombreux domaines. Parmi eux, on trouve l'aménagement, l'urbanisme, la construction, la gestion des réseaux routiers, la géotechnique, les terrassements, les chaussées, les transports, les infrastructures, les ouvrages d'art, la sécurité routière, l'environnement.

1.1.2 Le Laboratoire Régional de Strasbourg

Le laboratoire de Strasbourg existe depuis 1959, mais à ce moment-là, il n'est qu'une annexe du Parc Départemental de l'Equipement. Durant les années 1960, il devient Laboratoire Régional des Ponts et Chaussées et vient compléter la liste des laboratoires régionaux.

Aujourd'hui, le laboratoire de Strasbourg est dirigé par M. Georges Kuntz, et est constitué, en plus des services généraux, de cinq groupes techniques : le groupe « Géotechnique, Terrassement, Chaussées », le groupe « Ouvrages d'art », le groupe « Construction », le groupe

1. Source utilisée pour cette partie : `http://www.cete-est.developpement-durable.gouv.fr`

« Acoustique » et le groupe « Méthodes Physiques ». Environ 80 personnes sont présentes dans ce laboratoire. Un organigramme est présenté sur la figure 1.1.

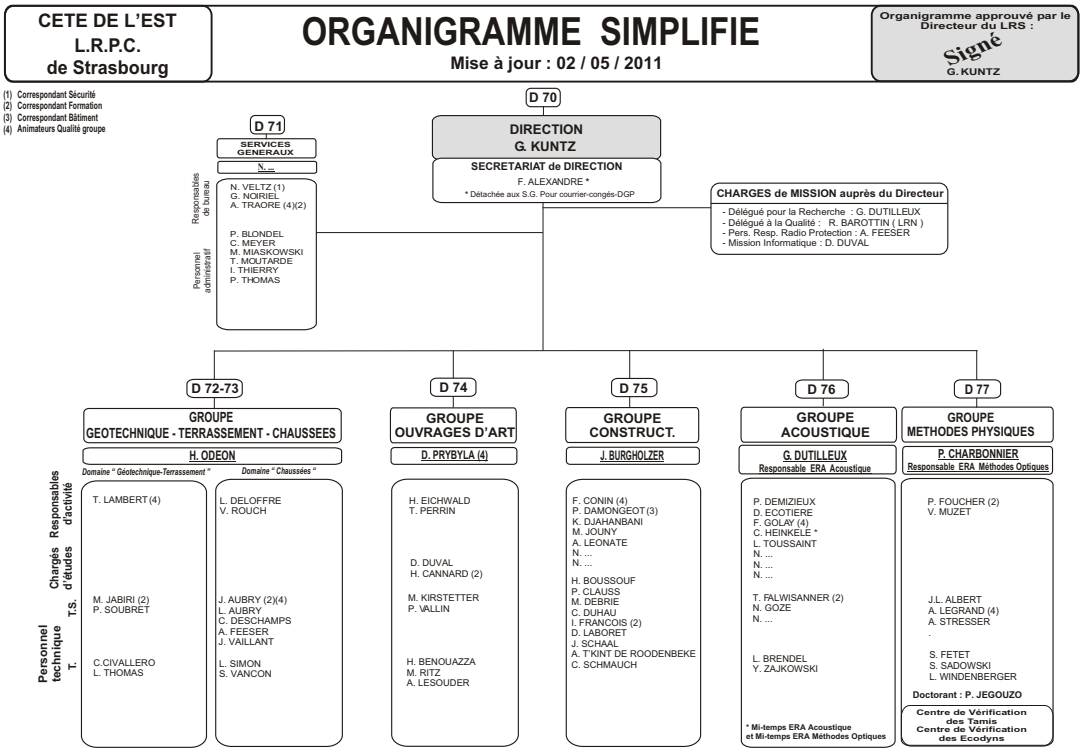


FIGURE 1.1 – Organigramme - Laboratoire Régional de Strasbourg

Chacun de ces groupes a une mission spécifique au sein du laboratoire mais peut être également relié à d'autres équipes ayant les mêmes missions dans d'autres CETE en France dans le cadre de Pôles de Compétences et d'Innovation (PCI).

Par exemple, le groupe « Ouvrages d'Art » s'occupe de tous les ouvrages que l'on peut trouver sur une route, comme les ponts, ou les tunnels. Certaines de leurs missions sont la gestion et le suivi de tout ce patrimoine et la recherche concernant les matériaux utilisés, mais aussi la conception et la réalisation d'ouvrages.

Nous pouvons également citer le groupe « Construction », qui a quatre missions principales : la qualité de la construction, la performance énergétique des bâtiments, leur qualité environnementale et la gestion de patrimoine immobilier.

Dans le laboratoire, la recherche concerne également le développement durable (par exemple : des matériaux moins énergivores).

1.1.3 le Groupe Acoustique

Le groupe Acoustique est composé d'une dizaine de personnes (hors vacataires) et est dirigé par M. Guillaume Dutilleux. Par des mesures sur le terrain, des études, des expertises, l'équipe contribue à lutter concrètement contre les nuisances sonores. La recherche fait aussi partie du travail effectué par le groupe, dans le cadre de l'Equipe Recherche Associée « Acoustique »

associée à l'IFSTTAR (Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux). Elle représente 30% de son activité.

Des clients, comme ceux présentés plus haut, peuvent ainsi contacter le groupe Acoustique pour des projets d'infrastructures. Par exemple, lors de constructions de nouvelles routes, ou de modifications de routes, une étude peut être effectuée afin de déterminer quelle installation anti-bruit sera la plus efficace (cela peut être les dimensions d'un mur anti-bruit par exemple).

Comme exemples de mission, nous avons la cartographie du bruit des grandes villes et infrastructures, des études concernant l'acoustique des bâtiments, ou encore des actions concernant le bruit industriel et de voisinage.

Les campagnes de mesure de bruit de roulement, qui font l'objet du stage, servent à classifier les différents revêtements routiers selon leurs performances acoustiques et le type des véhicules qui les empruntent.

L'équipe peut ainsi traiter des plaintes de voisinage concernant le bruit, faire des diagnostics de bruit sur des bâtiments (anciens ou neufs), ou encore contrôler la qualité des constructions. De plus, elle participe aux groupes de travail pour l'établissement de normes françaises et européennes sur le bruit.

De la recherche est faite également sur la caractérisation de l'émission sonore des véhicules, la pérennité des performances acoustiques des revêtements, la modélisation de la propagation acoustique à grande distance et l'optimisation des ouvrages de protection contre le bruit.

Le travail de cette équipe comporte donc de nombreuses facettes, même si toutes sont bien sûr liées à l'acoustique et contribuent à améliorer la qualité de vie des riverains de sources de bruit.

1.1.4 le Groupe Méthodes Physiques

Comme le groupe Acoustique, le groupe Méthodes Physiques est constitué d'une dizaine de personnes. Il est dirigé par M. Pierre Charbonnier.

La recherche occupe 60% du travail de ce groupe, dans le cadre de l'Equipe de Recherche Associée « Imagerie et Méthodes optiques » de l'IFFSTAR. Les 40% restants sont des applications servant à mettre en oeuvre et à valoriser les produits de la recherche.

Le volet recherche est orienté en particulier vers le traitement d'images et la reconnaissance de formes. Ce sont ces sujets qui amènent les chercheurs à considérer l'estimation robuste et l'algorithme EM, qui sont les principales méthodes utilisées durant le stage. Par exemple, les travaux de ce groupe sur l'estimation robuste ont notamment concerné la détection des marquages (lignes blanches) dans des séquences d'images routières. Cela a été réalisé en collaboration avec l'IFSTTAR et le LIVIC (Laboratoire sur les Interactions Véhicules-Infrastructure-Conducteurs). Dans ce cadre, nous pouvons citer [TIC07].

Les applications principales se font dans le domaine de la sécurité routière, ou encore dans l'inspection d'ouvrages d'art.

1.2 Contexte et objectifs du stage

1.2.1 Contexte

En France, il existe différents types de revêtements routiers dont la différence vient principalement du liant et des granulats utilisés. Selon les caractéristiques de ces revêtements, le bruit dû au trafic routier est plus ou moins important, toutes choses égales par ailleurs. L'une des missions du Groupe Acoustique du Laboratoire Régional de Strasbourg est de classifier les revêtements selon le niveau de bruit de roulement. Cela sert par exemple à faire des recommandations lors de constructions de routes ou lorsqu'il y a des plaintes de riverains. Il est important de disposer d'une loi donnant le niveau de bruit en fonction de la vitesse afin de pouvoir s'adapter à la vitesse limite réglementaire sur un site donné.

Pour faire cette classification, le groupe effectue des campagnes de mesures de bruit de roulement sur le terrain. Ces campagnes sont faites selon une norme, la norme AFNOR S31-119 ([31-93]).

Les campagnes qui nous intéressent sont celles qui suivent la procédure dite VI (Véhicules Isolés), c'est-à-dire qu'on effectue les mesures selon le trafic de la route en question : on enregistre tous les véhicules qui passent. Une campagne consiste d'abord à faire quelques mesures d'environnement (comme la température et la vitesse du vent). Ensuite, pour chaque passage de véhicule, un cinémomètre radar mesure sa vitesse, et un microphone relié à un enregistreur note le bruit de ce véhicule à 7,5m de l'axe de la chaussée et 1,2m de hauteur. Sur l'enregistrement, on trouve également les commentaires de l'opérateur qui fait les mesures. Celui-ci donne la catégorie du véhicule ainsi que sa vitesse, lue sur l'affichage du cinémomètre. Pour ces études, on distingue deux types de véhicules : les véhicules légers (VL) et les poids lourds (PL).

Les deux figures 1.2 et 1.3 montrent, en vue de dessus (« plan ») et en vue de profil (« travers »), un exemple d'installation permettant de faire de telles campagnes.

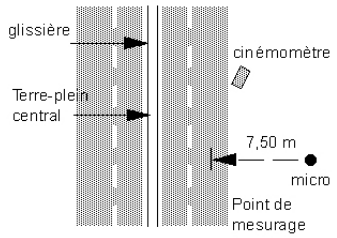


FIGURE 1.2 – Vue en plan

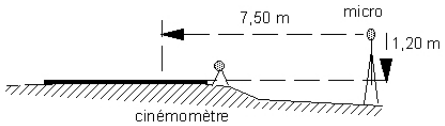


FIGURE 1.3 – Vue en travers

Avant de commencer les mesures, il faut également calibrer l'enregistreur par rapport à

une source de bruit de référence. Ceci permet de garantir que les niveaux sonores issus de deux campagnes soient comparables.

Vient ensuite le moment de l'importation des données et du dépouillement avec le logiciel `dBEuler`, développé par le LRPC sous l'environnement `Scilab`. Ce logiciel est présenté dans le manuel [Dut06]. Un opérateur écoute chaque enregistrement, examine son spectre et sa signature, le valide ou non. Il s'agit aussi de délimiter les passages au sein des enregistrements, car plusieurs passages de véhicules peuvent être effectués sur le même fichier. L'opérateur enregistre dans le logiciel les informations nécessaires à chaque véhicule comme sa catégorie et sa vitesse. `dBEuler` transforme ces fichiers audio et donne un niveau de bruit maximum pour le véhicule, noté L_{Amax} (en décibels).

Après une correction de sensibilité (due au calibrage) et de température (qui a un impact sur le bruit), on peut procéder à l'analyse statistique qui est une régression linéaire du niveau de bruit par rapport à la vitesse. En fait, un paramètre adimensionnel de vitesse est utilisé : on prend le logarithme (en base 10) de la vitesse. On fait donc la régression avec comme variable explicative :

$$X = \log_{10}\left(\frac{vitesse}{v_{ref}}\right) \tag{1.1}$$

où v_{ref} est la vitesse de référence (en général 90 km/h pour les véhicules légers et 80 km/h pour les poids lourds).

Par abus de langage, on pourra trouver pour les axes des abscisses de certains graphiques la mention « Vitesse modifiée », même si le paramètre en question n'est pas homogène à une vitesse, comme indiqué ci-dessus.

On cherche, à la fin de la campagne, une loi de la forme :

$$L_{Amax} = a \times \log_{10}\left(\frac{vitesse}{v_{ref}}\right) + b \tag{1.2}$$

où a et b sont les paramètres cherchés.

Le but de cette régression est d'obtenir un niveau de bruit (en décibels) de référence pour une vitesse donnée, par exemple 90 km/h. On lit simplement la valeur du niveau de bruit de référence sur la droite de régression, à l'abscisse voulue. Le graphique 1.4 page suivante montre comment cette valeur est lue. La droite tracée a été obtenue par régression linéaire.

Toutefois, comme tous les véhicules du trafic sont pris en compte, on a régulièrement des mesures aberrantes, par exemple une voiture dont le niveau de bruit au passage est beaucoup plus fort que la moyenne. Le problème est que, par une simple régression linéaire, l'analyse est faussée à cause de ces points. Ces valeurs aberrantes peuvent également être dues à des erreurs de mesure. Plusieurs sources d'erreur sont possibles : lors du passage du véhicule (bruit parasite, erreur de mesure, erreur de l'opérateur concernant la catégorie du véhicule ou sa vitesse) ou encore lors de l'importation des données (erreur de transcription dans le logiciel, pour la saisie par l'opérateur de la vitesse ou de la catégorie du véhicule).

`dBEuler` permet à un opérateur de supprimer manuellement des points qu'il juge aberrants, simplement en les désactivant via une interface graphique. L'objectif du stage porte sur l'automatisation de cette étape, l'élimination manuelle n'étant pas forcément objective.

Finalement, `dBEuler` produit un rapport d'analyse. Les points aberrants étant supprimés, on obtient deux valeurs : la pente et l'ordonnée à l'origine de la droite. Cette dernière constituera le niveau de bruit de référence pour la route étudiée. En effet, le niveau de bruit de

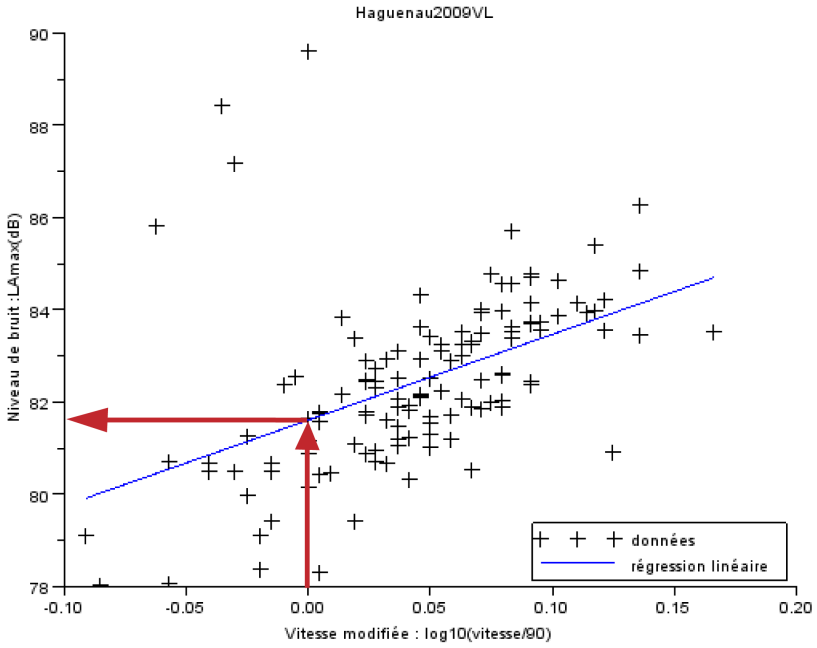


FIGURE 1.4 – Exemple de lecture du niveau de bruit de référence

référence est le bruit obtenu pour la vitesse de référence. D'après la formule 1.1 page précédente utilisée pour normaliser la vitesse, si $vitesse = v_{ref}$, alors la valeur de $\log_{10}\left(\frac{vitesse}{v_{ref}}\right)$ vaut 0. Cela justifie l'utilisation de l'ordonnée à l'origine.

1.2.2 Objectifs du stage

Dans le logiciel `dBEuler`, l'analyse statistique passe par une régression linéaire au sens des moindres carrés. Le problème de cette méthode est qu'elle est très sensible aux données aberrantes. Nous pouvons voir cela sur le graphique A.3 page 59 où une donnée aberrante change fortement la droite de régression obtenue.

Pour l'instant, dans `dBEuler`, il est possible de supprimer manuellement des données jugées aberrantes. Mais ce n'est pas une bonne solution car cela nécessite l'avis subjectif de l'opérateur pour déterminer quelles données sont aberrantes. L'objectif du stage est de trouver une méthode statistique robuste permettant de traiter ces mesures. Renaud Wintzer, stagiaire en 2010 au LRPC, a commencé à travailler sur ce sujet, comme cela est présenté dans son rapport [Win10]. J'ai donc pu récupérer quelques fichiers de code qu'il avait écrit.

Plusieurs solutions sont alors envisageables pour traiter les données aberrantes. Ce sera à des acousticiens de déterminer quelle méthode donnera les meilleurs résultats. Pour chaque jeu de données, ce qu'on cherche est la pente et la valeur à l'origine de la meilleure droite possible.

Il s'agit d'abord d'étudier les M-estimateurs en y ajoutant l'estimation de l'échelle.

Une ACP (Analyse en Composantes Principales) est également envisagée ainsi qu'une ACP robuste. La méthode du *bootstrap* est implémentée et étudiée.

Enfin, lors de précédentes campagnes, les nuages de points correspondants aux véhicules légers et aux poids lourds ont été superposés. Il a alors été remarqué que les mesures jugées aberrantes pour l'une des deux catégories de véhicules ont l'air « conforme » à l'autre catégorie, et inversement. L'un des objectifs est d'étudier la possibilité d'une classification non supervisée par un algorithme EM (ce qui signifie *Expectation-Maximization*), c'est-à-dire de déterminer la catégorie du véhicule sans que l'opérateur effectuant la mesure ne la donne. Cela permettrait d'éviter les éventuelles erreurs lors du dépouillement, ou du moins d'avertir l'opérateur lorsqu'un point « change » de catégorie à la fin de l'algorithme.

Dans ce cas également, on peut comparer les résultats obtenus de manière non-robuste et de manière robuste.

Pour implémenter toutes ces méthodes, le logiciel à utiliser est `Scilab`. En effet, comme indiqué précédemment, le logiciel `dBEuler` a été développé sous `Scilab` et le but est que ces méthodes soient intégrées à `dBEuler`. Mais les fonctions du logiciel `R` sont également utilisées pour quelques points particuliers.

Durant le stage, j'ai utilisé la version 5.3.3 de `Scilab` et la version 2.13.1 de `R`.

1.2.3 Structure du rapport

Après ce chapitre de présentation, deux chapitres traitent des principales méthodes d'estimation utilisées. Le chapitre 2 page suivante présente les M-estimateurs tandis que le chapitre 3 page 20 donne les analyses faites par ACP et ACP robuste. Ces deux méthodes sont traitées car elles permettent d'avoir une vision différente des choses : on calcule les résidus entre les points et la droite respectivement de manière verticale et perpendiculaire.

Ensuite, comme cela a été expliqué dans la section précédente, on peut faire un lien entre les points aberrants des deux catégories VL et PL. Le chapitre 4 page 29 présente l'analyse faite avec l'algorithme EM, qui est ici une autre manière de traiter les points aberrants. Il permet aussi de traiter ensemble les données VL et les données PL. Cet algorithme fournit une nouvelle classification des données en deux catégories. Les estimations étudiées dans les deux chapitres précédents sont appliquées aux nouvelles classes de données fournies ici.

Le chapitre 5 page 40, quant à lui, présente une autre manière de faire. En effet, la méthode du *bootstrap* non-paramétrique a ici été traitée sur les données pour tenter d'éliminer les éventuels points aberrants et d'avoir alors une meilleure estimation.

Le chapitre 6 page 44 présente la procédure finale qui a été adoptée afin de traiter les données par toutes les méthodes étudiées (mis à part le *bootstrap*). Cette procédure présente également un bilan de ces méthodes.

Enfin, avant la conclusion qui se trouve au chapitre 8 page 53, le chapitre 7 page 49 donne une méthode possible permettant de choisir une solution entre les différentes méthodes étudiées.

En annexe, on trouvera d'abord dans l'annexe A page 55 des explications concernant les outils mathématiques utilisés.

Ensuite, l'annexe B page 81 présente les données utilisées durant le stage.

Le code `Scilab` et `R` écrit durant le stage n'est pas donné dans le rapport. Il est disponible sous forme de fichiers informatiques.

Chapitre 2

Analyse des données par M-estimateurs

C'est une première méthode permettant de réduire la sensibilité aux valeurs jugées aberrantes dans une régression linéaire sans avoir à les supprimer. Le principe mathématique de cette méthode est expliqué dans l'annexe A.1 page 55. Pour illustrer les résultats de cette partie, le jeu de données utilisé est *Haguenau2009VL*, qui est un jeu de 128 données concernant des véhicules légers.

Les données, dont le contenu est plus détaillé dans l'annexe B page 81 sont représentées sur la figure 2.1 page suivante. Par exemple, les quatre points situés en haut à gauche de la figure pourraient être considérés comme des *outliers*.

En ce qui concerne l'implémentation dans `Scilab` de cette partie, une partie du code existait déjà. En effet, Renaud Wintzer, stagiaire au LRPC en 2010, avait commencé ce travail. J'ai donc apporté des modifications et des compléments au code existant ; il a également fallu l'adapter à mes fichiers.

2.1 Résultats de la régression linéaire

La M-estimation peut être considérée comme une régression linéaire robuste. Avant d'appliquer les M-estimateurs aux jeux de données, on a d'abord fait une simple régression linéaire suivant le modèle :

$$L_{Amax} = \beta_0 + \beta_1 X + \varepsilon \tag{2.1}$$

où X représente le paramètre adimensionnel de vitesse conforme au modèle cherché (voir définition dans 1.1 page 8) et ε l'erreur associée au modèle.

Cela a permis d'avoir un premier résultat, une première droite passant par les données et a servi à faire des comparaisons. De plus, c'est cette méthode qui est utilisée pour l'instant par les opérateurs et le logiciel `dBEuler`.

Comme cela a déjà été indiqué plus haut, la régression linéaire est très sensible aux données aberrantes. Cela se vérifie encore avec ces données. On voit en effet que la droite n'est pas ajustée par rapport au nuage de points « principal » et que les *outliers* « entraînent » la droite. Un exemple est celui du nuage de points situé plus haut. La figure 2.2 page 13 ci-dessous montre la droite obtenue par régression linéaire.

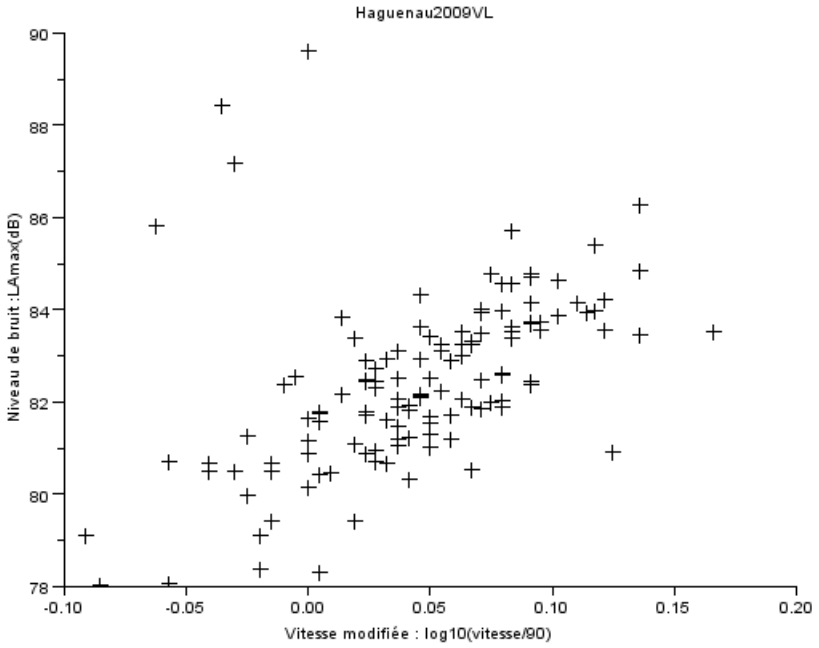


FIGURE 2.1 – Un jeu de données

L'équation de la droite obtenue par régression linéaire est :

$$L_{Amax} = 81.61 + 18.59 \times \log_{10}\left(\frac{vitesse}{90}\right) \tag{2.2}$$

Pour les autres jeux de données utilisés, la conclusion est la même : la régression linéaire simple n'apporte pas les résultats espérés. C'est pourquoi les personnes utilisant ces jeux de données suppriment les points qu'ils jugent aberrants. C'est aussi la raison pour laquelle nous nous tournons vers les M-estimateurs.

2.2 Estimation de l'échelle

Utiliser les M-estimateurs nécessite d'abord d'estimer σ qui représente l'échelle. Plusieurs possibilités s'offrent à nous afin de faire cette estimation. Ces possibilités sont expliquées dans l'annexe A.3 page 63.

La première solution est de calculer la valeur du MADN en ayant calculé une première estimation de type L_1. On peut raffiner cette valeur par le calcul itératif d'un M-estimateur de l'échelle. Pour ce dernier algorithme, la fonction de poids utilisée est celle dite de Hebert et Leahy et vaut :

$$\rho(r) = \ln\left(1 + r^2\right) \tag{2.3}$$

Les deux solutions ont été essayées pour plusieurs fichiers ; les résultats de l'estimation de l'échelle sont donnés dans le tableau 2.1 page suivante.

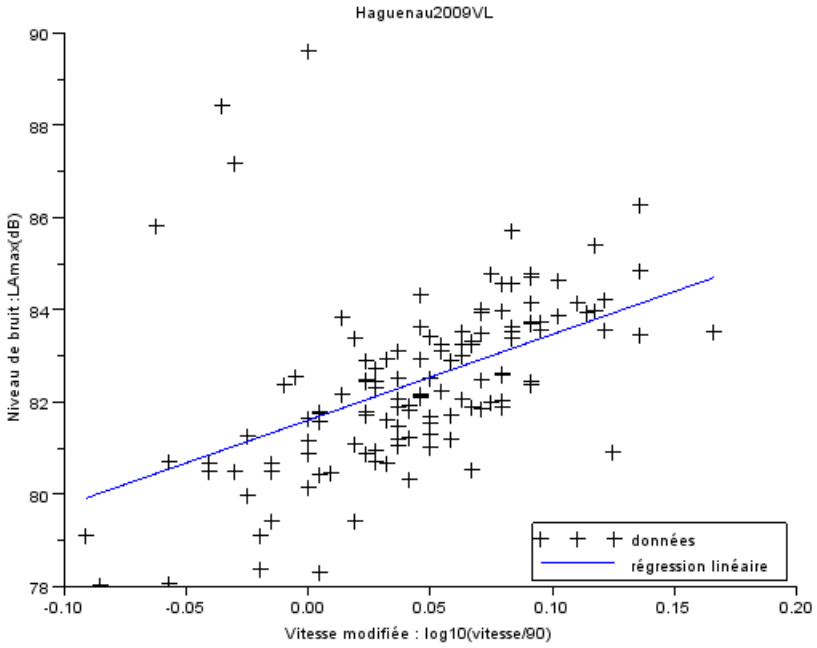


FIGURE 2.2 – Droite obtenue par régression linéaire

Méthode	Haguenau2009VL	Haguenau2009PL	Rothau2009VL	Rothau2009PL
Nb données	128	47	122	43
Solution L_1	1.0775	1.2690	2.0024	1.9571
MADN	1.0859	1.3050	1.3090	2.6118
Raffinement	1.1198	1.4238	1.5819	2.4950

TABLE 2.1 – Estimation de l'échelle

On remarque que les changements dans l'estimation de l'échelle sont plus petits quand le nombre de données est plus important. D'après mes maîtres de stage, le MADN est l'estimateur le plus utilisé. En effet, dans la littérature, on ne calcule souvent que cette quantité.

Dans d'autres fichiers de données de taille plus importante, les changements observés étaient très faibles. On peut donc penser que si le nombre d'observations est « assez grand », le MADN suffit pour estimer l'échelle. Celui-ci est également facile à calculer.

Malgré les changements observés dans le tableau ci-dessus, et après consultation avec mes maîtres de stage et grâce à leur expérience, nous décidons de nous limiter au calcul du MADN pour estimer l'échelle σ.

2.3 Fonctions utilisées pour la M-estimation et résultats

Comme cela a été vu dans les annexes, il y a différentes possibilités de fonctions de poids pour le calcul des M-estimateurs. Pour les estimations effectuées lors du stage, plusieurs fonctions ont été utilisées : la fonction ρ de Geman et McClure, ainsi que plusieurs fonctions de la SEF (*Smooth Exponential Family*).

2.3.1 la fonction de Geman et McClure

Nous rappelons la forme de cette fonction :

$$\rho_{GM}(r) = \frac{r^2}{1+r^2} \tag{2.4}$$

En utilisant cette fonction, on remarque un changement relativement important dans la droite de régression obtenue. Sur les données *Haguenau2009VL*, il semble que la droite se rapproche de ce qu'on attendait. Du moins elle semble se tourner dans l'axe du nuage de points « principal ».

Sur la figure 2.3, on trouve la droite obtenue par M-estimation de Geman et McClure. La droite obtenue par régression linéaire simple a été laissée à titre de comparaison.

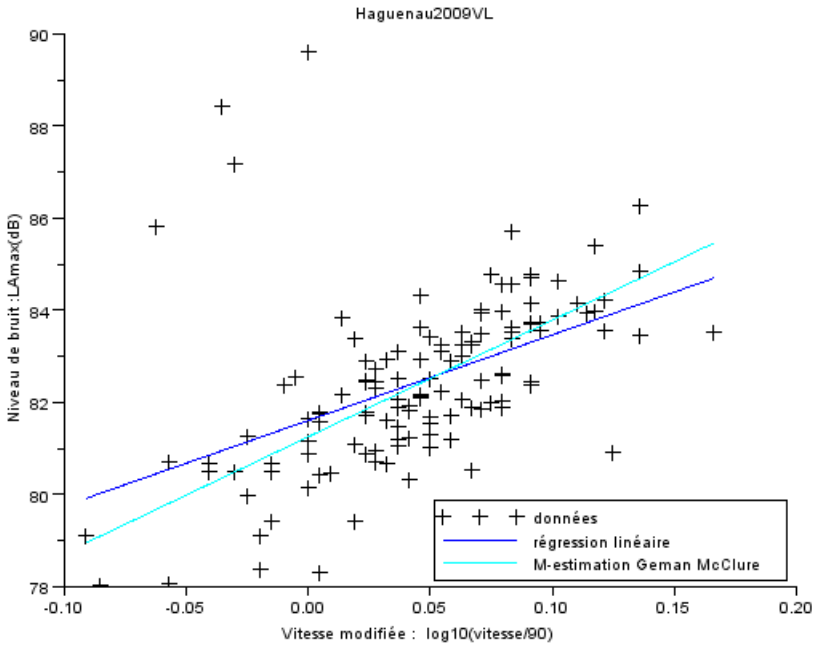


FIGURE 2.3 – Droite obtenue par M-estimation (Geman et McClure)

Dans le tableau 2.2 page suivante se trouvent les estimations de la pente et de l'ordonnée à l'origine des deux droites obtenues pour l'instant : régression linéaire et M-estimation avec fonction de poids de Geman et McClure. Pour ces données, les changements observés se trouvent surtout au niveau de la pente.

	pente	ordonnée à l'origine
Régression linéaire simple	18.6	81.6
M-estimation avec Geman & McClure	25.3	81.3

TABLE 2.2 – Comparaisons - paramètres de droites - données *Haguenau2009VL*

2.3.2 la Smooth Exponential Family

Comme cela est expliqué en annexe, cette fonction dépend d'un paramètre α. Rappelons la forme de cette fonction :

$$\rho_\alpha(r) = \frac{1}{\alpha}((1+r^2)^\alpha - 1) \tag{2.5}$$

Pour la M-estimation, les paramètres utilisés sont : 1, 0.5, 0, -0.5, -1. Ces valeurs ont été prises de manière arbitraire. Toutefois, en regardant de plus près, on se rend compte qu'en prenant $\alpha = -1$, on retombe sur la fonction ρ_{GM} de Geman et McClure. De plus, en prenant $\alpha = 1$, on obtient la régression linéaire simple.

En traçant les droites obtenues après ces estimations, on observe également des changements par rapport à la régression linéaire, même si certains sont moindres que ceux obtenus avec la fonction de Geman et McClure.

Les figures 2.4, 2.5 page suivante et 2.6 page 17 montrent la droite obtenue par régression linéaire simple à laquelle on a superposé la droite obtenue par M-estimation avec les fonctions de poids de la SEF où $\alpha = 0.5, 0, -0.5$.

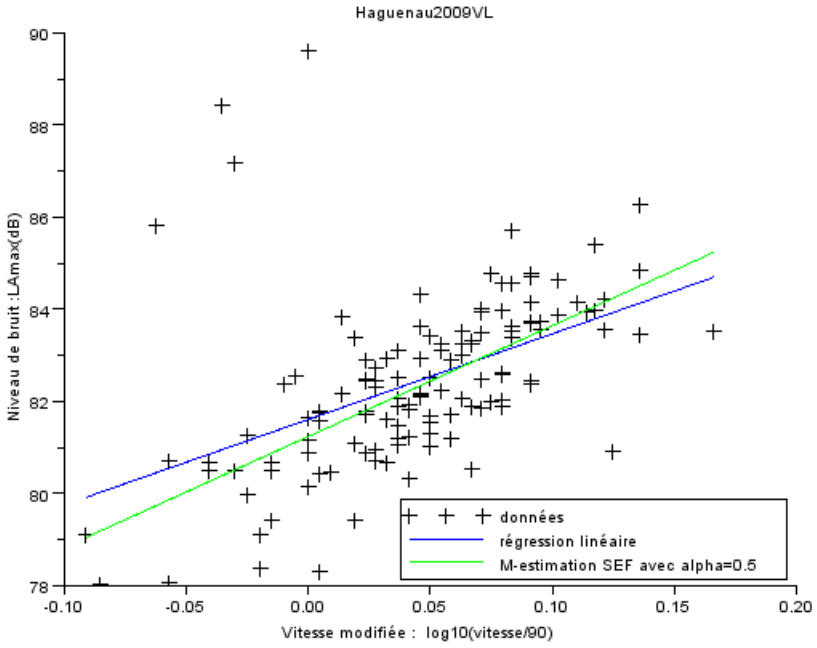


FIGURE 2.4 – Droite obtenue par M-estimation (SEF $\alpha = 0.5$)

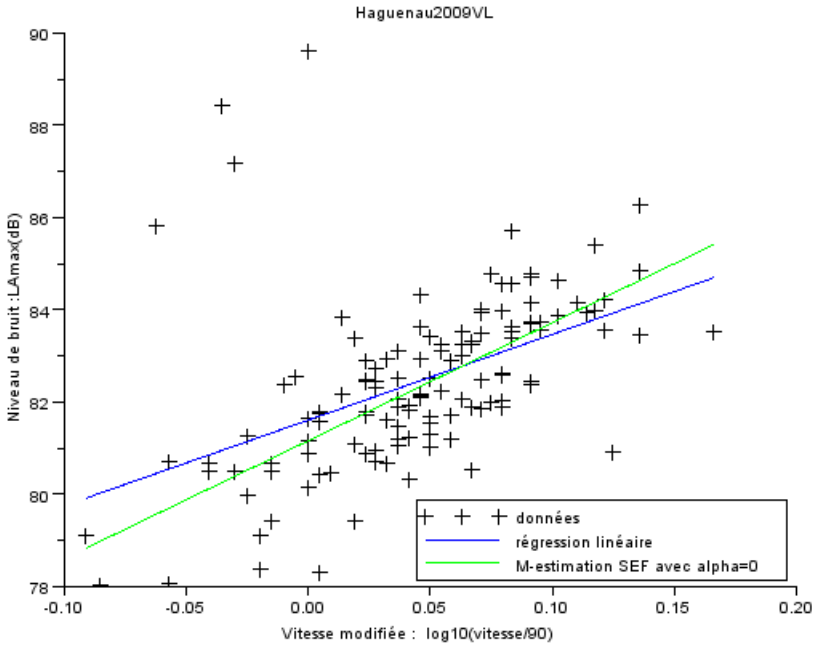


FIGURE 2.5 – Droite obtenue par M-estimation (SEF $\alpha = 0$)

A nouveau, nous donnons dans le tableau 2.3 les pentes et ordonnées à l'origine des droites obtenues. Afin de pouvoir comparer, les paramètres des droites concernant la régression linéaire et la M-estimation avec la fonction ρ_{GM} sont également ajoutés au tableau.

	pente	origine
Régression linéaire simple	18.6	81.6
M-estimation avec SEF - $\alpha = 0.5$	24.1	81.2
M-estimation avec SEF - $\alpha = 0$	25.6	81.2
M-estimation avec SEF - $\alpha = -0.5$	25.7	81.2
M-estimation avec Geman & McClure	25.3	81.3

TABLE 2.3 – Comparaisons - paramètres de droites - données *Haguenau2009VL*

Les estimations obtenues des paramètres sont plutôt stables entre les différentes fonctions utilisées pour calculer les M-estimateurs. La pente est très différente de celle obtenue par régression linéaire, mais l'ordonnée à l'origine reste quasiment la même partout.

2.3.3 Comparaisons et conclusions

Dans les paragraphes ci-dessus, nous avons vu les résultats de la régression linéaire ainsi que de divers M-estimateurs sur les données *Haguenau2009VL*. Nous allons donner les résultats obtenus pour les autres fichiers disponibles, à savoir les fichiers *Haguenau2009PL*, *Rothau2009PL*, *Rothau2009VL*, *Motos30082009*.

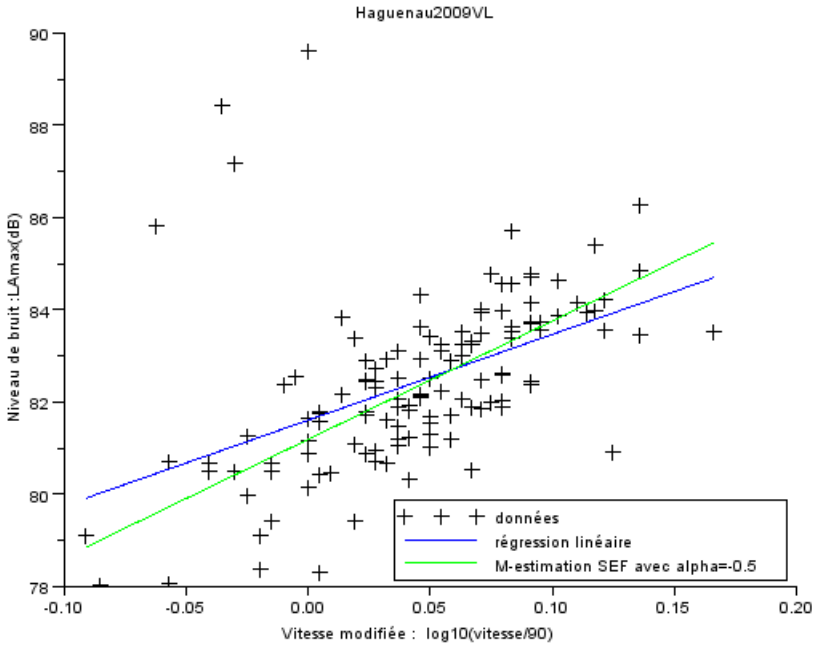


FIGURE 2.6 – Droite obtenue par M-estimation (SEF $\alpha = -0.5$)

Nous rappelons que ce qui intéresse l'opérateur dans cette étude est le niveau de bruit pour la vitesse de référence (qui correspond à l'ordonnée à l'origine) ainsi que la pente de la droite de régression obtenue. De plus, pour les fichiers « PL » (Poids Lourds), la vitesse de référence vaut 80 km/h et non 90.

Pour les motos, nous rappelons que les données correspondant aux deux sens de circulation peuvent être rassemblées.

Le tableau 2.4 page suivante rassemble donc toutes les coordonnées des droites obtenues par les différentes manières présentées dans ce chapitre, et ce pour tous les fichiers disponibles. La ligne *Régression linéaire « experte »* donne les résultats obtenus par les acousticiens lors du dépouillement des campagnes correspondantes, dans `dBEuler` avec élimination interactive des points aberrants.

Concernant la régression linéaire « experte », les données pour les motos ne sont pas disponibles pour les deux sens confondus.

On observe que selon les fichiers, les modifications sont plus ou moins importantes. Toutefois, dans tous les cas, la différence du niveau de bruit de référence est très faible ; c'est surtout la pente de la droite qui diffère selon les estimations.

D'autre part, si on observe les graphiques associés à chaque fichier et chaque estimation (voir planches en annexe), on voit que la M-estimation a l'air de moins bien fonctionner pour les données PL que pour les données VL. La droite obtenue ne s'aligne pas très bien par rapport à ce qu'on attend. En fait, on a environ trois fois moins de données pour les PL que pour les VL (environ 40 et 120 données). De plus, on sait que les méthodes statistiques fonctionnent souvent mieux quand le nombre de données augmente. Or le problème

	Haguenau2009VL		Haguenau2009PL		Rothau2009VL	
	pente	origine	pente	origine	pente	origine
Régression linéaire « experte »	25.6	81.1	36.2	86.7	30.9	78
Régression linéaire	18.6	81.6	32.0	86.5	14.2	77.5
SEF - $\alpha = 0.5$	24.1	81.2	34.3	86.7	24.4	77.8
SEF - $\alpha = 0$	25.6	81.2	37.4	86.7	29.7	78.0
SEF - $\alpha = -0.5$	25.7	81.2	39.7	86.7	30.5	78.0
Geman & McClure	25.3	81.3	41.1	86.6	30.4	78.0

	Rothau2009PL		Motos30082009	
	pente	origine	pente	origine
Régression linéaire « experte »	23.9	84.8		
Régression linéaire	14.4	84.1	31.5	79.6
SEF - $\alpha = 0.5$	16.5	84.1	28.9	79.2
SEF - $\alpha = 0$	18.0	84.0	26.3	78.9
SEF - $\alpha = -0.5$	18.6	83.8	24.4	78.6
Geman & McClure	18.6	83.6	23.2	78.5

TABLE 2.4 – Comparaisons - paramètres de droites

est que dans ces campagnes, l'opérateur ne peut mesurer que les véhicules qui passent sur la route étudiée. Le nombre de données est donc fonction du trafic au moment des mesures. En particulier, on imagine qu'il y a bien moins de poids lourds que de véhicules légers sur une route départementale. C'est peut-être une limite de ce type d'études, mais on ne peut faire autrement.

Finalement, on peut dire que de manière générale, les M-estimateurs ont bien fonctionné sur ces données : la droite se « déplace » conformément à ce qui était attendu. Par contre, le résultat est quand même moins bon sur les PL. En regardant le tableau, il semblerait que les fonctions de poids de Geman et McClure et de la SEF avec $\alpha = -0.5$ soient les plus proches de la régression experte.

D'autres données issues de campagnes de mesures, *Stotzheim2009* et *Erstein*, m'ont été mises à disposition. On peut donc comparer les résultats de la régression experte et de ces deux fonctions de poids sélectionnées pour ces nouvelles données. Les résultats sont rassemblés dans le tableau 2.5 page suivante.

Dans ces jeux de données, on observe que les ordonnées à l'origine des droites obtenues sont assez proches les unes des autres. Par contre, de grandes différences subsistent concernant les pentes, quelle que soit la catégorie de véhicules. Il faudrait étudier ces données de manière plus détaillée pour comprendre le problème.

Ce chapitre a traité de méthodes de régression robuste où le résidu n'est fonction que de la variable dépendante du modèle, à savoir ici le L_{Amax}. Le chapitre suivant s'intéresse à une approche où le résidu dépend de deux variables.

	Stotzheim2009VL		Stotzheim2009PL	
	pente	origine	pente	origine
Régression linéaire « experte »	18	75.6	21.4	83.5
SEF - $\alpha = -0.5$	14.1	75.9	28.7	83.3
Geman & McClure	11.8	76.1	33.4	83.2

	ErsteinVL		ErsteinPL	
	pente	origine	pente	origine
Régression linéaire « experte »	25.9	72.5	41	81.2
SEF - $\alpha = -0.5$	26.9	72.5	35.1	81.3
Geman & McClure	27.2	72.4	35.3	81.2

TABLE 2.5 – Comparaisons - paramètres de droites avec d'autres données

Chapitre 3

Analyse des données par ACP

3.1 Introduction

L'ACP est une méthode de régression prenant en compte les erreurs d'une manière différente de la régression linéaire. En effet, lorsqu'on n'a que deux variables, comme c'est le cas ici, l'ACP consiste à prendre en compte les erreurs non seulement par rapport aux ordonnées mais aussi par rapport aux abscisses. Cela semble plus logique car les deux variables L_{Amax} et la vitesse sont entachées d'incertitudes. Les techniques de l'ACP et de l'ACP robuste sont introduites dans l'annexe A.4 page 66.

Au lieu de chercher à minimiser les erreurs d'ordonnée entre la droite et les points observés comme en régression linéaire, on minimise la distance euclidienne à la droite cherchée.

L'ACP robuste permet, quant à elle, de prendre en compte des données aberrantes, sur le même principe que les M-estimateurs.

Comme cela est présenté dans le chapitre, plusieurs façons de faire ont été essayées. De plus, un problème est survenu lors de l'implémentation. En effet, les figures présentées dans les chapitres précédents sont tracés dans un repère non orthonormé. Compte tenu de la différence entre les valeurs des deux variables (entre 78 et 90 pour l'axe des ordonnées, pour *Haguenau2009VL* et entre -0.1 et 0.2 pour l'axe des abscisses), ce n'est pas possible d'utiliser un tel repère pour la représentation des données.

Or cela pose un problème pour les calculs. Pour l'ACP, on peut dire que l'on calcule la distance euclidienne entre les points et la droite cherchée. Comme les valeurs sur les deux axes sont très différentes, la distance est biaisée par rapport à un axe, et « avantage » l'axe ayant les plus grandes valeurs. C'est pourquoi les droites obtenues dans ce cas sont assez verticales.

Ce problème n'apparaissait pas dans le cas de la régression linéaire, car on ne calcule la distance que par rapport à l'axe des ordonnées. Les échelles des deux axes n'ont donc pas d'importance.

Pour parer à ce problème, on effectue une transformation des données, appelée ici « normalisation », afin de ramener les données dans le rectangle $[-1, 1] \times [-1, 1]$. Il faut ensuite revenir au repère de départ afin de comparer avec les résultats précédents. Le problème de cette méthode est qu'elle n'est pas neutre par rapport aux points aberrants. En effet, ce sont ces points-là qui vont être, par exemple, le maximum des données et qui vont alors être ramenés à 1.

La transformation à effectuer est simple. Comme on utilise cette méthode plus tard en confondant les données VL et PL, pour l'algorithme EM (cf 4 page 29), on effectue la nor-

malisation en prenant en compte les deux jeux de données. Supposons qu'ils soient répartis dans le rectangle $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$. Pour chaque point (x, y), on calcule le point (x_{norm}, y_{norm}) suivant :

$$x_{norm} = \frac{2}{x_{max} - x_{min}} x - \frac{x_{max} + x_{min}}{x_{max} - x_{min}} \tag{3.1}$$

$$y_{norm} = \frac{2}{y_{max} - y_{min}} y - \frac{y_{max} + y_{min}}{y_{max} - y_{min}} \tag{3.2}$$

On fait la transformation inverse pour revenir aux données de départ. De même, il y a une transformation à faire concernant les paramètres de droite. On obtiendra des droites de la forme : $y_{norm} = a_{norm} x_{norm} + b_{norm}$. Or on aimerait : $y = ax + b$. Cela donne :

$$a = a_{norm} \frac{y_{max} - y_{min}}{x_{max} - x_{min}} \tag{3.3}$$

et

$$b = \left(a_{norm} \frac{x_{max} + x_{min}}{x_{max} - x_{min}} + b_{norm} - \frac{y_{max} + y_{min}}{y_{max} - y_{min}} \right) \times \frac{y_{max} - y_{min}}{2} \tag{3.4}$$

On peut alors comparer les différentes estimations des paramètres de ces droites avec ceux obtenus précédemment.

Dans ce chapitre, on présente quelques résultats obtenus avant de normaliser les données afin de montrer le problème. Ensuite, l'étude est faite en ayant effectué cette transformation. Les graphiques sont présentés uniquement pour les données *Haguenau2009VL*. Comme dans le chapitre précédent, des résultats chiffrés sont donnés pour les autres fichiers.

J'ai réalisé moi-même l'implémentation en langage `Scilab` de cette partie, avec l'aide d'un code en langage `Matlab` de Pierre Charbonnier.

3.2 ACP sans normalisation des données

3.2.1 Sur les paramètres adimensionnels de vitesse (en prenant le $\log_{10}$)

Comme dans le chapitre précédent, on regarde d'abord ce que donne l'ACP classique, avant d'appliquer une fonction de poids aux erreurs. Le graphique 3.1 page suivante présente la droite obtenue par ACP, à laquelle a été superposée la droite obtenue par régression linéaire. On observe que l'ACP est également très sensible aux données aberrantes, mais d'une façon différente de la régression linéaire.

En effet, les deux droites sont très différentes, mais aucune ne correspond à ce qui est attendu. On remarque aussi l'effet de la différence de valeurs entre les axes qui a été expliqué en introduction.

On peut faire la comparaison au niveau des équations de ces droites. Pour la régression experte, on avait obtenu l'équation :

$$L_{Amax} = 81.1 + 25.6 \times \log_{10} \left(\frac{vitesse}{90} \right) \tag{3.5}$$

Pour la droite de l'ACP, sans normalisation des données, on a :

$$L_{Amax} = 78.8 + 82.3 \times \log_{10} \left(\frac{vitesse}{90} \right) \tag{3.6}$$

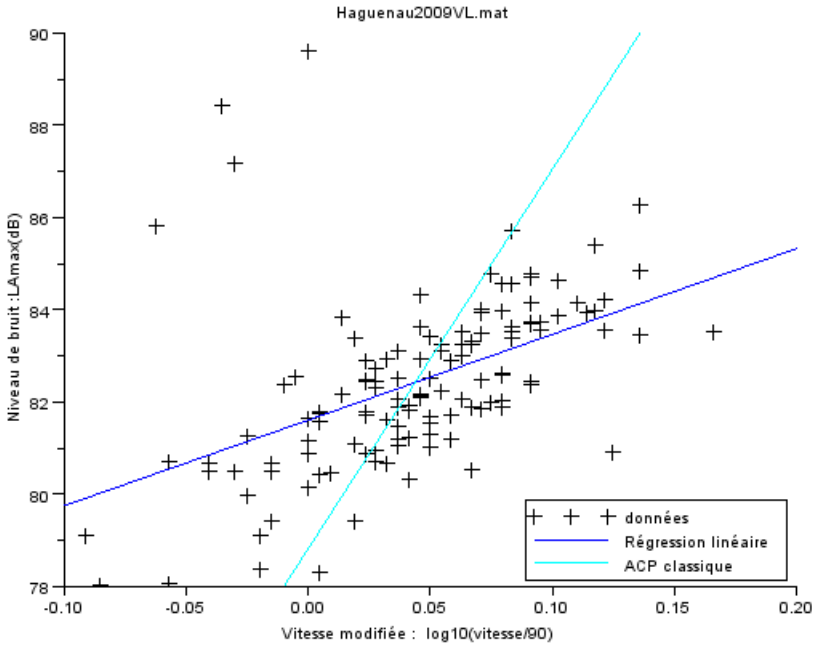


FIGURE 3.1 – Droite obtenue par ACP

Non seulement la différence entre les deux ordonnées à l'origine n'est pas négligeable, mais de plus la pente est très différente entre ces deux droites.

Un essai a également été effectué en faisant l'ACP sur les données brutes et non sur le paramètre adimensionnel de vitesse conformément au modèle cherché. Comme nous le voyons sur le graphique 3.2 page suivante, la technique a l'air de mieux se comporter. En fait, les deux droites de la régression linéaire et de l'ACP sont presque confondues. Cela peut s'expliquer, en partie, par le fait que, en prenant les vitesses brutes, on a à peu près les mêmes valeurs en abscisse et en ordonnée (on pourrait dire, entre 60 et 100). En tout cas, la différence est bien moindre que celle qu'on a avec les vitesses prises en $\log_{10}$. Le problème expliqué ci-dessus est donc beaucoup moins important ici.

Le problème est que pour les normes utilisées par le groupe « Acoustique » (détails dans [31-93]), il est nécessaire d'avoir un modèle linéaire par rapport au paramètre a dimensionnel de vitesse. Si on ajuste un modèle linéaire par rapport aux vitesses brutes, on ne pourra pas avoir le modèle souhaité dans le cadre de ces campagnes.

3.2.2 Variante sur les variables utilisées

Ci-dessus, j'ai calculé les estimations de l'ACP sur les vitesses brutes. On a vu que les résultats étaient meilleurs. Comme cela a été montré sur le graphique 3.2 page suivante, l'ACP et la régression linéaire sont très semblables dans ce cas.

Un autre essai a été de faire les calculs avec les vitesses brutes, mais d'afficher le résultat en fonction du paramètre adimensionnel de vitesse (variable X). On n'a alors plus de droites

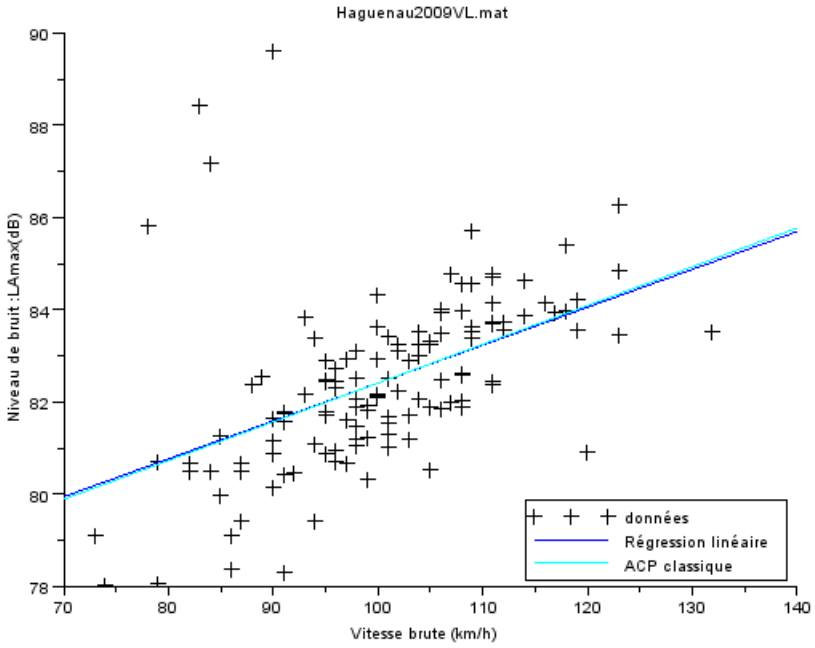


FIGURE 3.2 – Droite obtenue par ACP sur les vitesses brutes

mais des courbes (le modèle devient logarithmique). Le graphique 3.3 page suivante présente les courbes obtenues pour la régression linéaire, l'ACP et l'ACP robuste (voir à la fin du chapitre) avec poids de Geman et McClure. On remarque que la courbe issue de l'ACP robuste est relativement proche de ce qu'on pourrait espérer.

Comme cela a déjà été souligné, cet essai ne peut pas être exploité tel quel car le modèle ainsi obtenu ne correspond plus à ce qui est demandé dans les normes (il n'est plus linéaire par rapport au paramètre adimensionnel de vitesse). Il faut en effet un modèle du type $L_{Amax} = a \times \log_{10}\left(\frac{vitesse}{v_{ref}}\right) + b$.

On en conclut que malgré les résultats, qui sont assez probants, on ne peut pas utiliser cette méthode : le modèle n'est pas adapté.

3.3 ACP après normalisation des données

Il faut donc normaliser les données avant d'effectuer l'ACP, afin de ne favoriser aucun axe lors du choix de la droite.

3.3.1 ACP classique

Après avoir calculé les estimations des paramètres de la droite dans le repère orthonormé, il a fallu transformer ces estimations pour revenir dans le repère de départ. La figure 3.4 page 25 montre les droites obtenues par ACP et régression linéaire.

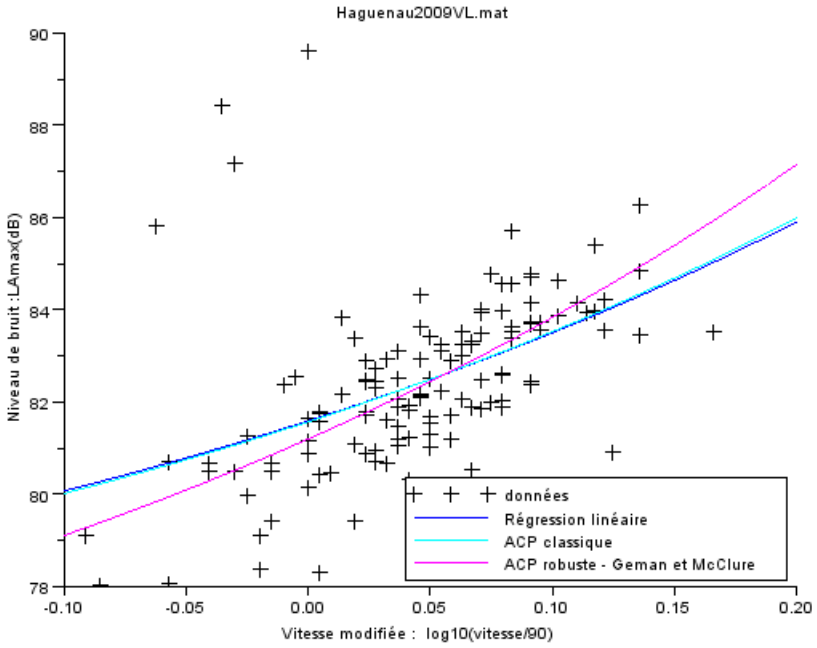


FIGURE 3.3 – Courbes obtenues par régressions sur les vitesses brutes

Contrairement à ce qui avait été obtenu précédemment (voir 3.1 page 22), la droite issue de l'ACP se rapproche de celle obtenue par régression linéaire. On peut comparer les équations de ces deux droites. Rappelons l'équation de la régression linéaire :

$$L_{Amax} = 81.61 + 18.59 \times \log_{10}\left(\frac{vitesse}{90}\right) \tag{3.7}$$

Celle de l'ACP est :

$$L_{Amax} = 81.15 + 29.26 \times \log_{10}\left(\frac{vitesse}{90}\right) \tag{3.8}$$

Nous allons passer maintenant à l'ACP robuste. Elle est à l'ACP ce que les M-estimateurs sont à la régression linéaire : on utilise une fonction de poids pour prendre en compte les observations ayant un résidu important.

3.3.2 ACP robuste

Estimation de l'échelle

Comme pour la M-estimation, il s'agit d'abord d'estimer l'échelle σ avant d'estimer les paramètres des différentes droites. Cela a posé problème au début de l'implémentation de cette méthode, car je n'avais pas pensé à cette estimation et je ne l'avais donc pas incluse dans l'algorithme.

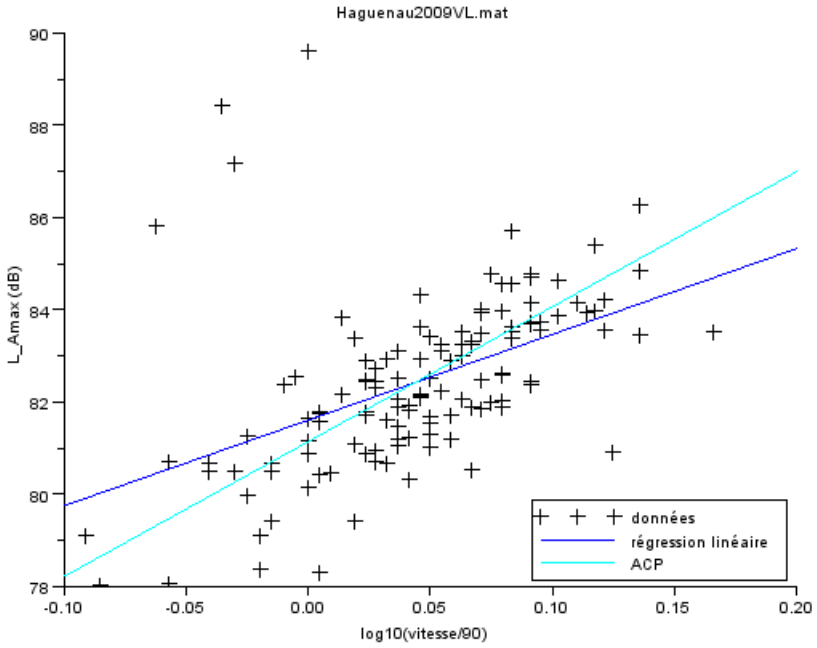


FIGURE 3.4 – Courbes obtenues par régression linéaire et ACP sur données normalisées

Dans la partie M-estimation, on appliquait un pseudo-algorithme IRLS pour estimer l'échelle grâce à la norme L_1. Cet algorithme-là utilisait lui aussi les M-estimateurs. Il a donc fallu modifier la fonction de calcul de l'échelle : celle-ci est calculée avec une ACP robuste. La fonction de poids utilisée est la même que pour les M-estimateurs, et vaut :

$$\rho(r) = \sqrt{\varepsilon + r^2} \quad \text{avec} \quad \varepsilon \to 0 \tag{3.9}$$

De la même façon que pour les M-estimateurs, on pourrait raffiner l'estimation de l'échelle par un algorithme itératif supplémentaire après le calcul du MADN. A nouveau, on s'arrêtera au calcul de ce dernier pour estimer l'échelle. Toutefois, la comparaison entre les deux méthodes n'a pas été faite pour cette partie.

Résultats de l'ACP robuste

A nouveau, on peut utiliser différentes fonctions de poids, comme par exemple celles de la « *Smooth Exponential Family* » ou celle de Geman et McClure qui ont déjà été utilisées précédemment. Ces fonctions ont été testées, mais les changements n'étaient pas significatifs. Cela a conduit à ne garder que la fonction de Geman et McClure pour cette étude.

De même, il faut encore faire les calculs avec les données normalisées et faire la transformation inverse pour comparer les résultats et visualiser les graphiques.

Le graphique 3.5 page suivante ci-dessous montre les droites obtenues par ACP classique et par ACP robuste. Pour ce fichier comme pour les autres, on observe des différences plus ou moins grandes dans la droite obtenue. De la même manière qu'entre régression linéaire et

M-estimation, la droite obtenue ici se rapproche de l'« axe principal » du nuage de points, ce qui est bon signe quant à l'efficacité de la méthode.

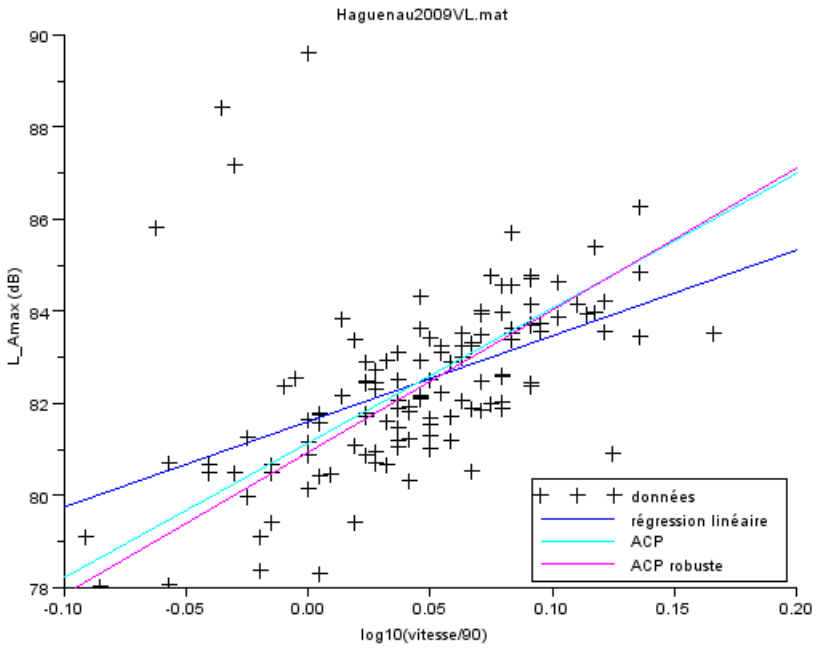


FIGURE 3.5 – Droites obtenues par ACP et ACP robuste

Le tableau 3.1 montre les coordonnées des deux droites issues de l'ACP ainsi que celles de la droite obtenue par régression linéaire.

	pente	ordonnée à l'origine
Régression linéaire simple	18.6	81.6
ACP classique	29.3	81.2
ACP robuste avec poids de Geman & McClure	30.9	80.9

TABLE 3.1 – Comparaisons - paramètres de droites - données *Haguenau2009VL*

De la même manière qu'entre régression linéaire et M-estimation, la droite obtenue ici se rapproche de l'« axe principal » du nuage de points, ce qui est bon signe quant à l'efficacité de la méthode. Dans le paragraphe 3.3.3, une comparaison sera faite entre cette méthode et la méthode des M-estimateurs.

3.3.3 Comparaisons et conclusions

Nous allons faire un bilan des résultats obtenus pour l'instant. Le graphique 3.6 page suivante montre les droites obtenues pour les estimations suivantes : régression linéaire, M-estimateur avec poids de Geman et McClure, ACP classique et ACP robuste avec poids de

Geman et McClure. Bien sûr, les paramètres de ces dernières droites ont été calculés après avoir normalisé les données, comme indiqué en introduction de ce chapitre.

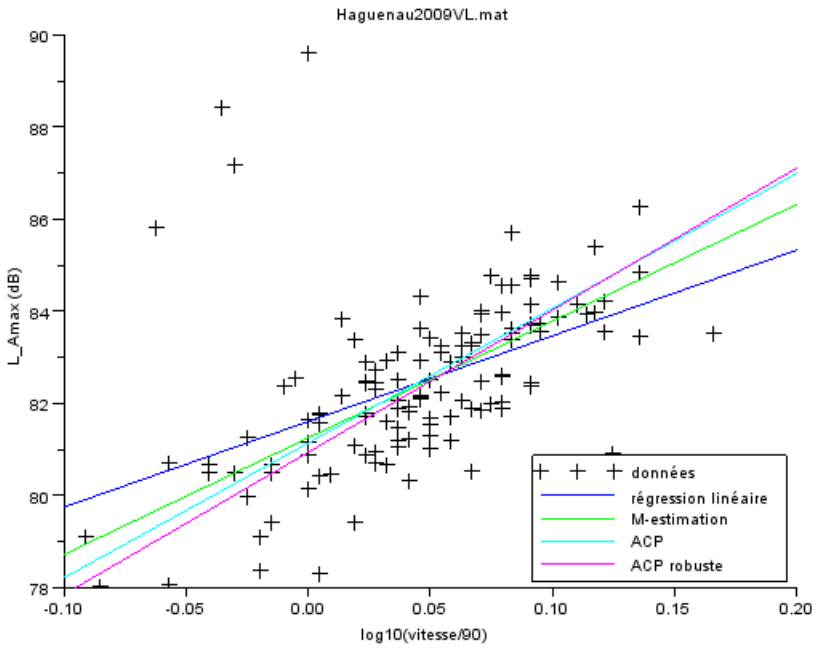


FIGURE 3.6 – Différentes estimations

Le tableau 3.2 page suivante montre également les paramètres des droites citées ci-dessus, et ce pour les fichiers disponibles. Cela permettra alors une première comparaison des méthodes.

On voit que, en général, les droites issues de l'ACP et de l'ACP robuste sont relativement proches l'une de l'autre. Cela peut s'expliquer par le fait que l'initialisation de la procédure d'ACP robuste est la droite de l'ACP.

Par contre, les droites obtenues par régression linéaire et par M-estimation sont un peu plus éloignées.

A nouveau, entre le robuste et le non-robuste, on a assez peu de différences en ce qui concerne l'ordonnée à l'origine. Par contre, les écarts entre les pentes sont beaucoup plus grands. On peut tout de même voir, sur la figure 3.6, que les droites obtenues par M-estimation et par les deux méthodes d'ACP sont assez proches de ce qu'on pourrait attendre.

On remarque également, en regardant les paramètres pour les données *Haguenau2009PL*, que les écarts sont très grands entre les différentes estimations. En effet, les observations correspondantes sont assez disparates. Il semble assez difficile, à l'oeil nu, de tracer une droite passant par ces données (on peut voir ces observations sur les figures du chapitre 4 page 29).

Après avoir consulté des personnes du groupe « Acoustique », il semblerait que la méthode de l'ACP (et donc de l'ACP robuste) ne serait pas à retenir, au vu des résultats sur les fichiers disponibles. En effet, cette méthode ne donne pas les résultats attendus. L'interprétation

	Haguenau2009VL		Haguenau2009PL		Rothau2009VL	
	pente	origine	pente	origine	pente	origine
Régression linéaire « experte »	25.6	81.1	36.2	86.7	30.9	78
Régression linéaire	18.6	81.6	32.0	86.5	14.2	77.5
M-estimation (poids GM)	25.3	81.3	41.1	86.6	30.4	78.0
ACP	29.3	81.2	106.9	84.7	29.0	78.8
ACP robuste (poids GM)	30.9	80.9	65.3	86.1	32.3	78.2

	Rothau2009PL		Motos30082009	
	pente	origine	pente	origine
Régression linéaire « experte »	23.9	84.8		
Régression linéaire	14.4	84.1	31.5	79.6
M-estimation (poids GM)	18.6	83.6	23.2	78.5
ACP	26.8	85.1	41.7	80.9
ACP robuste (poids GM)	29.1	84.6	38.7	81.1

TABLE 3.2 – Comparaisons - paramètres de droites

proposée est la suivante : bien que la vitesse soit mesurée avec un appareil et donc entachée d'incertitude (ce qui suggère de prendre en compte les deux variables L_{Amax} et vitesse dans le calcul du résidu), cette incertitude sur le paramètre adimensionnel de vitesse est faible par rapport à celle des mesures de niveau sonore.

Chapitre 4

Classification non supervisée

Le code `Scilab` écrit pour ce chapitre est issu d'un code `Matlab` écrit par Pierre Charbonnier. J'y ai toutefois apporté quelques modifications.

4.1 Faits observés et problème

Pour chaque campagne de mesures de bruit de roulement, l'étude est faite séparément pour les VL (véhicules légers) et les PL (poids lourds). Nous avons également remarqué que dans chaque fichier se trouvent des données aberrantes, à savoir qui ne correspondent pas aux autres données.

Or si on superpose, pour une même campagne, les données VL et PL, on observe que les points aberrants de la catégorie VL se trouvent dans le nuage de points des PL, et inversement. Il est par ailleurs à remarquer qu'on doit, dans ce cas, utiliser les vitesses brutes. En effet, dans le cas du paramètre adimensionnel de vitesse (variable X définie en 1.1 page 8), il y aurait un problème d'échelle (la vitesse de référence n'étant pas la même pour les PL et les VL).

Le graphique 4.1 page suivante montre les données VL et PL du jeu *Haguenau2009* avec lequel nous travaillons. On voit que quatre points VL sont dans le nuage de points PL. De même, un point PL pourrait correspondre à une mesure VL. Ce sont précisément ces cinq points qui posent le plus de problèmes dans les régressions car ils « attirent » vers eux les droites estimées.

Plusieurs questions se posent alors : d'où viennent ces points aberrants ? Peuvent-ils provenir d'une erreur de l'opérateur ? Les erreurs potentielles ont déjà été inventoriées, rappelons-les ici. Plusieurs sources d'erreurs sont envisageables : au moment du passage du véhicule (enregistrement de la mauvaise catégorie ou d'une vitesse erronée), au moment du dépouillement (erreur de transcription des deux paramètres dans le logiciel `dBEuler`). On pourrait également avoir un véhicule léger qui fait beaucoup plus de bruit que la « moyenne », ou un poids lourd qui fait très peu de bruit : dans ce cas, on obtient une valeur aberrante, mais c'est une mesure réelle et sans erreur. Peut-on retrouver les deux classes par analyse de données ?

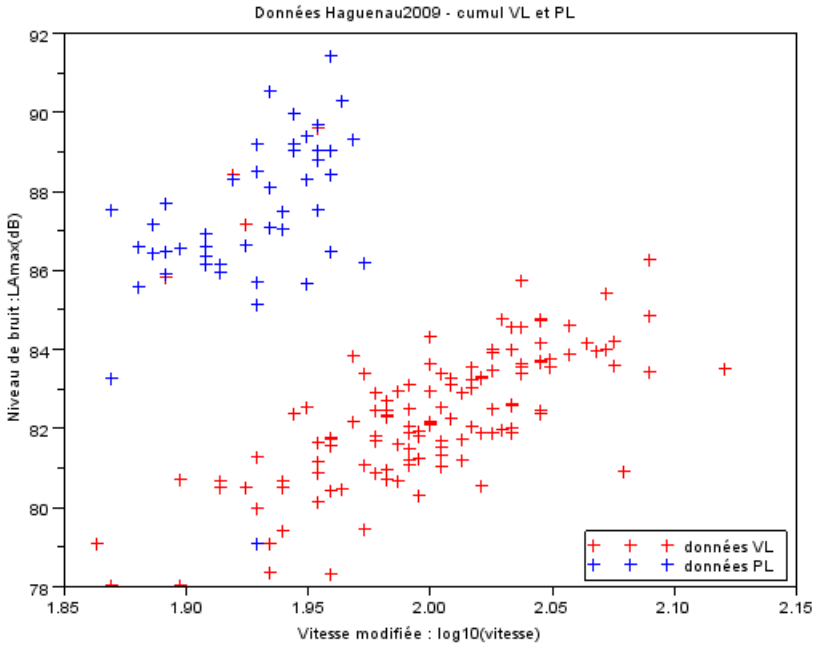


FIGURE 4.1 – Haguenau2009 : Superposition de données VL et PL

4.2 Principe général de cette analyse

Ce qu'on cherche à faire est une classification non supervisée, c'est-à-dire qu'on note les catégories des véhicules, puis on les « oublie ». Ensuite, à l'aide d'un algorithme EM (dont l'explication théorique se trouve en annexe A.5 page 70), on essaye de retrouver les deux classes.

On discutera à la fin du chapitre des applications possibles de cette analyse. En effet, des problèmes peuvent se poser : par exemple, peut-on assimiler un VL bruyant à un PL dans l'estimation des droites de régression ?

Il reste à régler un problème de représentation. On ne peut pas utiliser les paramètres adimensionnels de vitesse (variable X) telles qu'elles ont été définies au départ, car les deux catégories de véhicules n'ont pas la même vitesse de référence. Cela poserait alors un problème d'échelle. Mais on ne peut pas non plus prendre les vitesses brutes. En effet, si, après la classification, on refait une estimation des droites, le modèle obtenu ne sera pas conforme aux normes utilisées. Ce problème avait déjà été rencontré dans le chapitre sur l'ACP.

Un compromis a alors été trouvé. Pour traiter les vitesses, on prendra les variables :

$$\log_{10}(vitesse) \tag{4.1}$$

Comme on n'utilise pas de vitesse de référence, on n'aura pas de problème d'échelle entre les deux catégories. Comme les vitesses sont en « log », on pourra analyser les nouvelles données obtenues. Si on utilise les propriétés du logarithme, il suffira de faire l'opération suivante pour

retrouver les variables utilisés dans les calculs :

$$X = \log_{10}(vitesse) - \log_{10}(v_{ref}) \tag{4.2}$$

On notera également que le but de cette analyse n'est pas d'enlever le travail de l'opérateur, mais juste de l'aider dans le dépouillement des mesures et de parer à certaines erreurs qui ont pu être commises.

4.3 Algorithme utilisé

L'algorithme utilisé est expliqué à l'annexe A.5 page 70, mais nous allons le présenter rapidement de façon non mathématique pour fixer les idées. A la base, on a deux classes, VL et PL, utilisées pour l'initialisation de l'algorithme.

Après avoir pris quelques caractéristiques de ces classes (moyenne, matrice de variance covariance, proportion de points dans chaque classe), on oublie ces étiquettes pour ne garder qu'une seule classe. L'algorithme utilisé est itératif.

Par une formule et grâce aux caractéristiques récupérées, on calcule pour chaque point sa « chance » de se trouver dans chaque classe. On utilise alors ces chances comme des poids pour calculer de nouvelles moyennes et matrices de variance-covariance pondérées. On peut à nouveau calculer la proportion d'individus dans chaque classe, puis les chances pour chaque point d'appartenir à une classe.

Cela est fait un nombre fini de fois (par exemple 20 ou 50), jusqu'à stabilisation. A la fin, on a donc la probabilité qu'a chaque point de se retrouver dans chacune des deux classes. On regarde alors pour chaque point la catégorie pour laquelle cette probabilité est la plus grande. La catégorie sélectionnée est celle à laquelle le point appartient à la fin de l'algorithme.

Comme convenu, on obtient donc deux nouvelles classes de points. Les points peuvent être positionnés dans des classes différentes avant et après cet algorithme.

Par rapport à l'utilisation qu'on veut en faire ici, on classe les points en trois groupes :
- les points dits *VL*, à savoir ceux qui n'ont pas changé de classe et qui sont restés dans la catégorie VL ;
- les points dits *PL*, à savoir ceux qui n'ont pas changé de classe et qui sont restés dans la catégorie PL ;
- les points dits *incertains* : ce sont ceux qui ont changé de catégorie pendant l'algorithme. Ce sont les points pour lesquels l'opérateur devra vérifier qu'aucune erreur n'a été faite. Il pourra ensuite les reclasser ou les éliminer.

Le programme Scilab calcule également le nombre de points de chaque catégorie qui ont été affectés à l'autre catégorie suite à l'algorithme EM. De plus, trois fichiers texte sont créés : ils contiennent les coordonnées des points de chacun des trois groupes constitués, ainsi que leur catégorie d'origine.

Plusieurs affichages sont faits au début et à la fin du programme : les deux catégories de points telles qu'elles le sont au début, les moyennes et les ellipses d'isoprobabilité à deux écarts-type avant l'algorithme (cela représente les caractéristiques de la loi utilisée pour la première itération). Sur ce graphique, on trouvera aussi la droite issue de l'ACP, calculée avant l'algorithme.

Ces mêmes affichages sont faits à la fin de l'algorithme afin de montrer quels changements ont été apportés par ce dernier.

Nous avons soulevé, dans le chapitre 3 page 20, un problème concernant les données. En effet, comme les données correspondant aux abscisses et aux ordonnées prennent des valeurs très différentes, les résultats donnés par l'ACP sont faussés. Il avait fallu introduire une « normalisation » des observations, afin de mettre les données dans le carré $[-1,1] \times [-1,1]$. Ceci est également valable ici, puisqu'on utilise l'ACP. Certains graphiques seront donc présentés pour les données normalisées. Les graphiques servant aux comparaisons seront montrés, comme dans le chapitre précédent, de manière classique.

4.4 Test de normalité

Lors de l'application de l'algorithme EM, on suppose que les données sont issues d'une loi normale. Or cela n'est pas évident a priori. J'ai donc utilisé un test de normalité, que j'ai appliqué aux données disponibles. Celui-ci est issu du cours [Ber11].

Comme les données à tester sont à deux dimensions, il a fallu utiliser un test de multinormalité. Le test choisi est une modification du test de Shapiro-Wilk unidimensionnel qui date du début des années 1970.

Les hypothèses de ce test sont les suivantes :

- $\mathcal{H}_O$: les données sont issues d'une loi normale
- $\mathcal{H}_1$: les données ne sont pas issues d'une loi normale

On sait qu'un vecteur $V = (V_1, V_2)$ est gaussien si et seulement si toutes les combinaisons linéaires de ses composantes sont normales. La statistique utilisée pour ce test est alors la suivante :

$$W = \inf_A W_Y \tag{4.3}$$

où A est l'ensemble des combinaisons linéaires Y des composantes de X et W_Y est la valeur de la réalisation de la statistique de Shapiro-Wilk calculée pour la combinaison linéaire Y. Cette dernière valeur est calculée grâce à l'estimateur corrigé de la variance de l'échantillon et aux statistiques d'ordre de celui-ci. Pour plus de détails sur ce test, le lecteur consultera [Ber11].

On cherche, dans ce test, à trouver une combinaison linéaire Y qui pourrait réaliser le minimum W.

Soit c la valeur critique du test. L'hypothèse nulle (donc l'hypothèse de multinormalité) sera acceptée lorsque $W \geqslant c$. On peut dire aussi que l'hypothèse nulle $\mathcal{H}_O$ sera acceptée au niveau $\alpha = 5\%$ lorsque la p-valeur associée au test est supérieure ou égale à ce niveau α. Dans ce cas, il faudrait calculer la puissance du test pour connaître l'erreur (qui est une erreur de deuxième espèce) commise en prenant la décision d'accepter l'hypothèse nulle.

Dans le cadre du stage, ce test a été appliqué au vecteur $V = (\log_{10}(vitesse), L_{Amax})$ puisque c'est ce vecteur qui est supposé gaussien lors de l'application de l'algorithme EM.

Comme ce test a été appliqué à titre de vérification, cela a été fait après la classification issue de l'algorithme EM. Cela évite d'avoir trop de points aberrants. La fonction `mshapiro.test` de la bibliothèque `mvnormtest` de `R` a ainsi été utilisée. Les résultats obtenus sont présentés dans le tableau 4.1 page suivante.

Données	P-valeur du test
Haguenau2009VL	0.17
Haguenau2009PL	0.40
Rothau2009VL	2.2×10^{-5}
Rothau2009PL	0.46

TABLE 4.1 – P-valeurs du test de normalité

On observe que l'hypothèse nulle $\mathcal{H}_0$ de multinormalité est acceptée au niveau $\alpha = 5\%$ pour trois des quatre jeux de données testés, à savoir *Haguenau2009VL*, *Haguenau2009PL* et *Rothau2009PL*. Par contre, cette même hypothèse est rejetée pour le jeu de données *Rothau2009VL*, au même niveau.

Pour cette dernière classe, lorsqu'on regarde graphiquement, on voit que quelques points, même s'ils sont classés dans la catégorie VL, peuvent poser problème quant à l'hypothèse de normalité. Cela est le cas, par exemple, des trois points d'abscisses les plus faibles. Le test a été effectué à nouveau en enlevant ces trois points. Dans ce cas, la p-valeur obtenue pour le test de Shapiro-Wilk pour la multinormalité est $p = 0.09$. L'hypothèse nulle est donc acceptée au seuil de $\alpha = 5\%$. Le risque associé à cette décision est un risque de seconde espèce que l'on pourrait évaluer à l'aide de la puissance du test.

Finalement, après classification, on peut supposer que les données issues des classes obtenues suivent une loi normale à deux dimensions. Par contre, ce n'est pas une preuve : le test a été fait après avoir enlevé les points aberrants, et même, pour l'un des jeux de données, après avoir enlevé quelques points « gênants ».

L'algorithme EM peut alors être appliqué aux données disponibles.

4.5 Résultats obtenus

Comme précédemment, nous allons montrer les résultats obtenus avec cet algorithme pour les données *Haguenau2009*. Comme expliqué dans la section précédente, on utilise pour les vitesses les variables $\log_{10}(vitesse)$. On ne peut donc pas comparer directement les équations de droites obtenues avec celles des chapitres précédents et des autres méthodes. Toutefois, un calcul simple permet de passer d'une équation à l'autre.

Par contre, les équations d'avant et après l'algorithme EM sont directement comparables.

Les graphiques suivants montrent les résultats de Scilab. Certains de ceux-ci sont présentés avec les données normalisées, pour facilité de visualisation, d'autres sont « dénormalisés », afin de permettre une comparaison avec les résultats précédents. Nous avons d'abord, sur le graphique 4.2 page suivante, les points avec leur étiquette de catégorie, et les droites obtenues par ACP et par régression linéaire pour chacune des deux catégories. Ensuite, sur le graphique 4.3 page 35, les points n'ont plus leur étiquette de couleur. De plus, on peut voir les moyennes et les ellipses d'isoprobabilité à deux écarts-type. Ces caractéristiques de lois sont celles des deux lois utilisées pour l'initialisation. Ce graphique est effectué en utilisant les données normalisées, elles sont donc concentrées sur le carré $[-1, 1] \times [-1, 1]$

Ensuite, le graphique 4.4 page 36 présente ces mêmes données, mais les caractéristiques représentées sont celles obtenues à la fin de l'algorithme EM. A nouveau, les données sont nor-

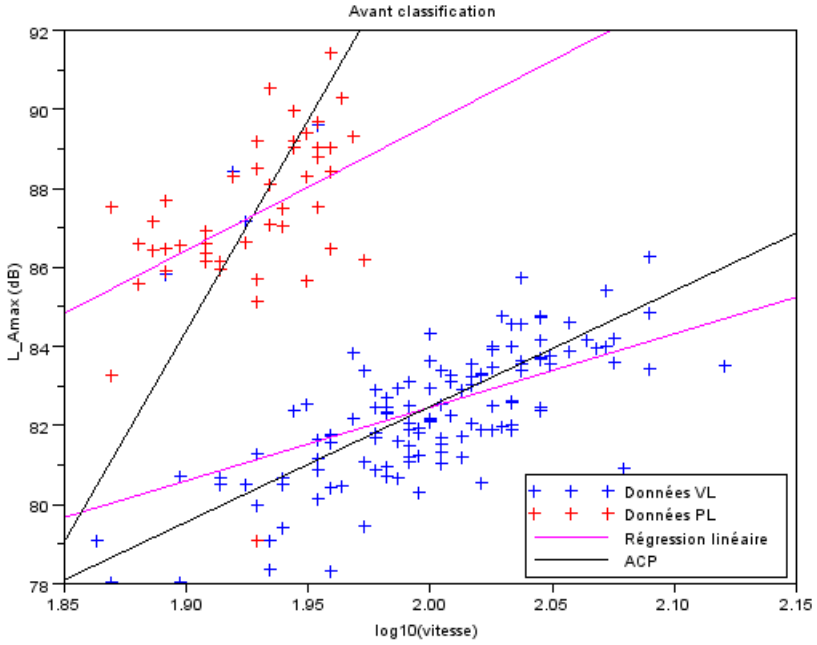


FIGURE 4.2 – Avant l'algorithme - droites ACP

malisées. On peut remarquer les changements dans les ellipses et dans les points représentant la moyenne.

Enfin, le graphique 4.5 page 37 représente les deux nouvelles catégories obtenues. Les nouvelles droites issues de l'ACP et de la régression linéaire sont également représentées pour les deux nouvelles classes. On voit que les cinq points posant problème, qui avaient été remarqués au début de ce chapitre, sont passés dans la classe de véhicules opposée. L'algorithme a donc effectué ce que l'on souhaitait.

On remarque que pour les VL, la droite issue de l'ACP ne change pas beaucoup avec la classification. Par contre, pour les PL, la droite se rapproche de ce que l'on attendait. Ces constatations sont les mêmes concernant la régression linéaire.

Quant à l'écriture, `Scilab` affiche aussi les points ayant changé de catégorie. On trouvera ci-dessous ces affichages pour les données de Haguenau.

```
nombre de points dans la categorie VL : 128
nombre de points VL passes en PL : 4

nombre de points dans la categorie PL : 47
nombre de points PL passes en VL : 1

nombre total de points: 175
nombre total de points ayant change de categorie : 5

les points passes de VL a PL sont les points du vecteur vl d'indice:
33
```

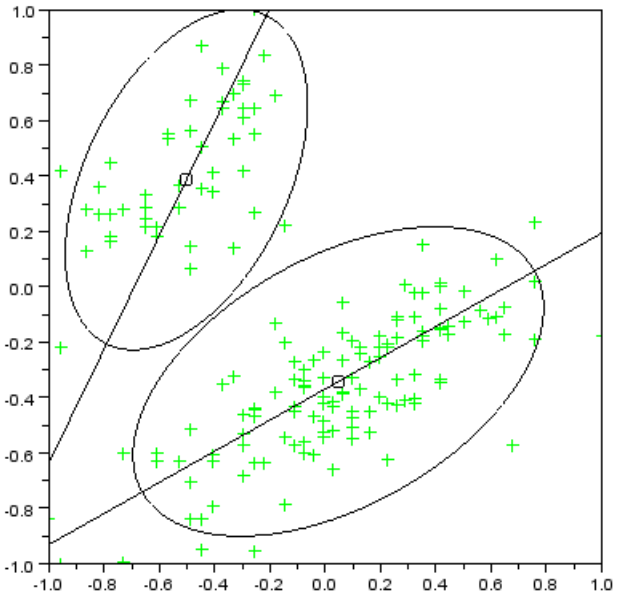


FIGURE 4.3 – Avant l'algorithme - initialisation avec deux lois normales

```
36
52
78


les points passes de PL a VL sont les points du vecteur pl d'indice:
38
```

Le texte ci-dessous montre le contenu de l'un des fichiers texte réalisés. Il correspond à la catégorie des « points incertains ».

```
donnees issues du fichier Haguenau2009

les indices utilises correspondent a ceux des donnees dans les fichiers de depart
ATTENTION : les vitesses sont sous la forme log10(vitesse)
la vitesse de reference n'a pas ete enlevee

indice   categorie depart     log10(vitesse)   L_{Amax}(dB)

33       VL                   1.954243         89.596583
36       VL                   1.924279         87.158541
52       VL                   1.919078         88.408002
78       VL                   1.892095         85.798173
38       PL                   1.929419         79.073639
```

Comparons maintenant les équations des droites : dans l'algorithme EM, après avoir dé-

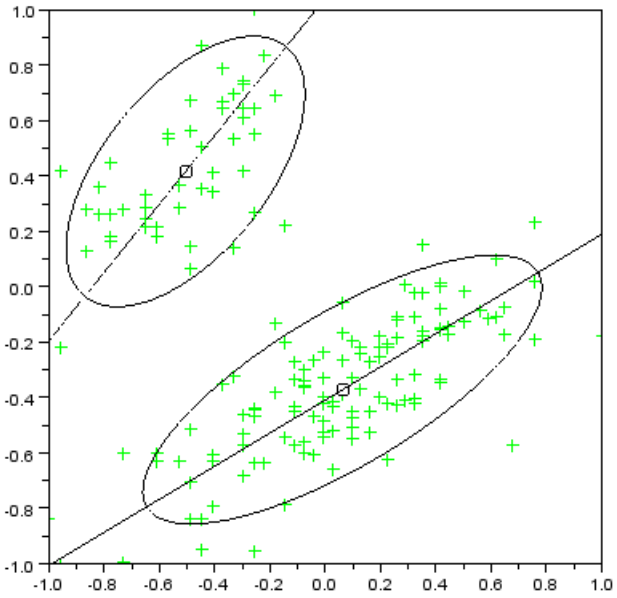


FIGURE 4.4 – Après l'algorithme - représentation des deux lois normales obtenues

normalisé les données et après avoir adapté les équations des droites, on obtient une droite d'ACP dont l'équation est de la forme :

$$L_{Amax} = a \times \log_{10}(vitesse) + b \tag{4.4}$$

Or on aimerait :

$$L_{Amax} = a' \times \log_{10}\left(\frac{vitesse}{v_{ref}}\right) + b' \tag{4.5}$$

En comparant ces deux équations, on remarque que la pente de la droite reste la même ($a' = a$), et on a directement :

$$b' = b + a \times \log_{10}(v_{ref}) \tag{4.6}$$

A titre d'exemple, on peut faire la comparaison avec la droite d'ACP pour *Haguenau2009VL* et la droite pour la catégorie VL avant l'algorithme EM : ce sont les mêmes.

Avant l'algorithme, l'équation de la droite d'ACP pour les VL est :

$$L_{Amax} = 29.3 \times \log_{10}(vitesse) + 24.0 \tag{4.7}$$

En faisant la transformation ci-dessus, on obtient l'équation :

$$L_{Amax} = 29.3 \times \log_{10}\left(\frac{vitesse}{90}\right) + 81.2 \tag{4.8}$$

On retrouve bien l'équation de la droite obtenue dans le chapitre 3 page 20 sur l'ACP.

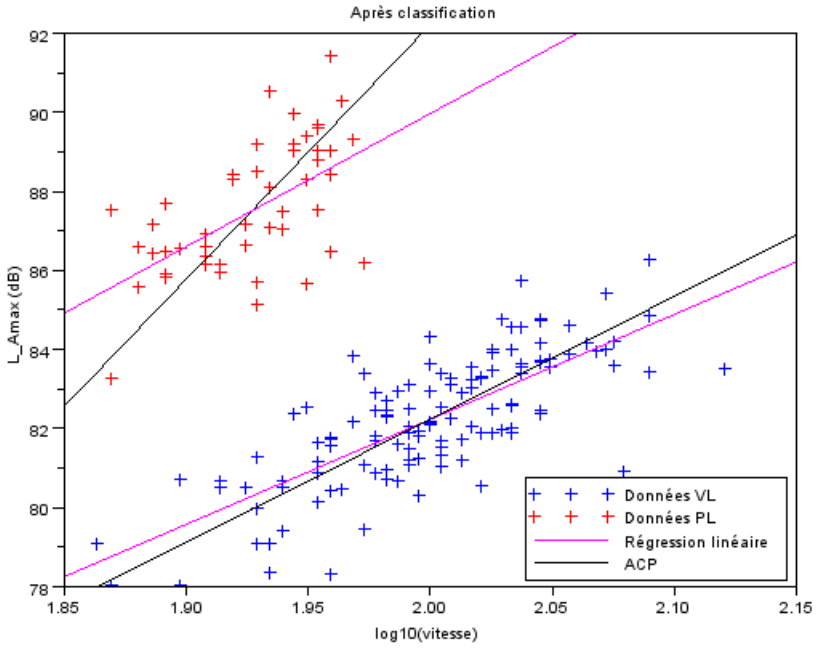


FIGURE 4.5 – Classification après l'algorithme

	Haguenau2009VL		Haguenau2009PL	
	pente	origine	pente	origine
ACP - avant EM	29.3	81.2	106.9	84.7
ACP - après EM	31.1	80.8	64.4	86.0

TABLE 4.2 – Comparaisons - paramètres de droites

Le tableau 4.2 montre les équations des droites issues de l'ACP, avant et après l'application de l'algorithme EM, pour les données VL et PL du jeu *Haguenau2009*. Les conversions nécessaires ont été faites afin de pouvoir comparer.

On observe une grande différence dans la pente des droites comparées pour les données PL. Pour les véhicules légers, cette différence existe mais elle est moindre. Concernant l'ordonnée à l'origine, qui représente le niveau de bruit pour la vitesse de référence, on observe tout de même une différence d'environ 1.5 dB.

Il faut toutefois prendre garde à l'interprétation qui est faite lors de cette comparaison : on a changé des points de catégorie alors qu'on ne sait pas s'ils proviennent d'erreurs ou non.

Cet algorithme ne peut être appliqué aux motos, car une seule catégorie est utilisée (les sens de circulation ne sont pas des catégories).

Concernant les données *Rothau2009*, on a une différence notable par rapport à Haguenau, où seulement quelques points isolés avaient changé de catégorie. Ici, une vingtaine de points

VL ont l'air de correspondre aux points PL, comme on peut le voir sur le graphique 4.6.

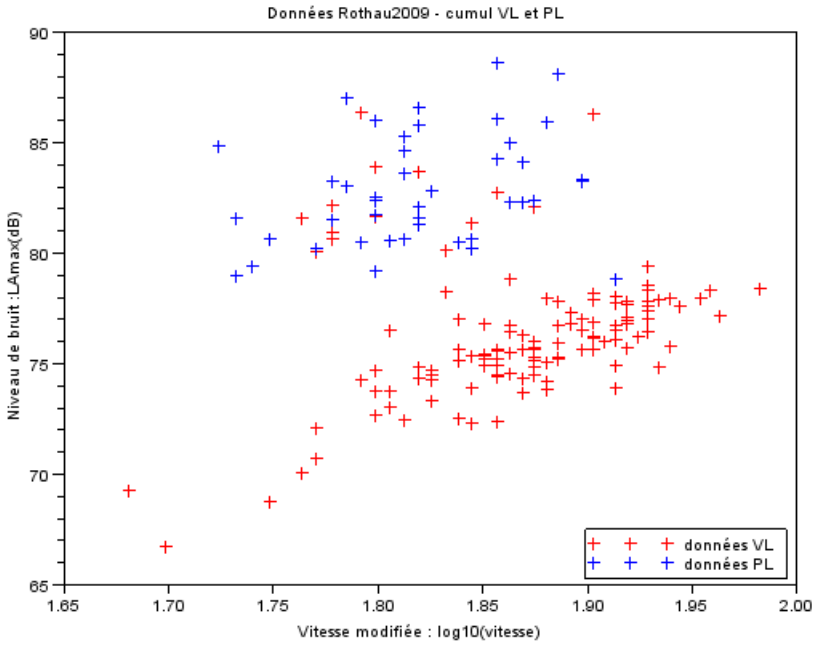


FIGURE 4.6 – Rothau2009 : Superposition de données VL et PL

Si on applique l'algorithme EM à ces données, cela fonctionne comme on « l'attendait » : cette vingtaine de points est classée PL. Un point PL change également de catégorie. On passe de 43 points PL à 59. Cela rend donc les estimations complètement différentes. L'interprétation s'avère encore plus difficile dans ce cas.

4.6 Nouvelles estimations de droites

Le chapitre 6 page 44, en présentant la procédure finale qui sera utilisée, montre aussi les résultats des différentes estimations faites après la classification.

Pour les VL, les estimations sont très proches les unes des autres. Par contre, pour les PL, cela est plus difficile. Certaines estimations sont encore assez différentes. On trouvera plus de détails sur cette analyse dans le chapitre suivant. De plus, un critère de comparaison des différents résultats sera présenté dans le chapitre 7 page 49.

4.7 Perspectives et questions

Les résultats obtenus par l'application de cet algorithme sont bons et conformes à ce que l'on attendait. Cette méthode va être utilisée dans la procédure d'analyse des données qui est présentée dans le chapitre suivant, dans le but d'être intégrée au logiciel `dBEuler`.

Le but de l'utilisation de l'algorithme EM n'est pas de supprimer le travail de l'opérateur : celui-ci devra toujours enregistrer les catégories des véhicules mesurés, et transposer ceci dans le logiciel. De plus, cela donne une très bonne initialisation à l'algorithme. Mais les résultats obtenus pourraient servir à rendre attentif ce même opérateur à de potentielles erreurs. Les points qui ont « changé » de catégorie seront à vérifier (par exemple réécouter l'enregistrement audio correspondant à ces véhicules). Ces vérifications permettront alors de minimiser les erreurs de transcription.

Une perspective possible est de former deux nouvelles classes de PL et de VL à la fin de l'algorithme, conformément aux changements qui ont été observés. On peut alors faire des estimations robustes sur ces deux classes. Les conclusions de ces estimations seront présentées dans le chapitre 6 page 44.

Par contre, un problème reste toujours présent : un véhicule léger qui fait beaucoup de bruit peut-il, lors de l'estimation, être assimilé à un poids lourd ? Autrement dit, il s'agit de savoir si les estimations qui ont été faites après l'application de l'algorithme EM ont un sens. La réponse à cette question n'est pas évidente et nécessiterait d'examiner aussi le spectre par bandes de tiers d'octave et les lois de vitesse associées.

Chapitre 5

Analyse par *bootstrap*

5.1 Utilisation du *bootstrap* pour ces données

La méthode du *bootstrap* peut s'utiliser de deux manières : la manière paramétrique et la manière non-paramétrique.

Rappelons l'utilisation de ces méthodes. La méthode paramétrique suppose de connaître la famille de lois à laquelle appartiennent les données. On estime, grâce à l'échantillon disponible, les paramètres de cette loi. Ensuite, on tire de manière aléatoire des échantillons suivant la loi trouvée. La statistique que l'on cherche à estimer peut alors être recalculée.

La méthode non-paramétrique consiste, quant à elle, à faire des tirages avec remise de l'échantillon de départ. A partir des nouveaux échantillons, on peut à nouveau calculer la valeur de la statistique cherchée.

Ces méthodes sont par ailleurs présentées dans l'annexe A.6 page 74.

La méthode paramétrique implique une hypothèse forte, à savoir de donner la famille de lois de l'échantillon. Or a priori, dans les données disponibles, nous n'avons pas d'information par rapport à cette famille de lois. Cette méthode ne sera donc pas utilisée.

La méthode non paramétrique a alors été testée. Le cas en dimension 1 est testé en annexe sur un jeu de données de référence. La généralisation a deux dimensions a été comparée à une implémentation de référence faute de jeu de données de référence. L'implémentation a donc été faite dans `Scilab`, mais a été vérifiée en utilisant la fonction `boot` issue du paquet `boot` du logiciel `R` et en comparant les résultats obtenus avec les deux logiciels.

La procédure a été la suivante : le nombre d'échantillons tiré est $N = 500$. Supposons que l'échantillon initial soit de taille n. On tire donc N échantillons de taille n depuis l'échantillon de départ. Pour chacun des N nouveaux échantillons, on calcule, en utilisant la M-estimation avec fonction de poids de Geman et McClure, les paramètres de la droite de régression. On obtient donc N valeurs de la pente et N valeurs de l'ordonnée à l'origine. La valeur finale choisie est, pour chaque paramètre, la médiane des N valeurs.

Un intervalle de confiance à 95% par la méthode des percentiles (cf. A.6 page 74) a également été calculé pour chacun des paramètres.

5.2 Résultats obtenus

En fait, avant d'utiliser la M-estimation sur les échantillons obtenus par *bootstrap*, j'ai d'abord effectué une simple régression linéaire sur ces échantillons. En effet, le *bootstrap* devait servir à éliminer le problème dû aux points aberrants. Si cela fonctionnait, une régression linéaire suffirait à estimer les paramètres. La figure 5.1 montre le résultat obtenu sur les données de Haguenau. On voit que l'on retombe sur le même résultat que celui de la régression linéaire classique.

On a donc appliqué les M-estimateurs à ces échantillons issus du *bootstrap*. En effet, il est intéressant de continuer cette méthode de cette manière car cela permet d'obtenir des intervalles de confiance pour les estimations des paramètres.

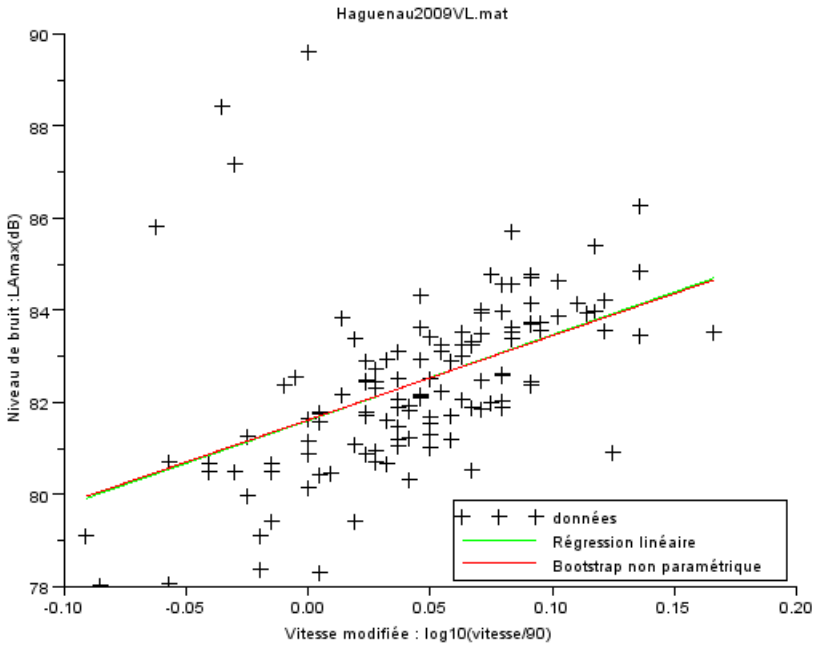


FIGURE 5.1 – Comparaison - droite issue de régression linéaire et droite issue du *bootstrap* non paramétrique, suivi de régression linéaire

Avec `Scilab` comme avec `R`, les résultats obtenus sont très proches des résultats obtenus par M-estimation avec l'échantillon initial, comme on peut le voir sur la figure 5.2 page suivante. Cela pouvait être prévisible : en effet, malgré les tirages qui ne donnent pas toujours les mêmes échantillons (certains points sont tirés plusieurs fois, d'autres ne sont pas tirés), on utilise toujours les mêmes données.

Comme les résultats de `Scilab` et de `R` sont similaires dans ce cas, on peut supposer que l'implémentation faite dans `Scilab` est correcte.

Dans la table 5.1 page 43, on peut voir les différentes estimations (M-estimation avec échantillon de départ, *bootstrap* avec chacun des logiciels).

De plus, la table 5.2 page 43 montre les intervalles de confiance à 95% obtenus pour

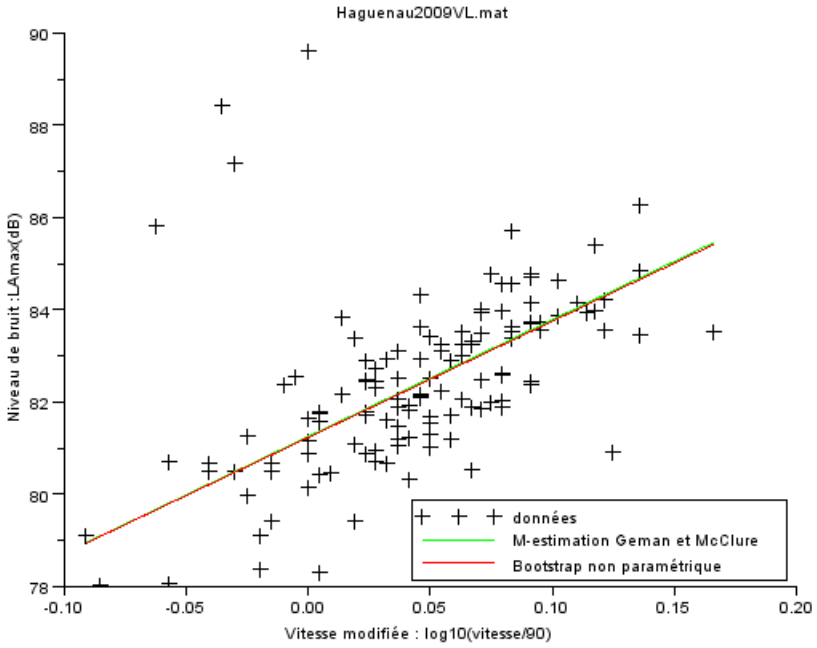


FIGURE 5.2 – Comparaison - droite issue de M-estimation et droite issue du *bootstrap* non paramétrique, suivi de M-estimation

chaque jeu de données. On observe que ceux-ci sont très larges pour la pente de la droite de régression. Ils le sont beaucoup moins pour l'ordonnée à l'origine.

On remarque à nouveau la proximité entre les intervalles obtenus avec les deux logiciels.

5.3 Conclusions

Des essais ont été fait avec le *bootstrap* classique non paramétrique. Les résultats observés ne sont pas très convaincants : ce sont les mêmes que ceux obtenus par estimation sur l'échantillon de départ.

De plus, les *outliers* posent de nouveau un problème. Lorsqu'on fait du *bootstrap* non paramétrique, donc des tirages avec remise, ces points aberrants ont la même probabilité d'être tirés que les points « normaux ». Les échantillons tirés auront donc tendance à contenir plus d'*outliers* que l'échantillon de départ et ne seront donc pas conformes à la réalité. Malgré l'estimation robuste, cela peut provoquer des changements importants dans les estimations des paramètres. Il suffit de voir les intervalles de confiance très larges, et qui sont construits à partir de valeurs réellement calculées.

Une généralisation du *bootstrap* pour les jeux de données contenant des *outliers* a été publiée dans [ZSB02]. Ce type de *bootstrap* robuste, appelé « *fast-bootstrap* », permet d'intégrer ces données aberrantes. Par contre, ce n'est pas adapté aux estimateurs utilisés ici, car l'article développe la méthode avec des MM-estimateurs et des S-estimateurs. Ceux-ci n'ont pas été

Données	Logiciel	Pente	Origine
Haguenau2009VL	`Scilab`	25.3	81.2
	`R`	25.2	81.3
Haguenau2009PL	`Scilab`	40.2	86.6
	`R`	41.0	86.6
Rothau2009VL	`Scilab`	30.3	78.0
	`R`	30.3	78.0
Rothau2009PL	`Scilab`	18.6	83.5
	`R`	18.3	83.6

TABLE 5.1 – Comparaisons - paramètres de droites

Données	Logiciel	Pente	Origine
Haguenau2009VL	`Scilab`	[20.6,32.7]	[80.7,81.5]
	`R`	[20.1,32.2]	[80.7,81.6]
Haguenau2009PL	`Scilab`	[1.7,60.1]	[85.8,87.2]
	`R`	[-2.1,59.2]	[85.9,87.3]
Rothau2009VL	`Scilab`	[23.6,34.5]	[77.5,78.5]
	`R`	[23.8,34.3]	[77.5,78.4]
Rothau2009PL	`Scilab`	[-7.7,43.9]	[81.5,86.7]
	`R`	[-6.67,43.9]	[82.3,86.9]

TABLE 5.2 – Intervalles de confiance

étudiés dans le cadre du stage et ne sont donc pas à notre disposition.

Il pourrait être intéressant de réfléchir à cette méthode, qui devrait permettre la détermination d'intervalles de confiance réalistes. Le *bootstrap* serait donc complémentaire des outils étudiés jusque là.

Chapitre 6

Procédure finale

Un code `Scilab` propre et commenté a été mis en place pour la fin du stage. Celui-ci peut avoir deux finalités. D'abord, il peut servir à continuer l'analyse qui a été débutée ici. C'est un outil de travail que l'on peut facilement appliquer à d'autres jeux de données. D'autre part, il peut être utilisé comme base pour l'intégration de l'estimation robuste dans le logiciel `dBEuler`. Il faudra toutefois faire un choix quant à la méthode d'estimation qui sera alors utilisée.

Pour ce faire, plusieurs fichiers de fonctions ont été créés, ainsi qu'un fichier nommé `base.sce` qui contient l'algorithme en lui-même. Comme cela a déjà été précisé, les scripts de ces fichiers sont disponibles sous forme de fichiers informatiques.

Les exemples dans ce chapitre seront à nouveau présentés pour les données *Haguenau2009*.

6.1 Déroulement de la procédure

La procédure est appliquée à la fois sur les données VL et PL, car elle inclut l'application de l'algorithme EM qui nécessite les deux fichiers.

Les résultats (figures, tables de données au format `Matlab`, fichiers avec extension `.txt`) sont mis dans un dossier à part. En fait, un dossier ayant pour nom celui du jeu de données (ici `Haguenau2009`) est créé.

6.1.1 Analyse avant classification

Les chapitres précédents ont présenté différentes manières d'analyser les données, grâce à des méthodes de statistique robuste et non-robuste. Toutes ces méthodes sont utilisées dans l'algorithme présenté ici.

Pour les calculs, les données sont normalisées, conformément à ce qui a été expliqué dans le chapitre 3 page 20. Les calculs des estimations des paramètres des droites, pour les données VL et PL, sont effectuées par les méthodes suivantes :

- Régression linéaire
- M-estimation avec poids de Geman et McClure
- ACP
- ACP robuste avec poids de Geman et McClure

Une figure est enregistrée dans le dossier de résultats, avec toutes les droites correspondantes pour les deux catégories. Lorsqu'elle est disponible, la droite de régression experte,

donc celle trouvée par les opérateurs, est également dessinée. Enfin, deux segments verticaux indiquent, pour chaque catégorie, où se trouve l'origine (elle vaut $\log_{10}(80)$ pour les PL et $\log_{10}(90)$ pour les VL).

La figure est faite sur les données dénormalisées afin de permettre une comparaison avec les figures précédentes. La figure 6.1 présente ce graphique. Dans cet exemple, le nom de ce fichier est `avant-Haguenau2009.png`.

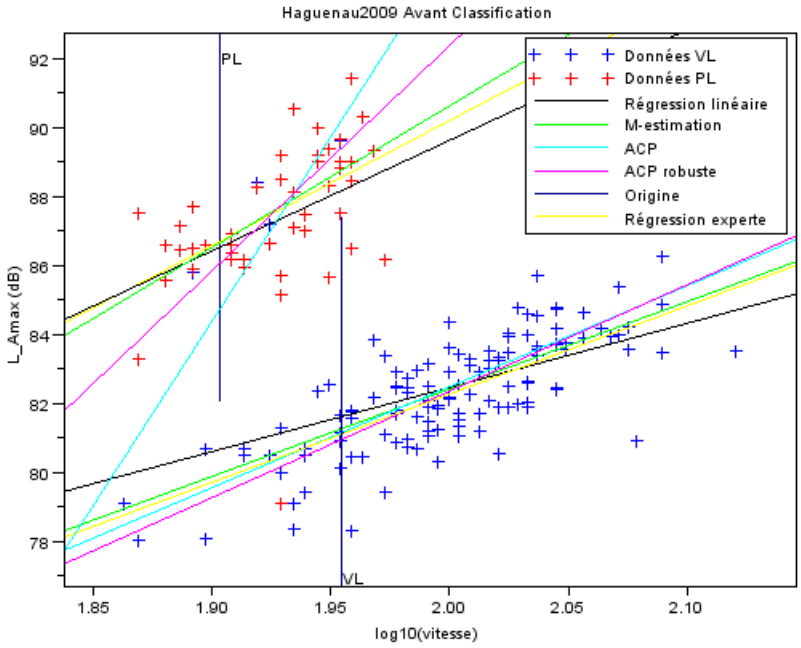


FIGURE 6.1 – Haguenau2009 : Résultats avant classification

Les tables avec l'extension `.mat` (qui sont des fichiers binaires, pour le logiciel `Matlab`, version 6.5) contenant les données des deux classes de départ sont également mises dans le dossier de résultats. Elles s'appellent ici `avant-Haguenau2009VL.mat` et `avant-Haguenau2009PL.mat`

En sortie, on a des matrices contenant les paramètres des droites calculées. Elles sont également enregistrées avec l'extension `.mat` pour une utilisation ultérieure. Les fichiers créés s'appellent `avant-Haguenau2009-vl-par.mat` et `avant-Haguenau2009-pl-par.mat` (un pour chaque catégorie de véhicules).

Cette sortie est présentée ci-dessous.

```
-----------------------------------------------------------------
----------------------AVANT CLASSIFICATION-----------------------
-----------------------------------------------------------------

Parametres des droites - avant classification - donnees VL
sous la forme : L_Amax=a*log10(vitesse/vref)+b

!Methode              Pente      Ordonnee a l'origine  !
!                                                      !
```

```
!Regression lineaire  18.594314  81.613542            !
!                                                     !
!M-estimation         25.33296   81.259035            !
!                                                     !
!ACP                  29.261733  81.147195            !
!                                                     !
!ACP robuste          30.859072  80.945023            !
!                                                     !
!Regression experte   25.6       81.1                 !
```

```
Parametres des droites - avant classification - donnees PL
sous la forme : L_Amax=a*log10(vitesse/vref)+b

!Methode              Pente      Ordonnee a l'origine  !
!                                                     !
!Regression lineaire  32.028835  86.531816            !
!                                                     !
!M-estimation         41.073083  86.653913            !
!                                                     !
!ACP                  106.8928   84.724991            !
!                                                     !
!ACP robuste          65.274522  86.05565             !
!                                                     !
!Regression experte   36.2       86.7                 !
```

6.1.2 Classification

L'algorithme EM, tel qu'il a été présenté dans le chapitre 4 page 29 est appliqué. Lorsque la classification sera intégrée dans le logiciel `dBEuler`, l'opérateur pourra décider d'accepter ou de refuser la classification proposée par cet algorithme, selon les vérifications qu'il aura faites.

Dans le dossier de résultats seront rangés les trois fichiers avec extension `.txt` contenant respectivement les points VL dont on est certain, les points PL dont on est certain et les points incertains (ceux que l'algorithme change de catégorie). Ces fichiers s'appellent `apres-Haguenau2009-donnees-vl.txt`, `apres-Haguenau2009-donnees-pl.txt` et `apres-Haguenau-2009-incertains.txt`.

De plus, la même sortie que celle présentée page 34 est effectuée ici.

Enfin, les nouvelles tables d'extension `.mat`, contenant les données selon la proposition de l'algorithme, sont exportées dans le dossier de résultats. Ces fichiers sont nommés `apres-Haguenau2009-classe-vl.mat` et `apres-Haguenau2009-classe-pl.mat`.

6.1.3 Analyse après classification

La même analyse qui a été menée dans la section 6.1.1 page 44 est reproduite ici, en supposant que la classification proposée par l'algorithme EM ait été acceptée. On utilise donc les nouvelles données VL et PL, contenues dans les tables présentées dans la section précédente.

Le fichier contenant la figure avec toutes les droites, nommé `apres-Haguenau2009.png` est présenté dans la figure 6.2 page suivante.

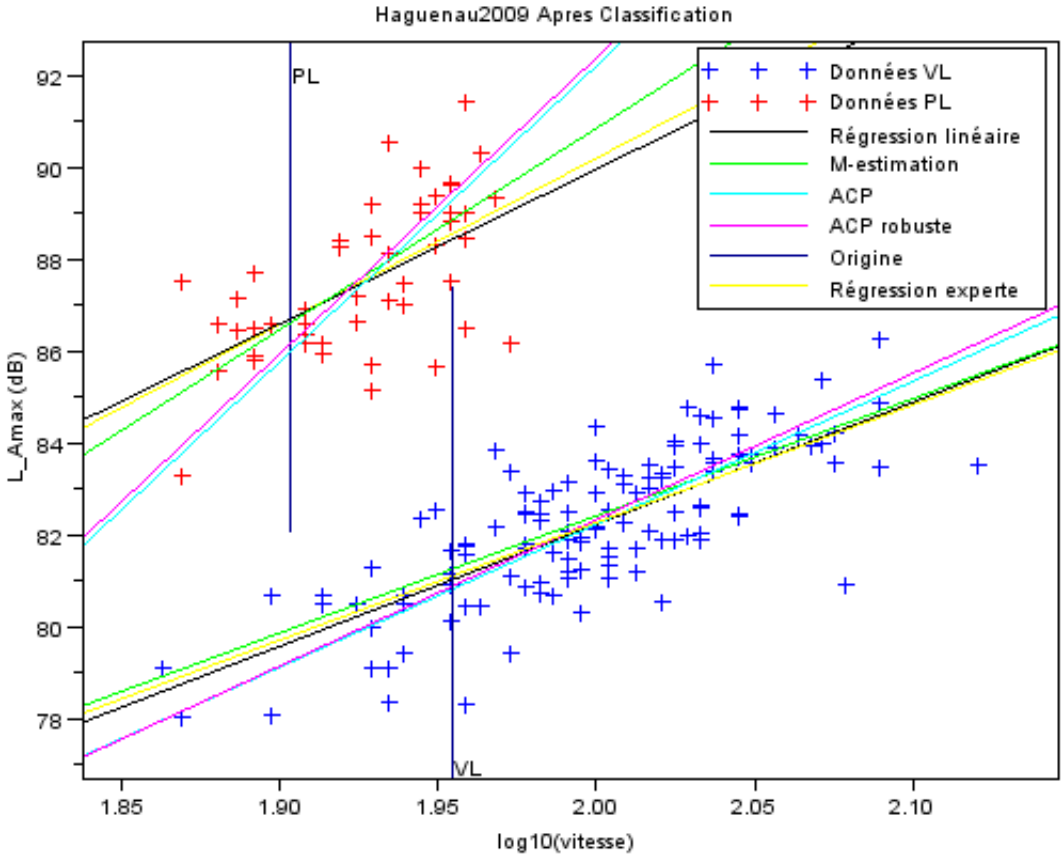


FIGURE 6.2 – Haguenau2009 : Résultats après classification

Enfin, la sortie de la procédure est semblable à celle d'avant la classification. Les valeurs des différentes estimations sont enregistrées dans `apres-Haguenau2009-vl-par.mat` et `apres-Haguenau2009-pl-par.mat`. Voici cette sortie :

```
-----------------------------------------------------------------
----------------------APRES CLASSIFICATION----------------------
-----------------------------------------------------------------

Parametres des droites - apres classification - donnees VL
sous la forme : L_Amax=a*log10(vitesse/vref)+b

!Methode              Pente      Ordonnee a l'origine  !
!                                                      !
!Regression lineaire  26.561916  81.018716            !
!                                                      !
!M-estimation         25.490041  81.25147             !
!                                                      !
!ACP                  31.113095  80.811247            !
!                                                      !
```

```
!ACP robuste          31.879096  80.877334            !
!                                                     !
!Regression experte   25.6       81.1                 !
```

```
Parametres des droites - apres classification - donnees PL
sous la forme : L_Amax=a*log10(vitesse/vref)+b
```

```
!Methode              Pente      Ordonnee a l'origine  !
!                                                     !
!Regression lineaire  33.680284  86.705837            !
!                                                     !
!M-estimation         43.852468  86.612358            !
!                                                     !
!ACP                  64.410141  85.97733             !
!                                                     !
!ACP robuste          64.416621  86.153504            !
!                                                     !
!Regression experte   36.2       86.7                 !
```

6.2 Conclusion

La procédure présentée dans ce chapitre permet d'analyser des données issues des campagnes de mesure de bruit de roulement. Plusieurs méthodes d'estimation des paramètres de la droite cherchée sont utilisées.

Il reste une chose à faire : déterminer quelle méthode est la meilleure, et donc quelles valeurs sont à garder. Ceci doit être fait d'un point de vue acoustique. La régression experte faite par un acousticien peut définir la référence.

Le chapitre 7 page suivante explore une approche alternative. Il s'agit en fait de calculer une certaine grandeur pour chaque estimation. Ces grandeurs seront ensuite comparées.

Chapitre 7

Calcul du niveau sonore équivalent (L_{eq})

Cette partie a été rédigée grâce aux informations trouvées dans le manuel [BHL+09]. Le code correspondant à cette partie existe également sous forme de fichiers informatiques. Il a été écrit avec `Scilab`.

7.1 Introduction

Un certain nombre de méthodes d'estimation ont été étudiées. Il reste maintenant une question qui est de savoir laquelle choisir. Cela doit être décidé d'un point de vue acoustique, car c'est au niveau des résultats acoustiques que l'on peut juger une méthode.

Au-delà de la classification des revêtements de chaussée, la finalité de la mesure de bruit de roulement est la prévision des niveaux sonores auxquels les riverains sont exposés. Dans ce cas, la grandeur la plus couramment utilisée est le niveau sonore équivalent L_{eq} sur une durée longue (exprimé en décibels), plutôt que le L_{Amax}.

Cette grandeur est calculée pour chaque méthode d'estimation et est comparée avec celle obtenue avec la régression experte.

Les détails du calcul de cette mesure se trouvent dans le paragraphe suivant. Toutefois, on peut dire qu'elle se calcule à partir de couples de la forme (vitesse, L_{Amax}).

Avec les méthodes d'estimation, on peut obtenir en fait différents jeux de données. Il y a les données « brutes », qui sont celles sur lesquelles on a travaillé. Ensuite, pour chaque méthode, on obtient une équation de droite. A partir de celle-ci, on peut définir un nouveau jeu de données sous forme de couples (vitesse, L_{Amax}) en calculant le L_{Amax} pour chaque vitesse grâce à la nouvelle équation de droite.

7.2 Calcul du niveau sonore moyen, $\overline{L_{eq}}$

Nous avons à notre disposition deux vecteurs de même taille : X contient la vitesse et Y le L_{Amax}. Soit n la taille de ces deux vecteurs. La première chose à faire est de passer du L_{Amax} au niveau de puissance sonore L_W. Une approximation à 0.01dB nous donne la relation :

$$L_{Amax} = L_W - 25.59 \tag{7.1}$$

Les opérations suivantes sont faites pour chaque couple (vitesse, L_W).

Il s'agit de calculer un certain nombre de valeurs de la fonction $L_p(t)$ qui est l'évolution du niveau sonore en un point en fonction du temps. La vitesse et la distance du véhicule au récepteur sont également pris en compte dans cette fonction. L'évaluation suppose un véhicule se déplaçant en ligne droite à vitesse constante. On suppose que cela est vrai, étant donné que la valeur est calculée pour un court laps de temps. En fait, la fonction $L_p(t)$ a une courbe en cloche, et il s'agit de calculer $L_p(t)$ sur un intervalle de temps centré sur le maximum de la fonction. Il faut aussi connaître l'abscisse des deux points se trouvant à l'ordonnée $(maximum - 10)$dB. La figure 7.1 montre un exemple de la détermination des abscisses, notées $t_{1,i}$ et $t_{2,i}$.

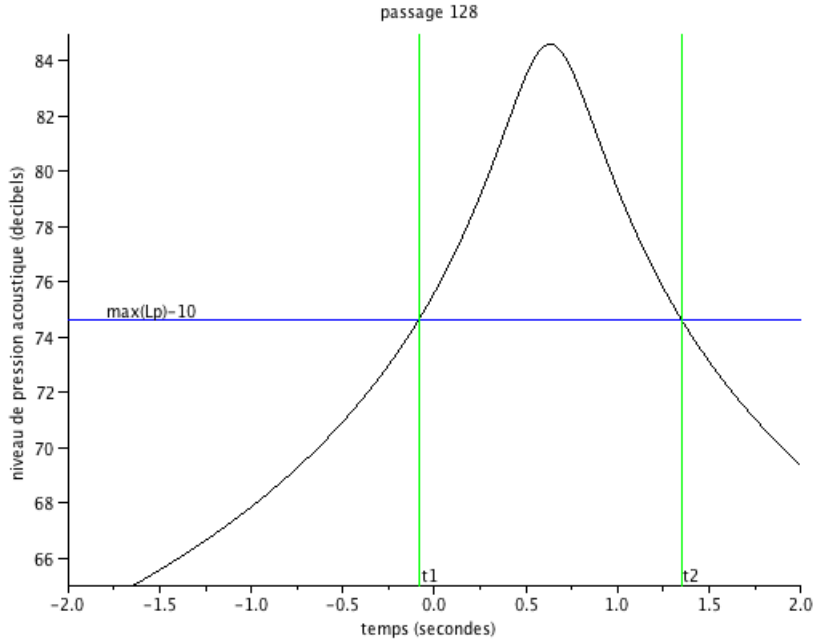


FIGURE 7.1 – Evolution du niveau sonore - Détermination des abscisses $t_{1,128}$ et $t_{2,128}$

Ayant les abscisses de ces deux points, on calcule $T_i = |t_{1,i} - t_{2,i}|$ qui est la durée utilisée pour le couple d'observations i. Cette grandeur est exprimée en secondes.

Enfin, on peut calculer le niveau sonore équivalent pour le passage de véhicule i, noté $L_{eq,i}$, toujours pour chaque point. Celui-ci vaut :

$$L_{eq,i} = 10 \log_{10} \left(\frac{\Delta t}{T_i} \sum_{t_{1,i}}^{t_{2,i}} 10^{\frac{L_p(t)}{10}} \right) \tag{7.2}$$

En fait, la somme est calculée pour un certain nombre de valeurs de $L_p(t)$. Δt représente le pas d'échantillonnage, en seceondes, de $L_p(t)$. Celui-ci est constant et peut être par exemple de 0.01s.

Finalement, après avoir calculé le niveau sonore équivalent pour chaque passage, on obtient une nouvelle matrice ayant deux colonnes. La première contient les valeurs de $L_{eq,i}$, la deuxième contient les durées T_i. Chaque ligne correspond à un passage de véhicule. Pour obtenir la valeur moyenne du niveau sonore équivalent, notée $\overline{L_{eq}}$, il suffit de faire le calcul suivant :

$$\overline{L_{eq}} = 10 \log_{10} \left(\frac{\sum_{i=1}^{n} T_i \times 10^{\frac{L_{eq,i}}{10}}}{\sum_{i=1}^{n} T_i} \right) \tag{7.3}$$

Ce calcul fournit une première approximation du niveau sonore équivalent au point de mesure sur la durée de la campagne de mesures qui s'étale sur quelques heures.

7.3 Résultats obtenus

Le calcul du $\overline{L_{eq}}$ a été fait pour chaque méthode d'estimation de droite étudiée, et ce, pour chaque fichier de données disponible, avant et après la classification. Le tableau 7.1 donne les résultats obtenus pour les données *Haguenau2009VL*.

Méthode	Données brutes	Rég.linéaire	M-est.	ACP	ACP rob.
Avant classification	79.02	78.63	78.62	78.71	78.59
Après classification	78.59	78.47	78.64	78.51	78.62

TABLE 7.1 – Calcul du $\overline{L_{eq}}$ (exprimé en dB)- Résultats obtenus pour *Haguenau2009VL*

La comparaison de ces différentes valeurs se fait par rapport à une valeur de référence. C'est la valeur du $\overline{L_{eq}}$ pour la régression experte. Dans le cas de *Haguenau2009VL*, cette valeur vaut : 78.47 dB.

On remarque que les valeurs obtenues sont très proches les unes des autres. En effet, pour ces données, les valeurs tiennent en environ 0.5dB, ce qui est très peu. Il est donc difficile de déterminer une méthode d'estimation qui serait meilleure que les autres.

Il est vrai que les droites correspondant aux véhicules légers étaient assez proches les unes des autres. Mais quand on applique le même algorithme à d'autres données, comme par exemple les poids lourds pour lesquels les droites étaient plus éloignées, on arrive à la même conclusion : il n'y a pas plus d'un décibel de différence entre les différentes valeurs obtenues.

7.4 Conclusion

Comme cela a été mentionné dans le dernier paragraphe, les résultats obtenus par cette méthode ne sont pas très probants. Les valeurs obtenues pour le critère $\overline{L_{eq}}$ ne sont pas suffisamment discriminantes.

Lorsqu'on regarde les droites issues de différentes estimations, on peut remarquer que les valeurs du L_{Amax} pour la vitesse de référence sont toujours très proches les unes des autres, quelle que soit la méthode retenue. Par contre, pour certains projets, on pourrait avoir besoin de connaître une valeur de L_{Amax} pour une vitesse éloignée de la vitesse de référence, par exemple pour $60km/h$. Dans ces cas-là, on sait qu'il y a des différences notoires

entre les droites, et qu'il y a donc de grandes différences entre les valeurs de L_{Amax} selon les estimations.

C'est donc pour ces situations qu'il est important de choisir une méthode. Le $\overline{L_{eq}}$ s'avère peu pertinent. La question est donc encore ouverte pour choisir l'une des méthodes d'estimation étudiées.

Chapitre 8

Conclusions et perspectives

Différentes méthodes d'estimation robuste visant à pallier au problème des points aberrants lors de la mesure de bruit de roulement au passage ont été étudiées durant le stage. Concernant les M-estimateurs, il s'est avéré que les meilleurs résultats sont obtenus avec la fonction de poids SEF où $\alpha = -0.5$ et celle de Geman et McClure. Par contre, l'ACP et l'ACP robuste semblent ne pas convenir. En effet, il a fallu effectuer une normalisation des données afin de ne pas avantager l'un ou l'autre des axes. Malgré cette modification, les droites obtenues ne sont pas satisfaisantes.

La classification non supervisée, qui a été appliquée avec l'algorithme EM, a également porté ses fruits. Cela pourra permettre d'assister l'opérateur dans le dépouillement des campagnes de mesure. Après intégration de cette méthode dans `dBEuler`, l'opérateur pourrait s'appuyer sur la classification proposée par l'algorithme pour détecter d'éventuelles erreurs ou éliminer certains passages.

La méthode du *bootstrap* a également été testée. Nous avons remarqué que les estimations obtenues par cette méthode n'apportent rien de plus que celles déjà existantes. Malgré tout, il existe d'autres manières d'appliquer le *bootstrap* pour calculer des statistiques robustes, mais elles n'ont pas été étudiées. De plus, le *bootstrap* permet de construire facilement des intervalles de confiance pour les paramètres que l'on cherche à estimer. Cela donne une nouvelle information que l'on n'avait pas avec les autres méthodes d'estimation et pourrait donc être exploité.

Grâce à ces différentes méthodes, un algorithme général a été mis en place. Celui-ci permet de faire toutes les estimations, avant et après classification. Dans ce cas, les données concernant les poids lourds et les véhicules légers pour une même campagne peuvent également être traités ensemble.

Au final, nous nous retrouvons donc avec plusieurs estimations de droite. Le problème est maintenant d'en choisir une, et cela grâce à l'estimation de référence qui est la régression experte. Une solution possible est le calcul du niveau sonore équivalent $\overline{L_{eq}}$. Malheureusement, l'approche développée n'est pas suffisamment discriminante. Il reste donc à réfléchir à une autre méthode permettant de faire ce choix.

Lorsque le choix d'une méthode aura été fait, il restera à intégrer la procédure au logiciel `dBEuler`. Cela permettra de faire l'analyse en traitant les points aberrants d'une meilleure manière qu'actuellement, où ils sont supprimés manuellement par jugement de l'opérateur.

Annexes

Annexe A

Outils mathématiques utilisés

Les annexes concernant les M-estimateurs, la théorie semi-quadratique, l'estimation de l'échelle et l'ACP ont été écrites grâce au livre [Mar06] et à un cours de Pierre Charbonnier, [Cha11b], effectué dans le cadre du Master IRIV (Images, Robotique et Ingénierie pour le Vivant) de l'ENSPS (Ecole Nationale Supérieure de Physique de Strasbourg).

Quant à l'algorithme EM, les livres ayant servi à rédiger cette partie sont [DHS01], [Bis06] et [Bis95]. Un autre cours destiné aux étudiants du Master IRIV a également été très utile. Il s'agit de [Cha11a].

A.1 M-estimateurs

A.1.1 Modèle de départ et régression linéaire

Soit le modèle linéaire $Y = R.f + \eta$, où Y est le vecteur des réponses, R la matrice des données, f le vecteur des paramètres inconnus et η le bruit associé au modèle. Nous avons n réalisations indépendantes $y_i = (R.f)_i + \eta_i$ de ce modèle. On note m la dimension de f. Ce modèle s'écrit alors de manière matricielle :

$$\begin{pmatrix} y_1 \\ \vdots \\ \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & r_{1,1} & \dots & r_{m,1} \\ \vdots & \vdots & & \vdots \\ 1 & r_{1,n} & \dots & r_{m,n} \end{pmatrix} \cdot \begin{pmatrix} f_0 \\ \vdots \\ f_m \end{pmatrix} + \begin{pmatrix} \eta_1 \\ \vdots \\ \\ \eta_n \end{pmatrix} \tag{A.1}$$

L'une des hypothèses utilisées pour faire des tests d'influence ou des intervalles de confiance sur les estimateurs des paramètres inconnus est la normalité des variables η_i, et donc des variables réponses Y_i, dont les observations sont les y_i. Soient μ et σ^2 l'espérance et la variance de cette loi normale. Cela signifie que la probabilité, pour une réponse, de se trouver hors de l'intervalle $[-3\sigma, 3\sigma]$ est très faible. Par exemple, dans le cas d'une loi normale $\mathcal{N}(0,1)$, cette probabilité est de 0.27%. Or dans de nombreuses données, il existe des points appelés « *outliers* », ou points aberrants, qui, justement, se trouvent « loin » des autres points, ou du moins ne semblent pas correspondre à ce qui était attendu. Cela met alors en cause l'hypothèse de normalité décrite ci-dessus. Une solution serait d'utiliser des lois de probabilité avec d'importantes queues de distribution. Cela nous amène à considérer les M-estimateurs.

Les M-estimateurs ont été introduits par Huber en 1964, et « M » vient de « Maximum-likelihood type estimators » pour estimateurs du type maximum de vraisemblance.

A.1.2 Estimateur du maximum de vraisemblance et modèle gaussien

Si g est la densité d'une certaine loi de probabilité de paramètre θ inconnu, et si nous avons n observations $x_1, \cdots, x_n$, l'estimateur du maximum de vraisemblance de θ est donné par :

$$\hat{\theta} = arg \max_{\theta \in \mathbb{R}} \prod_{i=1}^{n} g(x_i; \theta) \tag{A.2}$$

ou encore, en passant au logarithme et en prenant l'opposé :

$$\hat{\theta} = arg \min_{\theta \in \mathbb{R}} - \sum_{i=1}^{n} \log(g(x_i; \theta)) \tag{A.3}$$

D'autre part, dans le cas gaussien, c'est-à-dire lorsque les variables erreurs η_i sont distribuées indépendamment selon une loi normale $\mathcal{N}(0, \sigma^2)$, les variables aléatoires Y_i suivent également une loi normale, de paramètres $(R.f)_i$ et σ^2. On peut alors donner la loi du vecteur Y : $Y \sim \mathcal{N}_n(R.f, \sigma^2.I_n)$; c'est une loi normale à n dimensions. La densité de celle-ci est proportionnelle à :

$$\exp(-\frac{1}{2\sigma^2}||Y - R.f||^2) \tag{A.4}$$

On en déduit que l'estimateur du maximum de vraisemblance $\hat{f}^{MV}$ de f est donné par :

$$\hat{f}^{MV} = arg \max_{f \in \mathbb{R}^m} \exp(-\frac{1}{2\sigma^2}||Y - R.f||^2) \tag{A.5}$$

ou encore :

$$\hat{f}^{MV} = arg \min_{f \in \mathbb{R}^m} ||Y - R.f||^2 \tag{A.6}$$

Soient $r_i(f) = y_i - (R.f)_i$, $i = 1, \cdots, n$ les résidus associés au modèle. L'estimateur des moindres carrés $\hat{f}$ de f vérifie l'équation suivante :

$$\hat{f} = arg \min_{f \in \mathbb{R}^m} ||r(f)||^2 = arg \min_{f \in \mathbb{R}^m} \sum_{i=1}^{n} r_i(f)^2 \tag{A.7}$$

On remarque alors que dans le cas gaussien décrit ci-dessus, l'estimateur du maximum de vraisemblance de f correspond à l'estimateur des moindres carrés de f.

A.1.3 Construction du M-estimateur

Dans l'équation A.6 ou A.7, les résidus sont élevés au carré pour faire le calcul de l'estimateur. On remarque également que si, pour une observation i, le résidu associé est important, celui-ci peut être une valeur aberrante. L'idée des M-estimateurs est de réduire l'incidence de telles observations. Au lieu d'utiliser la fonction $x \longmapsto x^2$, on va prendre une autre fonction, notée ρ, et chercher à minimiser en f la quantité

$$e_M = \sum_{i=1}^{n} \rho(r_i(f)) \tag{A.8}$$

On remarque que l'estimateur des moindres carrés est un cas particulier de M-estimateur. Le M-estimateur est en fait une généralisation de l'estimateur du maximum de vraisemblance du cas gaussien.

Si les données n'étaient pas distribuées selon une loi normale, l'estimateur du maximum de vraisemblance n'aurait pas de bonnes propriétés. Les M-estimateurs aident à pallier ce problème.

Dans la littérature, il existe un certain nombre de fonctions, ou de familles de fonctions ρ, qui ont des propriétés différentes. Nous ne donnerons ici que quelques exemples. Le M-estimateur de Geman et McClure est donné par la fonction suivante, qui est représentée sur le graphique A.1 :

$$\rho_{GM}(r) = \frac{r^2}{1+r^2} \tag{A.9}$$

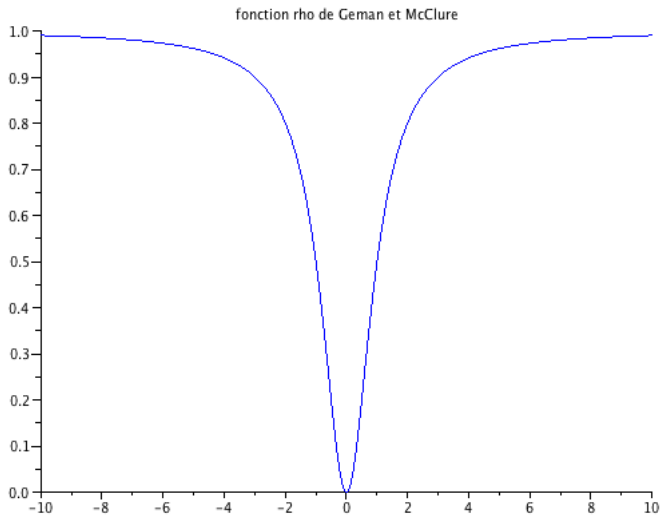


FIGURE A.1 – Geman et McClure

On peut également citer la « *Smooth Exponential Family* », ou SEF, paramétrée par une valeur α, dont quelques représentations sont données sur la figure A.2 page suivante :

$$\rho_\alpha(r) = \frac{1}{\alpha}((1+r^2)^\alpha - 1) \tag{A.10}$$

Par rapport à cette famille de fonctions, nous pouvons citer [TIC02].

Nous avons introduit les M-estimateurs afin de pallier le problème de la loi normale avec laquelle nous pouvons difficilement traiter les *outliers*.

Ce qu'on cherche avec les M-estimateurs est la robustesse de l'estimation, c'est-à-dire de trouver un estimateur qui ne perde pas ses propriétés ou ses qualités en présence d'*outliers*.

A.1.4 Robustesse du M-estimateur

Pour rendre l'estimation plus robuste, nous souhaitons donc moins prendre en compte les données aberrantes.

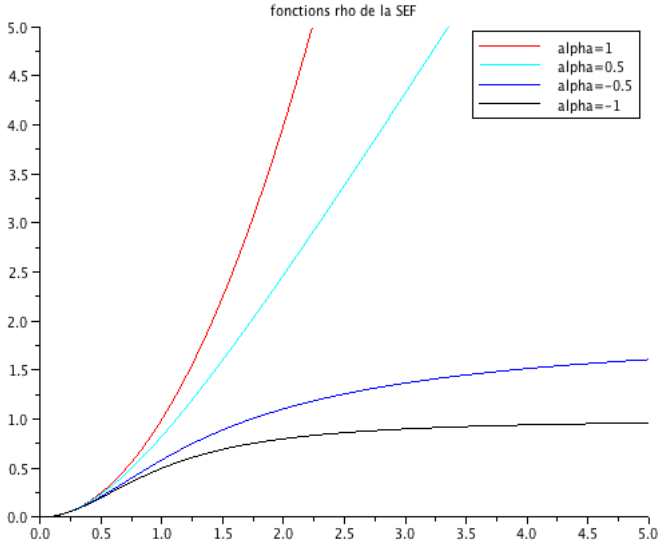


FIGURE A.2 – Smooth Exponential Family

Un estimateur est robuste si il est peu ou pas biaisé et si sa variance est faible. De plus, son « breakdown point » doit être le plus important possible ; c'est la proportion minimale d'*outliers* dans les données à partir de laquelle le biais de l'estimateur est infini.

On peut définir pour une donnée aberrante x_0 sa mesure de sensibilité sur un estimateur $\hat{\theta}$. Celle-ci est donnée par :

$$NSC(x_0) = \frac{\hat{\theta}_{n+1}(x_1, ..., x_n, x_0) - \hat{\theta}_n(x_1, ..., x_n)}{1/(n+1)} \tag{A.11}$$

où on calcule l'estimateur respectivement sur $n+1$ puis sur n données. Cette mesure calcule l'influence d'un *outlier* en fonction de sa position.

Il existe une version asymptotique de A.11, qui ne sera pas reproduite dans ce rapport mais qui est définie dans [Mar06]. Cette généralisation de la mesure de sensibilité s'appelle la *fonction d'influence* et est notée IF.

L'intérêt de cette fonction est que si elle est bornée, alors un point ne peut influencer l'estimateur que de manière bornée. Cela rendra alors l'estimation de f plus robuste.

Une propriété des M-estimateurs est donnée par : $IF(r) \propto \rho'(r)$. On aimerait donc que la dérivée $\rho'(r)$ soit bornée afin d'avoir un estimateur plus robuste.

On peut alors donner une explication de la non-robustesse de l'estimateur des moindres carrés : comme $\rho(r) = r^2$, on a $\rho'(r) = 2r$, qui n'est pas bornée ; on revient ensuite à la fonction d'influence ci-dessus : une observation dont le résidu est grand donnera une grande valeur de la fonction d'influence. On peut voir sur le graphique A.3 page suivante que si un seul point est aberrant (ici en bas à droite), l'équation de la droite de régression est faussée.

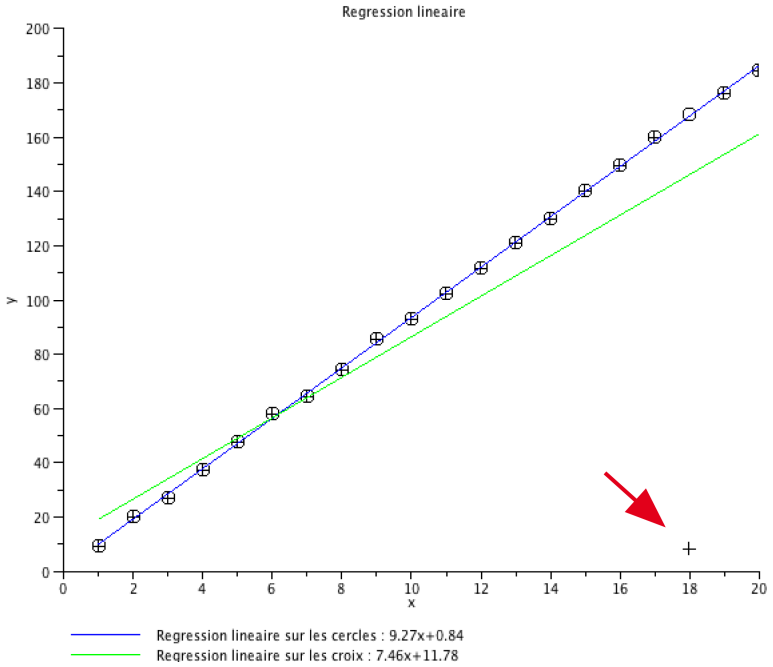


FIGURE A.3 – Droite des moindres carrés et point aberrant

A.1.5 Lien avec le cas quadratique et la régression linéaire

Revenons à la recherche du minimum de e_M, défini en A.8 page 56. Une propriété des fonctions ρ est qu'elles sont paires (un résidu doit être traité de la même façon, qu'il soit positif ou négatif). On peut donc restreindre la recherche à :

$$e_M = \sum_{i=1}^{n} \rho(|r_i(f)|) = \sum_{i=1}^{n} \rho(|r_i|) \tag{A.12}$$

Pour trouver le minimum de e_M, il s'agit alors de résoudre :

$$0 = \sum_{i=1}^{n} \rho'(|r_i|) sgn(r_i) \frac{\partial r_i}{\partial f_j} \quad \forall j = 0, \cdots, m \tag{A.13}$$

où

$$sgn(x) = \begin{cases} 1 & \text{si } x \geqslant 0 \\ -1 & \text{si } x < 0 \end{cases}$$

Or on a également $sgn(x) = x/|x|$. On obtient alors un ensemble d'équations :

$$0 = \sum_{i=1}^{n} \frac{\rho'(|r_i|)}{2|r_i|} 2r_i \frac{\partial r_i}{\partial f_j} \quad \forall j = 0, \cdots, m \tag{A.14}$$

Ces équations ressemblent fortement au cas quadratique (maximum de vraisemblance pour la loi normale) où il s'agit de résoudre

$$0 = \sum_{i=1}^{n} 2r_i \frac{\partial r_i}{\partial f_j} \quad \forall j = 0, \cdots, m \tag{A.15}$$

La différence résulte en un poids appliqué à chaque observation et qui vaut $\frac{\rho'(|r_i|)}{2|r_i|}$.

Ces valeurs de poids doivent vérifier quelques propriétés pour que l'estimateur résultant soit robuste. On aimerait en effet que les observations dont le résidu est petit soit traitées « normalement », donc comme pour les moindres carrés. Le poids pour ces observations devrait alors être proche de 1. Inversement, lorsqu'un résidu est grand, l'impact de l'observation correspondante devrait être limité ; on s'attend à un poids proche de 0. Enfin, pour assurer la cohérence de l'ensemble, la fonction de poids devrait être décroissante.

Finalement, pour assurer la robustesse d'un M-estimateur, il s'agit d'avoir une fonction ρ qui vérifie les conditions suivantes sur $[0; +\infty[$:

$$\begin{cases} \lim_{r\to 0} \frac{\rho'(r)}{2r} & = 1 \\ \lim_{r\to +\infty} \frac{\rho'(r)}{2r} & = 0 \\ \frac{\rho'(r)}{2r} & \text{strictement} \quad \text{décroissante} \end{cases} \tag{A.16}$$

A.1.6 Conclusion

Pour résumer ce chapitre, nous pouvons dire qu'un M-estimateur permet de faire une estimation robuste c'est-à-dire qu'il permet de faire une estimation avec de bonnes propriétés malgré la présence d'*outliers*.

Il est défini par une fonction ρ et résulte de la résolution de A.13 page précédente en f_j, pour tout j. De plus, la fonction ρ vérifie les propriétés présentées en A.16.

Lors de la minimisation, les poids utilisés sont supposés constants. Or ils dépendent des résidus, qui dépendent eux-mêmes de l'estimation des paramètres. On ne peut donc pas faire cette optimisation de prime abord. La section suivante donne une solution à ce problème.

A.2 Théorie semi-quadratique et Algorithme IRLS

La théorie semi-quadratique définit une famille de fonctions ρ telles que

$$\rho(r) = \inf_b \rho^*(r, b) \tag{A.17}$$

Lorsque b est fixé, ρ^* est quadratique par rapport à r ; lorsque r est fixé, ρ^* est convexe par rapport à b, d'où le nom de « semi-quadratique ». Nous pouvons ainsi donner deux théorèmes donnant une forme possible pour ρ^*. De plus, pour ces théorèmes, les seules hypothèses sont que ρ vérifie les conditions A.16 page précédente. Cela est donc applicable pour les M-estimateurs utilisés.

A.2.1 Premier théorème

Si ρ vérifie les conditions données en A.16 page précédente, alors il existe une fonction ξ, strictement convexe et décroissante, telle que :

$$\rho(r) = \inf_{0 \leqslant b \leqslant 1} (br^2 + \xi(b)) \tag{A.18}$$

De plus, pour $r \geqslant 0$ fixé, l'inf de ρ^* est atteint en un unique point b_r donné par :

$$b_r = \frac{\rho'(r)}{2r} \tag{A.19}$$

A.2.2 Second théorème

De la même façon que pour le premier théorème, si ρ vérifie les conditions données en A.16 page précédente, alors il existe une fonction ζ telle que :

$$\rho(r) = \inf_{b \in \mathbb{R}} ((b - r)^2 + \zeta(b)) \tag{A.20}$$

De plus, pour $r \geqslant 0$ fixé, l'inf de ρ^* est atteint en un unique point b_r donné par :

$$b_r = \left(1 - \frac{\rho'(r)}{2r}\right) r \tag{A.21}$$

A.2.3 Similitudes

Dans les deux cas, on observe que ρ^* s'écrit comme la somme de deux fonctions dont l'une est quadratique et fonction de b et de r, l'autre est uniquement fonction de b. De plus, ρ^* est convexe afin d'assurer l'unicité de b_r.

De plus, la valeur optimale (ici minimale) de b pour les fonctions obtenues a une forme simple dans laquelle on retrouve la fonction de poids $\frac{\rho'(r)}{2r}$ utilisée pour les M-estimateurs.

A.2.4 Algorithme IRLS

Comme nous l'avons vu dans l'annexe A.1 page 55, lors du calcul d'un M-estimateur, il s'agit de minimiser en r la quantité e_M définie dans l'équation A.12 page 59 et qui dépend de ρ. Il n'est pas évident de calculer cet estimateur directement. C'est pour cela que nous utilisons un algorithme itératif.

D'après la forme de ρ présentée dans les deux théorèmes ci-dessus, on peut voir qu'il s'agit en fait de minimiser ρ^* en r et en b. Nous allons maintenant définir un algorithme permettant de trouver ce minimum. Concrètement, il permet de trouver un estimateur du vecteur des paramètres f introduit au début de l'annexe A.1 page 55.

D'après la forme semi-quadratique de ρ^*, on peut trouver l'idée de cet algorithme : minimiser d'abord en r avec b fixé (il s'agit alors de résoudre un problème quadratique) ; dans un deuxième temps, on optimisera par rapport à b, en ayant fixé r (pour cela, il suffit de calculer la valeur b_r donnée dans les deux théorèmes).

Lorsqu'on est dans le cas du premier théorème, cela donne l'algorithme IRLS pour « Iterative Re-Weighted Least Squares » ou « Moindres Carrés Re-pondérés Itérés ».

Dans ce cas-là, il s'agit de minimiser en r et en b la quantité :

$$e^*(r,b) = \sum_{i=1}^{n}(b_i r_i^2 + \xi(b_i)) \tag{A.22}$$

Cela peut se réécrire :

$$e^*(r,b) = ||y - Rf||_B^2 + C \tag{A.23}$$

où y, R et f sont définis dans l'annexe A.1 page 55 sur les M-estimateurs et où B est une matrice diagonale définie par :

$$B = diag\left(\frac{\rho'(r_i)}{2r_i}\right)_{i=1\cdots n} \tag{A.24}$$

La constante C est en fait fonction des b_i qui ont été calculés.

On désigne par $||x||_B^2$ la norme pondérée associée à la matrice B. Cette norme se calcule par :

$$||x||_B^2 = x'Bx \tag{A.25}$$

où x' désigne le vecteur transposé de x. Si la longueur de x est n, alors la matrice B doit être de dimension $n \times n$.

En effet, le vecteur des résidus r est donné par $r = y - Rf$ et le premier théorème de la théorie semi-quadratique nous donne la forme des b_i.

Nous allons maintenant donner l'algorithme IRLS qui, comme son nom l'indique, est itératif (le numéro de l'itération est noté k). On définit une valeur ε qui détermine l'erreur que l'on accepte dans le calcul de l'estimateur de f.

- *étape 1 :* choisir un estimateur initial f^0 ; $k = 1$;
- *étape 2 :* calculer les résidus $r^k = y - Rf^{k-1}$ et calculer la matrice de pondération $B^k = diag\left(\frac{\rho'(r_i^k)}{2r_i^k}\right)_{i=1\cdots n}$;
- *étape 3 :* résoudre $R'B^kRf^k = R'B^ky$ qui donne en fait l'estimateur de Gauss-Markov dans le cas du modèle des moindres carrés pondérés ;
- *étape 4 :* Si $\frac{||f^k - f^{k-1}||^2}{||f^{k-1}||^2} > \varepsilon$, on incrémente k et on retourne à l'étape 2. Sinon on termine l'algorithme et f^k est l'estimateur obtenu par IRLS.

Il existe un algorithme similaire pour la forme de ρ^* du deuxième théorème, qui s'appelle « RSD » pour « *Residual Steepest Descent* ». En général, ces deux algorithmes convergent vers la solution cherchée : un minimum global (resp. local) de ρ si ρ est convexe (resp. non convexe).

A.3 Estimation de l'échelle

A.3.1 Problème

Les M-estimateurs ne sont pas invariants par rapport au paramètre d'échelle σ, qui est inconnu *a priori*. Par exemple, dans notre cas, ce résultat pourra dépendre des unités de mesure utilisées. Afin de pouvoir faire les estimations des paramètres de la droite de régression, il s'agit donc d'abord d'estimer cette échelle σ.

Revenons au calcul du maximum de vraisemblance des paramètres d'un vecteur aléatoire gaussien (cf. annexe A.1 page 55).

Soit Y un vecteur aléatoire gaussien de dimension n dont les composantes Y_i sont indépendantes et de loi $\mathcal{N}(\theta_i, \sigma^2)$.

La vraisemblance à maximiser, dans ce cas, en σ, est alors donnée par :

$$L(y_1, \cdots, y_n; \theta_i, \sigma) = \frac{1}{(\sqrt{2\pi\sigma^2})^n} exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \theta_i)^2\right) \tag{A.26}$$

En passant au logarithme, cela devient :

$$l(y_1, \cdots, y_n; \theta_i, \sigma) = -\frac{n}{2}\ln(2\pi) - n\ln(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \theta_i)^2 \tag{A.27}$$

Dans le cas de nos données, et en prenant les notations utilisées dans l'annexe A.1 page 55, nous aimerions donc trouver le minimum en σ de la quantité suivante :

$$e_\sigma = n\ln(\sigma) + \frac{1}{2\sigma^2}\sum_{i=1}^{n} r_i^2 \tag{A.28}$$

En calculant les M-estimateurs, nous avons introduit une fonction ρ qui est appliquée à la place de la fonction $x \longmapsto x^2$. En faisant ici cette transformation, on obtient :

$$e_{M,\sigma} = n\ln(\sigma) + \frac{1}{2}\sum_{i=1}^{n}\rho\left(\frac{r_i}{\sigma}\right) \tag{A.29}$$

Nous allons maintenant essayer d'estimer cette échelle σ grâce aux données.

A.3.2 Le cas de l'estimation L_1

Un autre problème qui se pose est que pour estimer σ, il faut estimer les autres paramètres du modèle (en effet, on a besoin du vecteur des résidus r qui vaut $Y - Rf$, où f est le vecteur des paramètres). Inversement, dans ce cas, pour estimer f on a besoin de l'estimation de σ. Il faut donc décider par où, et comment commencer.

Si on annule la fonction dérivée de $e_{M,\sigma}$ par rapport à σ, on obtient l'équation suivante :

$$1 = \frac{1}{n}\sum_{i=1}^{n}\rho_s(r_i/\sigma) \qquad \text{avec} \qquad \rho_s(r) = \frac{r}{2}\rho'(r) \tag{A.30}$$

On peut alors trouver une solution avec l'estimation L_p. La fonction ρ associée à cette estimation est $x \longmapsto \frac{1}{p}|x|^p$. En effet, grâce à l'équation A.30 page précédente, on peut avoir une forme simple de l'estimation de σ. On obtient :

$$\sigma = \left(\frac{1}{n} \sum_{i=1}^{n} |r_i|^p \right) \tag{A.31}$$

On remarque que pour $p = 2$ on obtient l'écart-type de l'échantillon.

Habituellement, on utilise la plus simple de cette famille de fonctions, à savoir celle de l'estimation L_1. La procédure à suivre est alors d'estimer σ par l'estimation L_1. Cette estimation est faite en prenant d'abord les résidus des moindres carrés puis en affinant grâce à un algorithme IRLS avec la fonction :

$$\rho(r) = \sqrt{\varepsilon' + r^2} \tag{A.32}$$

où $\varepsilon' \to 0$.

Ensuite, on calcule la valeur du $MADN$ (pour *Normalized Mean Absolute Deviation*) qui vaut :

$$MADN = \frac{med(|r_i|)}{0.6745} \tag{A.33}$$

où *med* désigne la médiane des observations.

Le $MADN$ fait partie des estimateurs de l'échelle les plus robustes. Malgré tout, il peut encore être amélioré et deux méthodes permettent ensuite d'estimer les paramètres.

Avant de continuer, nous allons donner une explication sur le $MADN$. Pour cela, nous allons d'abord définir le MAD (pour *Median Absolute Deviation*), qui est un équivalent de l'écart-type mais au sens L_1 :

$$MAD(x) = med\{|x - med(x)|\} \tag{A.34}$$

où x est une variable aléatoire.

Pour une loi normale $\mathcal{N}(0, 1)$, on a $MAD(x) = 0.6745$. C'est alors qu'on définit le $MADN$ de telle sorte qu'une variable aléatoire de loi $\mathcal{N}(\mu, \sigma^2)$ ait un $MADN$ égal à σ :

$$MADN(x) = \frac{med\{|x - med(x)|\}}{0.6745} \tag{A.35}$$

Dans le cas d'une loi normale, le $MADN$ vaut donc exactement σ. Sinon, c'est une bonne estimation de l'échelle d'une distribution. On se sert alors de cette quantité pour estimer σ. Dans notre cas, les variables aléatoires sont Y_i d'espérance $(Rf)_i$. Si on suppose que la loi est symétrique, $med(Y_i) = (Rf)_i$; cela explique la forme du $MADN$ dans l'équation A.33.

A.3.3 Amélioration

On peut améliorer l'estimation de σ obtenue par le $MADN$ en utilisant un M-estimateur de l'échelle. Un tel estimateur $\hat{\sigma}$ satisfait :

$$\delta = \frac{1}{n} \sum_{i=1}^{n} \rho_s(r_i/\hat{\sigma}) \tag{A.36}$$

où :

- $\rho_s(r) = r.\rho'(r)$ est défini à l'aide d'une fonction de poids ρ définie dans l'annexe A.1 page 55 sur les M-estimateurs
- $\delta = \mathbb{E}_\phi(\rho_s)$ est une constante positive et ϕ représente la fonction de répartition de la loi normale centrée réduite. La valeur de δ est obtenue de manière empirique : on génère des réalisations d'une loi normale centrée réduite, on applique ρ_s à ces réalisations puis on calcule la moyenne empirique associée.

Concrètement, on part d'une estimation MADN de σ et de l'identité suivante obtenue grâce à A.36 page précédente :

$$\sigma^2 = \frac{1}{n\delta} \sum_{i=1}^{n} \frac{\rho_s(r_i/\sigma)}{(r_i/\sigma)^2} r_i^2 \tag{A.37}$$

et on applique un algorithme itératif qui s'arrête lorsque la précision voulue est atteinte. A l'itération k, on a :

$$\sigma^{k+1} = \sqrt{\frac{1}{n\delta} \sum_{i=1}^{n} w_i^k r_i^2} \qquad \text{avec} \qquad w_i^k = \frac{\rho_s(r_i/\sigma^k)}{(r_i/\sigma^k)^2} \tag{A.38}$$

En ayant cette possibilité d'estimation, deux solutions peuvent être envisagées afin d'estimer tous les paramètres du modèle. Nous allons les présenter rapidement.

Première solution

On estime d'abord σ de la façon présentée ci-dessus et on applique un algorithme IRLS pour estimer le vecteur des paramètres f.

On prend comme initialisation la solution des moindres carrés. Ensuite, l'itération k se présente de la façon suivante :

- On calcule les résidus de l'étape k grâce aux estimations de l'étape $k-1$: $r^k = y - Rf^{k-1}$
- En utilisant l'estimation de σ faite auparavant, on calcule les poids :

$$B^k = diag\left\{ \frac{\rho'(r_i^k/\sigma)}{2(r_i^k/\sigma)} \right\}_{i=1\cdots n} \tag{A.39}$$

- On calcule l'estimation de f en résolvant $(R'B^kR)f^k = R'B^ky$
- On compare avec l'estimation précédente et on s'arrête au niveau de précision voulu.

Deuxième solution

Après avoir calculé les résidus obtenus par moindres carrés, on applique simultanément deux algorithmes pour estimer en même temps σ et les paramètres du vecteur f. En fait, on estime l'échelle par l'algorithme ci-dessus et en même temps on estime f par IRLS comme cela a été présenté plus haut.

Toutefois, cette solution est moins efficace que la première. Comme elle n'a pas été utilisée dans le cadre du stage, nous ne la détaillerons pas plus.

A.4 ACP et ACP robuste

A.4.1 Introduction et problème

La méthode de l'ACP (Analyse en Composantes Principales) part d'une constatation. Si on augmente le nombre de variables explicatives d'un modèle, on pourrait croire que l'on arrive alors à mieux expliquer la réponse. Or cela est faux, les résultats de classification ne sont pas meilleurs et peuvent même être pires.

Il s'agit alors, si on a beaucoup de variables, de réduire leur nombre, ce qui signifie réduire la dimension de l'espace des paramètres. Il y a alors deux possibilités : sélectionner directement les variables voulues ou faire une transformation des variables et sélectionner ensuite.

L'ACP consiste à appliquer d'une certaine manière la deuxième solution. On veut donc réduire la dimension de l'espace mais en ayant la meilleure représentation possible. Ceci revient à faire une rotation du système de coordonnées.

Dans la suite, on ne verra que les deux cas les plus simples (réduire en un point et réduire en une droite), car ce sont ces cas qui sont utilisés lors du stage. Toutefois, une généralisation est possible pour un nombre de variables quelconque.

A.4.2 Meilleure représentation par un point

Supposons qu'on ait des vecteurs $x_1, \cdots, x_n$ de dimension D. Le but est de les représenter par un point x_0.

On veut, comme pour les moindres carrés, minimiser une erreur, qui est ici l'erreur entre chaque point x_k et x_0. Celle-ci est représentée par $||x_k - x_0||^2$. Comme pour la méthode des moindres carrés, on veut alors minimiser la quantité suivante :

$$\frac{1}{n}\sum_{k=1}^{n} ||x_k - x_0||^2 \tag{A.40}$$

Un calcul (en dérivant la quantité ci-dessus par rapport à x_0) montre que le point cherché est :

$$x_0 = \frac{1}{n}\sum_{k=1}^{n} x_k \tag{A.41}$$

On retrouve la moyenne empirique, qui est déjà la meilleure représentation du jeu de données au sens des moindres carrés.

A.4.3 Meilleure représentation par une droite

On aimerait maintenant représenter les données par une droite, à savoir réduire les données pour aller en dimension 1. La droite cherchée doit passer par la moyenne x_0 trouvée dans la section A.4.2. Les autres paramètres cherchés sont un vecteur $\vec{e}$ et des scalaires a_k pour $k = 1, \cdots, n$. Le vecteur $\vec{e}$ est le vecteur directeur de la droite cherchée. Le nombre a_k représente la distance entre le projeté de x_k sur la droite et le point x_0. Notons μ le point x_0.

Un point x se trouvant sur la droite cherchée, dont la distance avec μ est a, vérifie alors :

$$x = \mu + a.\vec{e} \quad \text{avec} \quad ||\vec{e}|| = 1 \tag{A.42}$$

L'erreur que l'on fait en prenant un point $x_k^* = \mu + a_k.\vec{e}$ au lieu de x_k est mesurée par la quantité :

$$||x_k^* - x_k||^2 = ||(\mu + a_k.\vec{e}) - x_k||^2 \tag{A.43}$$

La quantité à minimiser est alors donnée par :

$$J = \frac{1}{n}\sum_{k=1}^{n} ||(\mu + a_k.\vec{e}) - x_k||^2 \tag{A.44}$$

Il faut la minimiser en chacun des a_k et en $\vec{e}$. Un calcul (dérivation) montre que les a_k vérifient :

$$a_k = \vec{e}^T(x_k - \mu) \tag{A.45}$$

où $\vec{e}^T$ désigne le vecteur transposé de $\vec{e}$.

Il s'agit maintenant de trouver le vecteur directeur $\vec{e}$. En remplaçant, dans J, les a_k par leur valeur, et en faisant quelques simplifications, on obtient :

$$J = -\vec{e}^T\Sigma\vec{e} + \frac{1}{n}\sum_{k=1}^{n} ||x_k - \mu||^2 \tag{A.46}$$

où $\Sigma = \frac{1}{n}\sum_{k=1}^{n}(x_k - \mu)(x_k - \mu)^T$ est la matrice de variance-covariance empirique des points x_k.

Comme le deuxième membre de A.46 ne dépend pas de $\vec{e}$, minimiser J en $\vec{e}$ revient à maximiser $\vec{e}^T\Sigma\vec{e}$ avec la contrainte $||\vec{e}||^2 = 1$.

Pour ce faire, on utilise un *multiplicateur de Lagrange* λ. Celui-ci est utile pour résoudre des problèmes d'optimisation lorsqu'on a des contraintes sur certains paramètres. Comme on n'a ici qu'une contrainte, on introduit un seul scalaire λ et on forme une combinaison linéaire entre la fonction à maximiser et la contrainte. Par définition, le coefficient de la contrainte est λ. Il s'agit alors de rendre maximale la quantité suivante en $\vec{e}$:

$$J' = \vec{e}^T\Sigma\vec{e} - \lambda(\vec{e}^T\vec{e} - 1) \tag{A.47}$$

On obtient :

$$\frac{\partial J'}{\partial \vec{e}} = 2\Sigma\vec{e} - 2\lambda\vec{e} \tag{A.48}$$

d'où :

$$\Sigma\vec{e} = \lambda\vec{e} \tag{A.49}$$

λ est donc une valeur propre de la matrice Σ. On observe également que $\vec{e}^T\Sigma\vec{e} = \lambda$. Rendre maximal le membre de gauche revient donc à prendre pour λ la plus grande valeur propre de Σ.

Le vecteur $\vec{e}$ choisi sera alors le vecteur propre de Σ associé à la plus grande valeur propre de cette matrice.

Pour conclure cette partie, nous pouvons dire qu'on fait une projection orthogonale des points issus des données sur la droite cherchée. On choisit la droite qui minimise les distances entre les points et leurs projetés. On observe alors que les erreurs à minimiser sont perpendiculaires à la droite. Pour la régression linéaire classique, ces erreurs étaient verticales ; pour le reste, la façon de faire est la même pour ces deux solutions.

A.4.4 Début de généralisation

Comme nous n'utilisons que les deux paragraphes ci-dessus dans le cadre du stage, celui-ci ne sera pas détaillé.

De la même manière que ci-dessus, on peut réduire la dimension de l'espace des paramètres pour obtenir un espace d'une dimension voulue quelconque (mais plus petite que l'espace de départ !). Pour avoir un espace de dimension d, il s'agit de trouver une matrice E dont les colonnes sont des vecteurs directeurs pour chaque dimension : e_i, $i = 1, \cdots, d$. On cherche également, pour chaque point, un vecteur de coordonnées a. Le modèle s'écrit :

$$x = \mu + \mathrm{E}a \tag{A.50}$$

La quantité à minimiser est :

$$\frac{1}{n}\sum_{k=1}^{n}\left|\left|\left(\mu + \sum_{i=1}^{d} a_{ki}e_i\right) - x_k\right|\right|^2 \tag{A.51}$$

Nous ne donnons ici que les résultats pour la recherche de minimum.

Le minimum par rapport à a est un vecteur dont les composantes sont les coefficients de la projection orthogonale des points sur l'espace : $a = \mathrm{E}^T(x_k - \mu)$.

Les vecteurs e_i formant les colonnes de E sont les vecteurs propres de Σ, la matrice de variance-covariance empirique des données définie plus haut, correspondant aux d plus grandes valeurs propres. Ces vecteurs sont orthogonaux.

On retrouve donc bien les mêmes résultats que ceux obtenus pour la réduction à un espace à une dimension. On remarque que lorsqu'on réduit la dimension, on projette les vecteurs orthogonalement sur l'espace de dimension plus petite cherché. Le résidu à minimiser est alors orthogonal à l'espace cherché.

A.4.5 ACP robuste

On a vu plus haut des similitudes entre la régression linéaire classique et l'ACP. Une autre ressemblance est que ces deux méthodes sont sensibles aux points aberrants, ou *outliers*. Nous avons présenté une manière de traiter ces points avec la régression linéaire en introduisant les M-estimateurs. De la même façon, nous allons présenter l'ACP robuste, une manière de limiter l'impact des *outliers* sans devoir les supprimer en amont.

Procédons de la même manière que dans l'annexe A.1 page 55 concernant les M-estimateurs. Au lieu de minimiser le carré de la norme des erreurs, nous allons les minimiser par rapport à une fonction ρ.

Il s'agit d'abord de trouver μ_r, semblable à μ de la section A.4.2 page 66. On minimise :

$$\frac{1}{n}\sum_{k=1}^{n}\rho(||x_k - \mu_r||) \tag{A.52}$$

Ici, $||.||$ représente la norme « par ligne ». En effet, $(x_k - \mu_r)$ est une matrice à deux colonnes, pour les deux coordonnées des vecteurs des erreurs. On cherche la longueur de ces vecteurs, qui est donc la norme de chaque ligne de la matrice ci-dessus.

En utilisant la théorie semi-quadratique, cela revient à rendre minimale la quantité :

$$\frac{1}{n}\sum_{k=1}^{n} b_k||x_k - \mu_r||^2 \quad \text{avec} \quad b_k = \frac{\rho'(||r||)}{||2r||} \quad \text{où} \quad r = x_k - \mu_r \tag{A.53}$$

Les b_k sont des coefficients considérés comme constants lors de la minimisation.

Un calcul similaire à celui déjà effectué plus haut nous donne la solution :

$$\mu_r = \frac{\sum_{k=1}^{n} b_k x_k}{\sum_{k=1}^{n} b_k} \tag{A.54}$$

Par contre, on observe que les coefficients b_k dépendent de la quantité μ_r qu'on est en train de calculer. En pratique, on calculera μ_r avec un algorithme IRLS. Celui-ci sera initialisé avec l'ACP classique.

Maintenant que nous avons calculé la moyenne pondérée des données, μ_r, nous pouvons chercher les autres paramètres. Ceux-ci se calculent de la même façon que ci-dessus, en redéfinissant les erreurs, et donc les coefficients b_k. La quantité à minimiser est :

$$J = \frac{1}{n}\sum_{k=1}^{n} \rho\left(||(\mu_r + a_k.\vec{e}) - x_k||\right) \tag{A.55}$$

En utilisant la théorie semi-quadratique, cela revient à rendre minimale la quantité :

$$J = \frac{1}{n}\sum_{k=1}^{n} b_k||(\mu_r + a_k.\vec{e}) - x_k||^2 \quad \text{avec} \quad b_k = \frac{\rho'(||r||)}{||2r||} \quad \text{et} \quad r = (\mu_r + a_k.\vec{e}) - x_k \tag{A.56}$$

En adaptant le calcul effectué à la section A.4.3 page 66, on retrouve une conclusion du même type : les coefficients a_k sont inchangés tandis que $\vec{e}$ est le vecteur propre associé à la plus grande valeur propre de C, matrice de variance-covariance pondérée des données définie ci-dessous.

$$C = \frac{1}{s}\sum_{k=1}^{n} b_k(x_k - \mu_r)(x_k - \mu_r)^T \quad \text{avec} \quad b_k = \frac{\rho'(r)}{2r} \quad \text{et} \quad r = (\mu_r + a_k.\vec{e}) - x_k \tag{A.57}$$

On utilise : $s = \sum_{k=1}^{n} b_k$.

De la même façon que pour la moyenne pondérée, la matrice C, et donc les a_k et $\vec{e}$, sont calculés grâce à un algorithme IRLS. Cet algorithme est initialisé avec l'ACP classique qui correspond au cas où la fonction ρ est quadratique.

A.5 Algorithme EM

A.5.1 Introduction

Cet algorithme est un outil mathématique, dont le nom vient de *Expectation Maximization*, qui va servir à faire de la classification non supervisée. Supposons qu'on ait un échantillon de N vecteurs $X = \{x_1, \cdots, x_N\}$. On ne connaît pas leur étiquette de classe ω_i. Le problème est de déterminer la structure des données.

Pour faire une telle classification, il existe de nombreux algorithmes. En effet, la question est difficile, mais la technique est également très utile dans de nombreux domaines. Nous allons voir l'un de ces algorithmes, qui est itératif.

Dans notre cas, nous aimerions modéliser un ensemble de fonctions de répartitions, chacune d'entre elles étant associée à une classe. Concrètement, on aura deux lois à estimer.

Lorsqu'on applique cet algorithme, on suppose connu le nombre de classes. On suppose aussi connaître la famille de lois à laquelle appartient chaque classe. Toutefois, les classes peuvent être issues de lois différentes.

A.5.2 Cas général

Au début, tous les vecteurs se trouvent dans un seul groupe. Soit C le nombre de classes, $\mathbb{P}(\omega_j)$ la probabilité *a priori* d'être dans la classe j. La densité de chaque vecteur x vaut alors :

$$\mathbb{P}(x|\theta) = \sum_{j=1}^{C} \mathbb{P}(x|\omega_j, \theta_j)\mathbb{P}(\omega_j) \tag{A.58}$$

où θ_j est le vecteur des paramètres de la loi suivie par les points de la classe j. C'est le principe d'un mélange de lois. Mais il s'agit d'estimer Θ, le vecteur de tous les paramètres inconnus, grâce à l'échantillon X. Θ comporte les paramètres de la loi de chaque classe, θ_j (pour une loi normale, on aurait : μ_j et Σ_j). Les « coefficients de mélange » π_j sont également compris dans Θ, avec $\sum_{j=1}^{C} \mathbb{P}(\omega_j) = 1$.

Formons d'abord la vraisemblance :

$$\mathbb{P}(X|\Theta) = \prod_{k=1}^{N} \sum_{j=1}^{C} \mathbb{P}(x_k|\omega_j, \theta_j)\mathbb{P}(\omega_j) \tag{A.59}$$

Dans ce cas, la log-vraisemblance s'écrit :

$$\ln \mathbb{P}(X|\Theta) = \sum_{k=1}^{N} \ln \sum_{j=1}^{C} \mathbb{P}(x_k|\omega_j, \theta_j)\mathbb{P}(\omega_j) \tag{A.60}$$

Cette quantité est à maximiser en Θ afin de trouver l'estimateur du maximum de vraisemblance de ce dernier. Nous verrons dans la section A.5.3 page suivante que les estimateurs des paramètres de lois normales sont relativement simples.

On peut montrer que les estimations des probabilités a priori sont, sous la condition $\sum_{j=1}^{C} \hat{\mathbb{P}}(\omega_j) = 1$:

$$\hat{\mathbb{P}}(\omega_j) = \frac{1}{N} \sum_{k=1}^{N} \mathbb{P}(\omega_j|x_k, \theta) \tag{A.61}$$

Les poids $w_{kj} = \mathbb{P}(\omega_j|x_k, \theta)$ peuvent être vus comme la chance, pour le vecteur x_k, d'appartenir à la classe j.

On peut avoir une expression des poids w_{kj} en utilisant le théorème de Bayes :

$$\mathbb{P}(\omega_j|x_k, \theta) = \frac{\mathbb{P}(x_k|\omega_j, \theta)\hat{\mathbb{P}}(\omega_j)}{\sum_{c=1}^{C} \mathbb{P}(x_k|\omega_c, \theta)\hat{\mathbb{P}}(\omega_c)} \tag{A.62}$$

A ce moment, une « référence circulaire » apparaît :
- On a besoin de connaître le vecteur des paramètres Θ pour calculer les poids $\mathbb{P}(\omega_j|x_k, \theta)$, si on regarde la formule ci-dessus.
- On a besoin de ces poids $\mathbb{P}(\omega_j|x_k, \theta)$ pour calculer Θ (pour une partie des paramètres, cela a déjà été vu plus haut ; pour les paramètres des lois, voir dans la section A.5.3 le cas gaussien).

Une solution pour pallier ce problème est d'utiliser un algorithme itératif de type point fixe. Le principe en est le suivant :
- On choisit un vecteur de paramètres d'initialisation noté Θ^0
- Pour chaque classe j et chaque vecteur x_k, on peut calculer $\mathbb{P}^1(\omega_j|x_k, \theta^0)$; on utilise ensuite les poids $\mathbb{P}^1(\omega_j|x_k, \theta)$ pour estimer Θ^1.
- De la même manière, on fait les itérations suivantes : calculer $\mathbb{P}^{n+1}(\omega_j|x_k, \theta^n)$, puis utiliser $\mathbb{P}^{n+1}(\omega_j|x_k, \theta)$ pour estimer Θ^{n+1}.

Lorsqu'on a atteint un nombre d'itérations suffisant, il s'agit, pour finir le travail, de faire une classification pour chaque vecteur x_k. En fait, il s'agit de déterminer à quelle classe appartient chacun de ces vecteurs. Cela peut se faire facilement. En effet, il suffit de considérer, pour chaque k, la classe j pour laquelle le poids w_{kj}^{n+1} est le plus élevé. Le poids choisi représente la classe dans laquelle le vecteur x_k a le maximum de chances de se trouver. C'est donc cette classe j qui était cherchée.

En fait, l'« algorithme EM » est une manière d'expliquer de théoriquement l'algorithme ci-dessus et en assure le bon fonctionnement.

Comme cela a déjà été indiqué, EM signifie *Expectation - Maximization*, ce qui représente les deux étapes de l'algorithme.

A.5.3 Cas gaussien

Dans le cas où les lois des différentes classes cherchées sont gaussiennes, de paramètres μ_j et Σ_j pour la classe j, on peut obtenir une estimation assez simple de ces paramètres. En effet, on a :

$$\hat{\mu_j} = \frac{\sum_{k=1}^{N} \mathbb{P}(\omega_j|x_k, \theta) \times x_k}{\sum_{k=1}^{N} \mathbb{P}(\omega_j|x_k, \theta)} \tag{A.63}$$

De plus,

$$\hat{\Sigma_j} = \frac{\sum_{k=1}^{N} \mathbb{P}(\omega_j|x_k, \theta) \times (x_k - \hat{\mu_j})(x_k - \hat{\mu_j})^T}{\sum_{k=1}^{N} \mathbb{P}(\omega_j|x_k, \theta)} \tag{A.64}$$

où v^T représente le vecteur transposé de v.

On rappelle les valeurs des estimations des proportions de chaque classe :

$$\hat{\mathbb{P}}(\omega_j) = \frac{1}{N}\sum_{k=1}^{N}\mathbb{P}(\omega_j|x_k,\theta) \tag{A.65}$$

On peut remarquer que ce sont des moyennes et des matrices de variance-covariance pondérées, de la même façon que lors de l'ACP robuste.

Dans le cas où les poids $\mathbb{P}(\omega_j|x_k,\theta)$ sont binaires (cela veut dire qu'ils valent 0 ou 1), cela revient à dire qu'on connaît par avance les classes dans lesquelles sont les points.

Dans ce cas, on aurait $\sum_k \mathbb{P}(\omega_j|x_k,\theta) = \sum_{x_k\in\omega_j} 1 = N_j$. De même, $\sum_k \mathbb{P}(\omega_j|x_k,\theta)x_k = \sum_{x_k\in\omega_j} x_k$.

Grâce à ces remarques, on obtient les estimations suivantes des paramètres :

$$\hat{\mu_j} = \frac{1}{N_j}\sum_{x_k\in\omega_j} x_k \tag{A.66}$$

$$\hat{\Sigma_j} = \frac{1}{N_j}\sum_{x_k\in\omega_j}(x_k-\hat{\mu_j})(x_k-\hat{\mu_j})^T \tag{A.67}$$

Les proportions de chaque classe deviennent alors : $\hat{\mathbb{P}}(\omega_j) = \frac{N_j}{N}$.

On retrouve bien les estimateurs connus.

Faisons maintenant une itération dans ce cas gaussien. On suppose être à l'étape n de l'algorithme. On a alors, pour chaque classe j, les estimations des paramètres $\hat{\mu_j}^n$, $\hat{\Sigma_j}^n$ et $\hat{\mathbb{P}}(\omega_j)^n$. Il s'agit de calculer les poids qui vont servir à estimer ces paramètres à l'étape $n+1$. Ils valent :

$$\begin{aligned} w_{kj}^n &= \mathbb{P}(\omega_j|x_k,\Theta^n) \\ &= \frac{\mathbb{P}(x_k|\omega_j,\theta^n)\mathbb{P}^n(\omega_j)}{\sum_{c=1}^{C}\mathbb{P}(x_k|\omega_c,\theta^n)\mathbb{P}^n(\omega_c)} \end{aligned} \tag{A.68}$$

Tous les paramètres qui sont à l'exposant n sont connus, car ils viennent d'être calculés. De plus, on a :

$$\mathbb{P}(x_k|\omega_j,\theta^n) = \frac{1}{(2\pi)^{d/2}|\Sigma_j^n|^{1/2}}\exp\left(-\frac{1}{2}(x_k-\mu_j^n)(\Sigma_j^n)^{-1}(x_k-\mu_j^n)^T\right) \tag{A.69}$$

En effet, c'est la densité d'une loi normale multivariée, de dimension d. A ce moment, on est capable de calculer les estimations des composantes du vecteur des paramètres Θ, à l'étape $n+1$. Cela donne :

$$\hat{\mu_j}^{n+1} = \frac{\sum_{k=1}^{N} w_{kj}^n x_k}{\sum_{k=1}^{N} w_{kj}^n} \tag{A.70}$$

$$\hat{\Sigma_j}^{n+1} = \frac{\sum_{k=1}^{N} w_{kj}^n (x_k-\hat{\mu_j}^{n+1})(x_k-\hat{\mu_j}^{n+1})^T}{\sum_{k=1}^{N} w_{kj}^n} \tag{A.71}$$

$$\hat{\mathbb{P}}^{n+1}(\omega_j) = \frac{1}{N}\sum_{k=1}^{N} w_{kj}^n \tag{A.72}$$

On en déduit alors que l'on peut, dans le cas gaussien, facilement utiliser cet algorithme EM.

A.5.4 Utilisation dans le cadre du stage

Nous avons deux classes : les véhicules légers et les poids lourds. Les indications enregistrées dans le logiciel `dBEuler` concernant la catégorie de chaque véhicule mesuré forment l'initialisation.

On suppose que ces deux classes sont issues d'une loi normale à deux dimensions. Pour les données utilisées, un test a été effectué afin de confirmer cette hypothèse. On peut facilement calculer une première estimation de la moyenne et de la matrice de variance-covariance pour chacune de ces classes. On peut aussi calculer la proportion de véhicules dans chacune des catégories. Tous ces calculs nous donnent Θ^0.

Ces étiquettes sont ensuite oubliées pour n'avoir plus qu'une seule classe. On commence alors l'algorithme. En utilisant la densité de la loi normale multidimensionnelle, on calcule les « poids » conformément aux formules du paragraphe précédent. On peut alors calculer les moyennes et les matrices de variance-covariance pondérées, et ainsi de suite.

Après un nombre fini d'itérations (choisi à l'avance), on obtient les paramètres de deux nouvelles lois gaussiennes, correspondant à chacune des classes. Grâce aux derniers poids calculés, on peut alors décider dans quelle classe mettre chaque point.

L'implémentation de cet algorithme est assez simple, mais sa convergence est relativement lente. Par contre, plus x_k est proche de la moyenne de l'une des classes, plus le poids de x_k associé à cette classe sera grand. Dans ce cas, on pourra facilement faire la classification. Dans les données disponibles, comme les classes de départ sont déjà formées, l'initialisation est très bonne et l'algorithme converge donc plus rapidement.

A.6 Méthode du *bootstrap*

Ce compte-rendu a été réalisé grâce à des éléments trouvés dans [LPM06], [Sap06] et [WW91]. Les articles [Sha10] et [You94] ont également aidé à la rédaction de cette partie.

A.6.1 Principe

La méthode du *bootstrap* a vu le jour à la fin des années 1970 avec B. Efron. C'est une méthode de rééchantillonnage qui permet de faire de l'estimation en créant de nouveaux jeux de données grâce à des données qui ont été observées. Le *bootstrap* n'était pas envisageable avant l'augmentation des moyens de calcul apportée par l'ordinateur.

Nous pouvons ainsi expliquer l'origine du mot « *bootstrap* ». Ce mot désigne une bande de cuir que l'on peut trouver sur le côté des bottes et qui sert à mieux les enfiler en y passant le doigt et en tirant. L'utilisation de ce nom fait référence aux aventures extraordinaires du Baron de Münchhausen, personnage historique et héros de la littérature allemande. L'une d'elles raconte qu'il se serait sorti d'un marécage en se tirant par les bottes. Cela a été repris par une expression américaine « *to pull oneself up by one's own bootstraps* » qui est une métaphore pour décrire une ascension sociale réussie uniquement par soi-même. Le lien avec la méthode statistique du *bootstrap* est qu'on arrive à produire des résultats et un gros échantillon en utilisant uniquement l'échantillon observé au départ et l'ordinateur.

La méthode est utilisée pour des cas complexes où les hypothèses habituelles ne sont pas vérifiées, ou alors lorsque la distribution des paramètres est inconnue. L'un de ses principaux avantages est justement que peu ou pas d'hypothèses sont nécessaires pour le mettre en oeuvre.

Supposons que l'on ait un n-échantillon indépendant $x = (x_1, \cdots, x_n)$ issu des variables alétoires $(X_1, \cdots, X_n)$. Ces variables aléatoires sont distribuées comme une certaine variable X de fonction de répartition F inconnue.

La distribution de X est donc inconnue ; par contre, nous avons des réalisations de cette variable aléatoire et nous pouvons alors construire une distribution empirique de X. C'est alors la meilleure solution si on veut avoir d'autres réalisations de X.

Ce qui nous amène à utiliser cette méthode est qu'on veut estimer un (ou des) paramètre(s) de la distribution de X, noté θ. Certaines méthodes de *bootstrap* permettent également de choisir le nombre d'axes lors d'une ACP (donc de choisir la meilleure dimension de l'espace de représentation).

A.6.2 Deux méthodes

Deux principales méthodes sont utilisées : la méthode paramétrique et la méthode non paramétrique.

Méthode paramétrique

Elle peut être utilisée par exemple si on connaît ou si on a une idée de la famille de distributions à laquelle appartient X. Dans ce cas, on estime la distribution de X avec le premier échantillon obtenu et on ajuste le modèle.

Ensuite, on fait des simulations afin d'obtenir d'autres échantillons par rapport à cette distribution estimée et non par rapport à l'échantillon de départ en lui-même. Par exemple, on simule m n-échantillons suivant la distribution estimée. Pour chaque échantillon, on peut recalculer l'estimation du paramètre θ.

Méthode non paramétrique

Lorsqu'on ne connaît pas du tout la distribution de X, on peut utiliser cette méthode. Comme ci-dessus, on génère m n-échantillons. Mais cette fois-ci, ils ne proviennent plus d'une distribution estimée ; ils sont directement obtenus grâce à l'échantillon de départ x.

En effet, on associe une probabilité $1/n$ à chaque élément x_i de X, et on fait, à chaque fois, n tirages avec remise dans cette population. En tout, il est possible d'avoir n^n échantillons différents.

Avantages et inconvénients

L'avantage de la méthode non paramétrique est qu'elle est toujours applicable, même lorsqu'on ne connaît pas la distribution de l'échantillon.

Par contre, si la forme de la loi est connue, l'approximation est meilleure lorsqu'on utilise la méthode paramétrique. En effet, on peut évaluer la fonction de répartition de la distribution de manière plus efficace. Il faut toutefois veiller à ne pas se tromper de loi, car dans ce cas, les résultats obtenus sont plus mauvais que ceux obtenus par la méthode non-paramétrique.

Estimation du paramètre et intervalle de confiance

Pour chaque échantillon simulé j, on obtient une valeur $\hat{\theta}_j$. La distribution empirique de ces $\hat{\theta}_j$ est une approximation de la distribution théorique de $\hat{\theta}$. On peut alors prendre la moyenne empirique de ces $\hat{\theta}_j$ comme une nouvelle estimation de θ.

Si m, le nombre total d'échantillons obtenus, tend vers $+\infty$, alors la moyenne des $\hat{\theta}_j$ converge vers l'estimateur du maximum de vraisemblance empirique (i.e. obtenu avec la fonction de répartition empirique de X). La variance empirique des $\hat{\theta}_j$ peut également être calculée.

En pratique, quelques centaines de réplications au plus sont effectuées.

Ces simulations successives permettent également de calculer de différentes manières des intervalles de confiance pour θ.

La première est appelée *méthode des percentiles*. C'est la méthode la plus simple. En effet, on a obtenu m valeurs estimées de θ qui forment une certaine distribution. Pour avoir par exemple un intervalle bilatéral à 95%, il suffit de déterminer les 95% des valeurs centrales de la distribution obtenue, et d'en prendre les bornes. Plus généralement, il s'agit de récupérer les quantiles souhaités dans la distribution formée par les m valeurs.

La deuxième possibilité est de faire une approximation normale. On calcule la moyenne ($\bar{\theta}$) et l'écart-type ($\hat{\sigma}$) des $\hat{\theta}_j$ et on utilise $[\bar{\theta} - 1,96\hat{\sigma}/\sqrt{n}); \bar{\theta} + 1,96\hat{\sigma}/\sqrt{n}]$ qui est un intervalle de confiance à 95% pour θ. Bien sûr, pour utiliser cette méthode, il faut avoir vérifié la normalité de l'échantillon utilisé, en utilisant un test.

Il existe également d'autres méthodes plus complexes.

Par rapport à l'intérêt de ces intervalles de confiance, des études menées par B. Efron indiquent que l'intervalle de confiance obtenu par *bootstrap* est de même amplitude que celui obtenu grâce à la vraie distribution, pour de nombreux paramètres statistiques.

A.6.3 Usages et Limites

Dans certaines applications, comme l'estimation de quantiles, la méthode du *bootstrap* a fait ses preuves.

Par contre, elle a aussi ses limites et a été critiquée dans certains articles, par exemple [You94]. De manière théorique, le *bootstrap* fonctionne bien, mais ce n'est pas toujours le cas en pratique.

Par exemple, si la taille de l'échantillon de départ x est faible, les intervalles de confiance auront tendance à être trop écroits. En effet, la méthode du *bootstrap* ne permet pas d'engendrer des valeurs autres que celles observées au départ.

Un autre exemple où elle ne fonctionne pas est l'estimation des bornes de l'intervalle définissant une loi uniforme. Pour la même raison que ci-dessus, on ne pourra pas obtenir de meilleure estimation de ces bornes en faisant du *bootstrap*. Ces bornes sont généralement estimées par le maximum ou le minimum de l'échantillon (selon la borne). Comme l'échantillon utilisé est toujours le même, on ne pourra pas obtenir d'estimation plus précise en rééchantillonnant.

Comme de nombreuses méthodes statistiques, la méthode du *bootstrap* fonctionne mieux pour de grands échantillons.

Selon l'usage que l'on veut en faire, la méthode du *bootstrap* ne nécessite pas le même nombre de réplications. On fera de 25 à 50 rééchantillonnages pour avoir un début d'information. Mais dans le cas où on veut évaluer des intervalles de confiance, on fera plutôt 500 réplications.

A.6.4 Exemple

Nous allons donner un exemple d'utilisation de la méthode du *bootstrap* non paramétrique. Cela me permet d'illustrer les explications ci-dessus et de comparer les résultats de [Sap06] et ceux obtenus avec ma programmation en langage `Scilab`. Cet exemple est tiré de [Sap06].

Nous avons un échantillon de $n = 100$ valeurs, et le but est de donner un intervalle de confiance pour la médiane de cet échantillon. Nous allons utiliser la méthode du *bootstrap* en faisant $N = 1000$ rééchantillonnages aléatoires de l'échantillon de départ. Nous allons ensuite construire des intervalles de confiance de deux types pour cette médiane : par la méthode des percentiles et la méthode de l'approximation normale.

La médiane de cet échantillon vaut $\hat{m} = 17,625$. Pour chaque échantillon j généré, nous calculons la médiane $\hat{M}_j$. Cela va nous donner 1000 valeurs de médianes qui formeront une distribution empirique de $\hat{m}$ et qui permettront de faire un intervalle de confiance. La table A.2 page suivante nous donne le premier échantillon généré dans lequel la médiane vaut $\hat{M}_1 = 18.16$.

Comparons d'abord les résultats du livre et ceux obtenus avec `Scilab` concernant des caractéristiques générales de $\hat{M} = \left(\hat{M}_j\right)_{i=1\cdots N}$. Par exemple, la « moyenne » représente la moyenne empirique du vecteur $\hat{M}$. La figure A.4 page 78 montre un histogramme représentant les valeurs du vecteur des médianes $\hat{M}$.

Nous calculons ensuite les intervalles de confiance de cette médiane par la méthode des percentiles et celle de l'approximation normale.

18.94	22.06	17.97	18.86	14.97	14.3	11.07	16.9	24.49	10.13
12.45	15.59	22.41	15.37	16.2	20.2	22.14	17.07	22.11	9.4
15.77	25.99	15.82	16.12	23.36	19.72	17.3	18.71	15.09	17.3
17.37	14	15.85	16.31	14.16	4.86	17.58	18.75	10.42	28.69
21.27	12.10	19.43	9.16	18.75	25.98	19.48	17.67	17.54	36.17
17.24	16.74	19.09	10.3	21.93	19.39	16.62	12.72	21.4	13.67
16.65	18.21	6.13	21.13	3.68	19.75	30.23	19.19	16.91	20.05
9.15	21.31	15.87	20.71	21.55	16.67	14.98	21.75	27.97	20.97
14.93	9.17	19.9	10.82	16.36	20.46	19.32	12.38	22.04	19.37
19.23	33.61	20.79	18.11	16.25	18.70	8.95	29.96	19.85	11.5

TABLE A.1 – Bootstrap : données de l'exemple

22.41	17.97	14.16	20.46	14.97	33.61	17.3	20.46	19.85	18.86
4.86	19.75	21.27	17.3	28.69	15.77	9.46	14.97	29.96	11.07
10.13	10.13	9.16	28.69	19.48	17.3	17.58	10.3	16.65	21.31
22.41	10.13	17.3	22.11	17.3	18.7	10.3	18.11	12.38	15.09
18.75	11.07	18.7	15.77	29.96	14.98	25.98	14.16	16.9	17.97
21.4	18.11	16.65	11.07	15.82	19.32	22.14	18.71	14.16	11.07
28.69	18.21	17.3	24.49	19.09	17.24	22.41	19.09	22.06	20.71
19.72	19.43	10.13	20.97	33.61	17.3	10.13	14.93	22.04	24.49
27.97	16.12	11.5	18.94	19.32	20.71	19.9	10.82	15.77	20.05
13.67	18.75	21.93	18.7	17.67	17.58	19.43	15.59	18.94	18.75

TABLE A.2 – Bootstrap : exemple d'échantillon généré avec les données de la table A.1

Pour les bornes de l'intervalle des percentiles, nous prenons la 25^{e} et la 975^{e} valeur du vecteur des médianes $\hat{M}$ précédemment trié par ordre croissant afin d'avoir un intervalle à 95%.

Pour faire l'approximation normale, il faudrait d'abord vérifier la normalité des données utilisées. Cela ne sera pas effectué ici puisque ces données ne rentrent pas dans le cadre du stage.

A nouveau, comparons les valeurs obtenues :

Dans tous les calculs effectués, nous observons que les résultats obtenus par calcul sont similaires à ceux lus dans [Sap06].

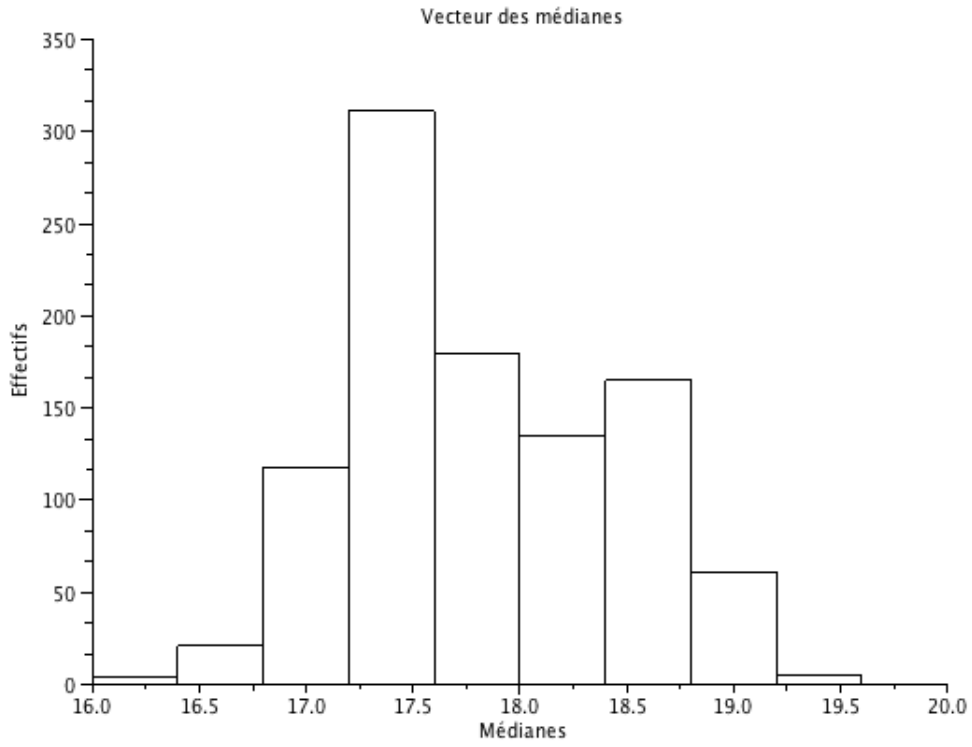


FIGURE A.4 – Histogramme - médianes de l'échantillon obtenu

Script `Scilab` et résultats pour l'exemple

```
-->// echantillon issu du livre de Saporta pp. 110 et suiv.
-->// exemple de bootstrap p.380
-->
-->x1[18.94,22.06,17.97,18.86,14.97,14.3,11.07,16.9,24.49,10.13,
 12.45,15.59,22.41,15.37,16.2,20.2,22.14,17.07,22.11,9.46,15.77,
 25.99,15.82,16.12,23.36,19.72,17.3,18.71,15.09,17.3,17.37,14,
 15.85,16.31,14.16,4.86,17.58,18.75,10.42,28.69,21.27,12.10,
 19.43,9.16,18.75,25.98,19.48,17.67,17.54,36.17,17.24,16.74,19.09];
-->x1=x1';
-->
-->x2=[10.3,21.93,19.39,16.62,12.72,21.4,13.67,16.65,18.21,6.13,
 21.13,3.68,19.75,30.23,19.19,16.91,20.05,9.15,21.31,15.87,20.71,
 21.55,16.67,14.98,21.75,27.97,20.97,14.93,9.17,19.9,10.82,16.36,
 20.46,19.32,12.38,22.04,19.37,19.23,33.61,20.79,18.11,16.25,18.70,
 8.95,29.96,19.85,11.5];
-->x2=x2';
-->
-->x=[x1;x2];
```

	Livre	Echantillon obtenu
Moyenne	17.7872	17.811565
Médiane	17,625	17.67
Ecart-type	0.630658	0.6258635
Minimum	15,87	16.25
Maximum	19,39	19.39

TABLE A.3 – Comparaison de caractéristiques générales

	Livre	Echantillon obtenu
Intervalle de confiance des percentiles	[16.70,18.92]	[16.79,19.01]
Intervalle de confiance avec approximation normale	[16.55,19.02]	[16.58,19.04]

TABLE A.4 – Comparaison d'intervalles de confiance

```
-->n=length(x)
 n  =
    100.
-->
-->//on fait 1000 reechantillonages
-->//dans chaque colonne de 'ech' se trouve un echantillon
-->//aleatoire obtenu par tirage avc remise ds l'ech de depart.
-->
-->ech=[];
-->N=1000
 N  =
    1000.
-->for i=1:N
-->  ech=[ech,sample(n,x)];
-->end
-->
-->//vecteur des medianes de chaque echantillon
-->meds=median(ech,'r');
-->meds=meds';
-->
-->//caracteristiques de ce vecteur
-->mmeds=mean(meds)
 mmeds  =
    17.811565
-->median(meds)
 ans  =
    17.67
-->smeds=sqrt((1/N)*sum((meds-mmeds*ones(meds)).^2))
 smeds  =
    0.6258635
-->min(meds)
 ans  =
    16.25
-->max(meds)
 ans  =
    19.39
```

```
-->
-->//intervalle des percentiles
-->meds_sort=gsort(meds,'g','i');
-->meds_sort(25)
 ans  =
    16.79
-->meds_sort(975)
 ans  =
    19.015
-->
-->//intervalle avec approximation normale
-->mmeds-1.96*smeds
 ans  =
    16.584872
-->mmeds+1.96*smeds
 ans  =
    19.038258
-->
-->//histogramme
-->c=16:0.4:20;
-->histplot(c,meds,normalization=%f)
-->xtitle('Vecteur␣des␣medianes','Medianes','Effectifs')
```

Annexe B

Données

B.1 Première étude

Les données utilisées proviennent principalement de 3 jeux :
- *Rothau2009* (avec une distinction VL et PL)
- *Haguenau2009* (avec une distinction VL et PL)
- *Motos30082009* (deux fichiers, *sens1* et *sens2*)

D'autres jeux de données ont été utilisés afin de tester les différents algorithmes. Les jeux de données ci-dessus ont servi aux premières analyses. Les fichiers correspondants aux motos n'ont pas de rapport direct avec les autres fichiers et ne correspondent pas aux normes, mais les mesures ont été faites dans un cadre de recherche et ont été exploitées comme une mesure classique de bruit de roulement. La vitesse de référence pour les motos est de 90 km/h.

Concernant les données sur les motos, des travaux ont été effectués en 2009 par Thierry Wilhelm, alors étudiant du Master Statistique. Ils sont rassemblés dans le rapport [Wil09] Ceux-ci ont montré que les fichiers résultant des mesures dans les deux sens de circulation (fichiers *sens1* et *sens2*) sur une même route peuvent être étudiés ensemble. En effet, d'après ses travaux, le sens de circulation des motos n'influe pas sur leur bruit de roulement.

Les résultats sur les motos sont donc obtenus en combinant les deux fichiers concernant une même route.

Dans les tableaux suivants, nous donnons quelques caractéristiques des différents fichiers étudiés.

	Hagu.VL	Hagu.PL	Roth.VL	Roth.PL	Motos
Nombre d'observations	128	47	122	43	96

Table B.1 – Nombre d'observations

Comme remarque générale, on peut dire que pour la variable L_{Amax}, la dispersion des observations est assez grande. De plus, pour la plupart des fichiers, les observations de la variable « paramètre adimensionnel de vitesse » (voir sa définition dans 1.1 page 8 ne sont pas centrées, à savoir que le maximum pour cette variable dépasse à peine 0. Cela est problématique, car c'est à cet endroit qu'on lira la valeur de l'ordonnée de la droite. Si peu de données se trouvent au delà de 0, cela peut fausser les résultats.

	Hagu.VL	Hagu.PL	Roth.VL	Roth.PL	Motos
Minimum	78.02	79.07	66.72	78.83	67.03
Maximum	89.60	91.40	86.35	88.55	85.75
Moyenne	82.43	87.30	76.29	82.85	75.58
Ecart-type	1.87	2.06	3.07	2.51	3.59

TABLE B.2 – Caractéristiques - variable L_{Amax}

	Hagu.VL	Hagu.PL	Roth.VL	Roth.PL	Motos
Minimum	-0.09	-0.03	-0.27	-0.18	-0.29
Maximum	0.17	0.07	0.03	0.01	0.02
Moyenne	0.04	0.02	-0.09	-0.08	-0.13
Ecart-type	0.05	0.03	0.06	0.05	0.08

TABLE B.3 – Caractéristiques - variable paramètre adimensionnel de vitesse (variable X)

Par ailleurs, les fichiers cités ci-dessus sont issus d'une étude comme celles qui ont été présentées en introduction. Un PV est disponible pour chaque étude. Dans ceux-ci, on peut trouver les paramètres de la droite qui ont été retenus. Dans le tableau B.4, on peut trouver ces coordonnées de droites. Cela permettra de faire une comparaison entre la méthode consistant à enlever manuellement les points aberrants et celles qui sont étudiées dans le cadre du stage. Les données pour les motos ont été séparées dans le PV concernant cette campagne.

	Hagu.VL	Hagu.PL	Roth.VL	Roth.PL	Motos.sens1	Motos.sens2
Pente	25.6	36.2	30.9	23.9	41.4	27.5
Origine	81.1	86.7	78	84.8	78.8	79.6

TABLE B.4 – Paramètres des droites selon le PV de mesure

B.2 Sous forme de fichiers texte

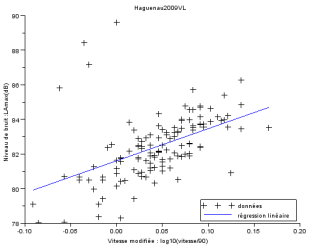
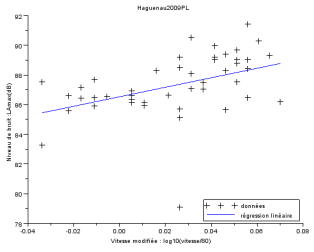
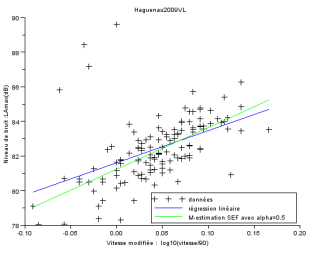
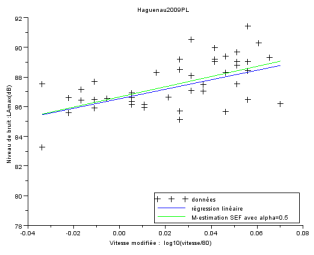
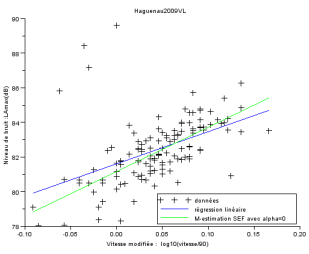
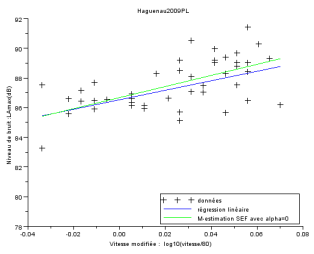
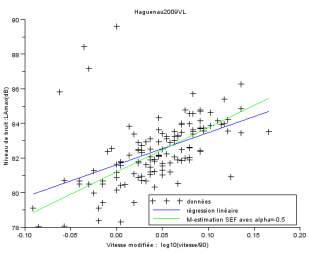
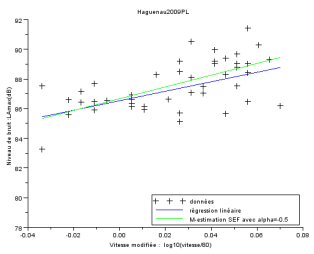
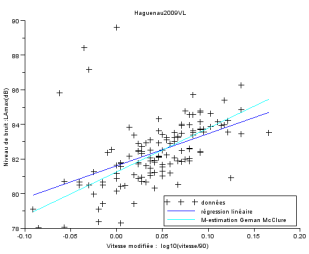
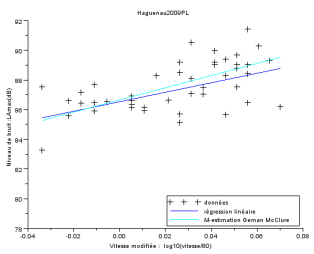
Les données étaient disponibles au format binaire. Après chargement dans `Scilab` et traitement des vitesses (pour les normaliser), j'ai obtenu des fichiers texte décrivant ces données. En fait, ce sont des tableaux dont les colonnes sont respectivement la vitesse, le paramètre adimensionnel de vitesse (variable X) et le niveau de bruit L_{Amax}.

B.3 Planches

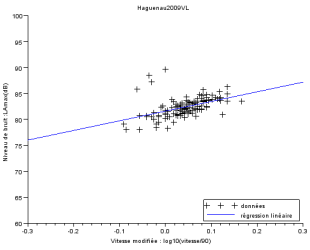
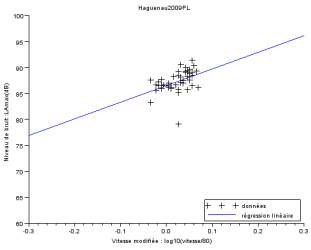
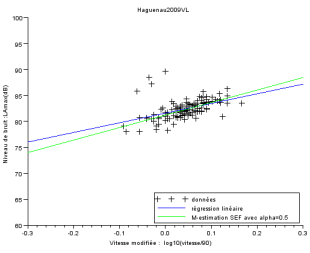
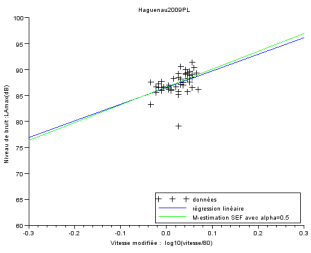
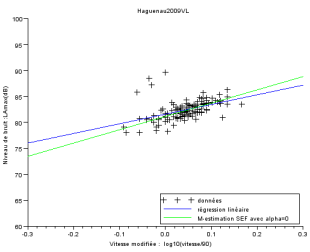
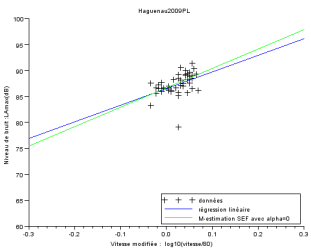
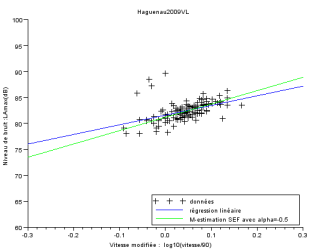
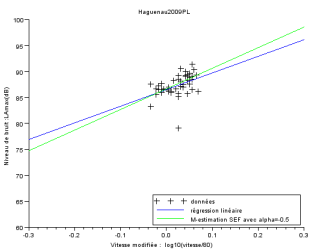
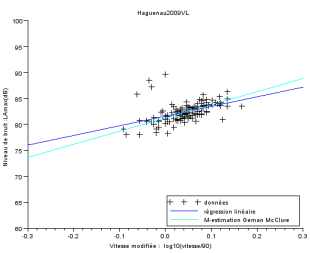
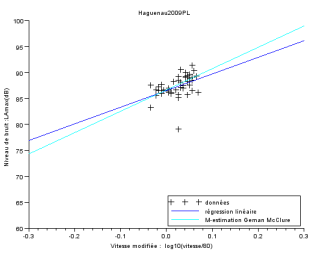
A partir de ces fichiers et des M-estimations, des planches ont été faites afin de comparer les différentes droites obtenues. Sur chaque graphique de ces planches sont représentées les données, la droite de régression linéaire, ainsi qu'une autre droite correspondant au M-estimateur indiqué dans la première colonne.

Quelques exemples sont donnés dans les pages suivantes : les données *Haguenau2009* avec axes automatiques, les mêmes données pour lesquelles les axes ont été fixés, et les données *Motos30082009*, avec les deux possibilités pour les axes.

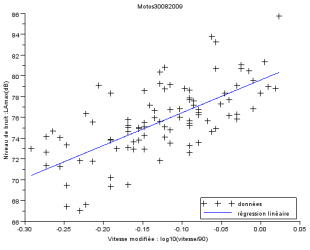
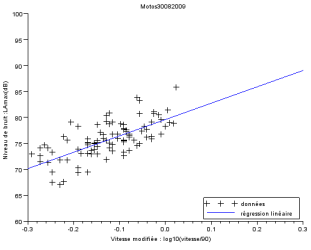
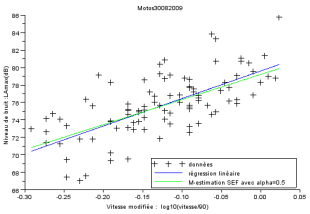
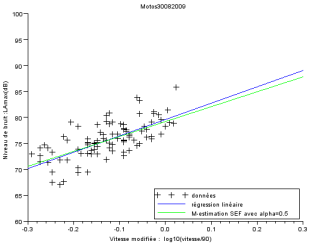
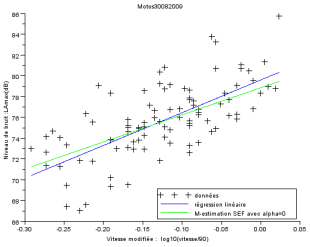
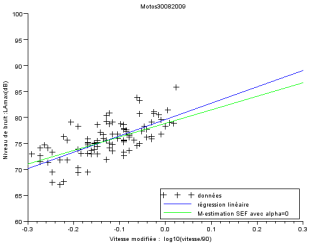
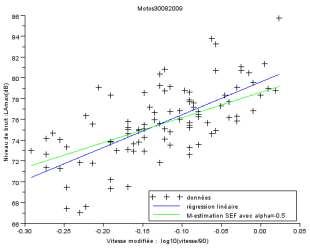
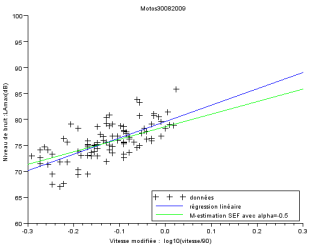
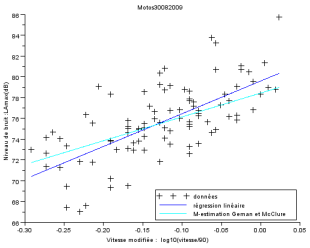
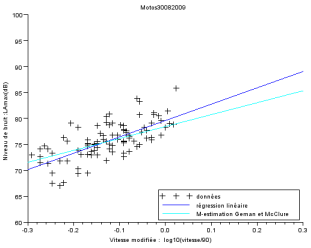
Graphiques à axes automatiques - campagne Haguenau 2009

Haguenau2009	VL	PL
Régression Linéaire		
SEF - $\alpha = 0.5$		
SEF - $\alpha = 0$		
SEF - $\alpha = -0.5$		
Geman et McClure - ou SEF $\alpha = -1$		

Graphiques à axes fixes - campagne Haguenau 2009

Haguenau2009	VL	PL
Régression Linéaire		
SEF - $\alpha = 0.5$		
SEF - $\alpha = 0$		
SEF - $\alpha = -0.5$		
Geman et McClure - ou SEF $\alpha = -1$		

Graphiques à axes automatiques et fixes - campagne Motos30082009

Motos30082009	axes automatiques	axes fixes (pour toutes les données)
Régression Linéaire		
SEF - $\alpha = 0.5$		
SEF - $\alpha = 0$		
SEF - $\alpha = -0.5$		
Geman et McClure - ou SEF $\alpha = -1$		

Bibliographie

[31-93] S 31-119. *Acoustique - Caractérisation in situ des qualités acoustiques des revêtements de chaussées - Mesurages acoustiques au passage (norme annulée)*. AFNOR, Octobre 1993.

[Ber11] Frédéric Bertrand. *Tests de normalité - Distributions univariées et multivariées*. IRMA, Université de Strasbourg, 2011.

[BHL+09] F. Besnard, J-F. Hamet, J. Lelong, E. Le Duc, V. Guizard, N. Fürst, and S. Doisy. *Prévision du bruit routier - 1) Calcul des émissions sonores dues au trafic routier*. SETRA (Service d'Etudes sur les Transports, les Routes et leurs Aménagements), juin 2009.

[Bis95] C. Bishop. *Neural Networks for Pattern Recognition*. Information science and Statistics. Oxford University Press, Oxford, Angleterre, 1995.

[Bis06] C. Bishop. *Pattern Recognition and Machine Learning*. Information science and Statistics. Springer, New-York, USA, première edition, 2006.

[Cha11a] Pierre Charbonnier. *Classification et reconnaissance des formes*. Master IRIV (2ème année) - ENSPS - Université de Strasbourg, 2011.

[Cha11b] Pierre Charbonnier. *Estimation robuste en traitement d'images*. Master IRIV (2ème année) - ENSPS - Université de Strasbourg, 2011.

[DHS01] R. Duda, P. Hart, and D. Stork. *Pattern classification*. Wiley-Interscience, seconde edition, 2001.

[Dut06] Guillaume Dutilleux. *Projet dBEuler : Outils pour la mesure de bruit de roulement au passage*, 2006.

[LPM06] Ludovic Lebart, Marie Piron, and Alain Morineau. *Statistique exploratoire multidimensionnelle*. Dunod, 2006.

[Mar06] Ricardo A. Maronna. *Robust Statistics - Theory and methods*. John Wiley and Sons, 2006.

[Sap06] Gilbert Saporta. *Probabilités, Analyse des données et Statistique*. Editions Technip, 2006.

[Sha10] Cosma Shalizi. The bootstrap. *American Scientist*, 98, 2010.

[TIC02] J.P. Tarel, S.S. Ieng, and P. Charbonnier. *Using robust estimation algorithms for tracking explicit curves*. Springer, 6th European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science, volume 2350, Copenhague, Danemark, mai 2002.

[TIC07] J.P. Tarel, S.S. Ieng, and P. Charbonnier. *Robust Lane Marking Detection by the Half Quadratic Approach.* Etudes et Recherches des Laboratoires des Ponts et Chaussées CR49, LCPC, novembre 2007.

[Wil09] Thierry Wilhelm. Etude de l'influence de divers paramètres sur l'émission acoustique de tramways et deux roues. Master's thesis, Université de Strasbourg, 2009.

[Win10] Renaud Wintzer. Rapport de stage : Evolution de dBEuler, 2010.

[WW91] Thomas H. Wonnacott and Ronald J. Wonnacott. *Statistique.* economica, 1991.

[You94] G. Alastair Young. Bootstrap : More than a stab in the dark ? *Statistical Science*, 9(3) :382–415, 1994.

[ZSB02] Ruben H. Zamar and Matias Salibian-Barrera. Bootstrapping robust estimates of regression. *The Annals of Statistics*, 30(2) :556–582, 2002.