UCL Université catholique de Louvain

École polytechnique de Louvain (EPL)

epl

Quantification de l'anonymat dans les bases de données

Mémoire présenté par
Sandrine ROMAINVILLE , Jolan WERNER

en vue de l'obtention du grade de Master
ingénieur civil en mathématiques appliquées

Promoteurs
Francois-Xavier STANDAERT, Vincent BLONDEL

Lecteur
Clément MASSART

Année académique 2016-2017

Table des matières

1 Introduction **1**
1.1 Motivations 1
1.2 État de l'art 6
1.3 Objectifs 10
1.4 Plan 11

2 Données expérimentales et notations **13**

3 Outils d'analyse **17**
3.1 Métriques utilisées 17
3.2 Validation croisée 23
3.3 Méthodes d'attaque 23
3.4 Méthodes de défense 26
3.5 Problème des outliers 27

4 Analyse des données simulées **33**
4.1 Information mutuelle 33
4.2 Optimisation des méthodes d'attaque 34
4.2.1 Modèle discret 34
4.2.2 Modèle par Noyau 35
4.2.3 Modèle par maximisation d'espérance 38
4.2.4 Méthode des K plus proches voisins 40
4.2.5 Méthode des arbres de décision 43
4.3 Comparaison des différentes méthodes 46

5 Analyse des données réelles **51**
5.1 Données brutes 52
5.2 Données agrégées 55
5.3 Données bruitées 57

Conclusion **61**

A Validation des résultats théoriques **63**

B Analyse complète des méthodes sur les données réelles **65**
B.1 Données brutes 65
B.2 Données agrégées 69
B.3 Données bruitées 71

Résumé

Avec le nombre grandissant de bases de données contenant maintes informations sur maints individus, les questions de la vie privée et de l'anonymat reviennent souvent aujourd'hui. Les firmes doivent d'ailleurs de plus en plus répondre à des critères visant à garder une certaine sécurité, un certain anonymat dans leurs bases de données. Il semble donc nécessaire d'avoir une estimation de l'information disponible au travers de ces données, notamment au travers de celles concernant la localisation. Pour ce faire nous utiliserons les métriques de l'information perçue et du taux de succès nous permettant de quantifier le niveau de location privacy via la caractérisation des utilisateurs. Nous tenterons de répondre à la question de savoir combien de données sur les utilisateurs sont nécessaires pour pouvoir en faire une bonne caractérisation et à quel point cette caractérisation permet d'identifier l'utilisateur qui a généré les données. Ces quantificateurs se servant de probabilités à posteriori, nous pourrons comparer plusieurs outils permettant d'en obtenir, à savoir le modèle par noyau, le modèle discret et celui par maximisation d'espérance ainsi que des méthodes permettant d'obtenir directement des pseudo-probabilités, telles que la méthode des K plus proches voisins et celle des arbres des décision. Différentes techniques de défense telles que l'agrégation de l'espace et le bruitage de données seront également testées afin d'essayer d'augmenter la location privacy. Les données utilisées seront premièrement des données simulées d'utilisateurs se trouvant dans l'espace à 2 dimensions et deuxièmement des données réelles d'utilisateurs présents sur la côte Est des Etats-Unis. Les résultats montrent que tous les outils d'analyse que nous considérons comme méthode d'attaque donnent une information perçue élevée dans les données simulées et surtout que le taux de succès atteint est très souvent maximal. Les données réelles n'offriront pas plus de location privacy et les résultats obtenus sur celles-ci semblent rester aussi convaincants que ceux des données simulées. Les moyens de défense permettent toutefois d'augmenter ce niveau de location privacy mais pas de façon miraculeuse à moins bien sûr d'altérer énormément les données. Au vu de ces résultats il semble évident que la réponse à notre question est malheureusement celle redoutée : oui, les données permettent bien trop facilement à un attaquant quelconque d'obtenir des informations privées quant aux individus se retrouvant dans une base de données de géolocalisation.

Chapitre 1

Introduction

A travers cette introduction nous parlerons tout d'abord du contexte et de la problématique dans lesquels s'inscrit notre mémoire. Nous présenterons les motivations qui nous ont poussés à traiter le sujet de la quantification d'anonymat dans les bases de données. Nous passerons ensuite en revue quelques métriques de la littérature visant à étudier cette thématique. Nous exposerons alors clairement nos objectifs et finirons en présentant l'organisation de ce qui va suivre.

1.1 Motivations

Dans cette première section nous exposerons les différents motifs qui nous ont motivés à entreprendre ce mémoire basé sur la quantification de l'"anonymat" dans les bases de données. Nous viserons à donner au lecteur un intérêt pour les différentes notions évoquées dans la suite de ce travail. Nous dénoncerons tout d'abord un problème lié à la vie privée qui semble aujourd'hui en danger au travers des bases de données, peu importe le sujet que ces dernières traitent. Pour ce faire nous utiliserons des exemples à la portée de tout lecteur afin qu'il puisse s'y identifier. Nous expliquerons ensuite les raisons de ce danger et nous nous dirigerons alors vers le domaine plus précis des données de géolocalisation et des dangers qu'elles recèlent.

Vie privée et anonymat, soyons alarmistes

Selon le dictionnaire Larousse [1], l'anonymat se définit comme l'"*État de quelqu'un, de quelque chose qui reste anonyme*". Un anonyme reste alors "*quelqu'un dont on ignore le nom*". La question que nous nous posons est la suivante : Avec le déploiement croissant des nouvelles technologies, avec les réseaux sociaux, applications et autres, est-il aujourd'hui encore possible de rester «anonyme» si on le désire ? En effet, il nous semble évident qu'un individu cherchera toujours à garder confidentielles certaines informations le concernant, comme ses dossiers médicaux par exemple. Chacun a droit à sa vie privée, et les lois sont de plus en plus strictes à cet égard. Il est donc important de savoir à quel point nos données restent confidentielles ou non, et si elles peuvent être ou non utilisées à notre insu. Cette interrogation, cette crainte, est aujourd'hui plus que légitime. Le nombre d'informations détenu sur des individus sans leur consentement éclairé est très probablement effrayant, et les exemples démontrant la violation de la vie privée ne manquent d'ailleurs pas. Un article du journal l'express [4] nous en cite quelques-uns, dont l'histoire d'un adolescent écrivant le nom de sa mère sur la barre de recherche du web et tombant sans difficulté sur des vidéos pornographiques de cette dernière. L'histoire d'un patron suivant l'historique de navigation d'un de ses employés et le faisant chanter suite aux visites successives de sites à caractères discutables est également dénoncée. Au-delà des exemples directement liés à l'Internet on retrouve également des exemples tels que celui d'une société de plomberie donnant des portables munis d'un GPS à ses commerciaux afin de les traquer, sans que ces derniers ne le sachent.

Il existe aujourd'hui un nombre incroyable d'outils capables de nous dérober des informations personnelles. A notre insu ? Pas forcément. Avec Google, les réseaux sociaux (tels que Facebook, Tweeter,...), les GPS, les puces, les caméras de surveillance, tant d'applications que nous utilisons consciemment, nous nous rendons consentants d'une avancée technologique donnant finalement raison au prophète George Orwell : "*Nous voilà plongés dans l'ère de la surveillance diffuse et des intrusions silencieuses*". Il est en effet assez aisé de se rendre compte que la population ne va pas à l'encontre de la nouvelle technologie. Dans les réseaux sociaux par exemple, on note aujourd'hui que sur les 7,4 milliards d'habitants, 3.42 milliards sont internautes (46%) et 2.31 milliards sont actifs sur les réseaux sociaux (31% de la population mondiale) [18]. Autres chiffres assez révélateurs, nous observons une évolution incessante de la vente des smart phones dans le monde, allant en 2017 jusqu'à 1.53 milliard de terminaux écoulés comme on peut le voir sur la figure 1.1 [7]. On note également une hausse des dépositions pour fraude à l'identité qui ne serait évidemment pas étrangère à l'évolution du nombre d'informations disponibles sur chacun. Nous observons en Belgique, en 2014, 4669 plaintes reçues pour fraude à l'identité, ce qui est une augmentation de 6% par rapport à l'année 2013 [6].

Il est donc plus que temps de se demander ce que nous acceptons de divulguer sur notre identité et dans quelles mesures. Tous ces exemples ne cherchent qu'à démontrer une menace bien réelle et très certainement déjà dévoilée, notamment via des films comme "Snowden" [1] qui aujourd'hui relatent des faits réels alarmistes.

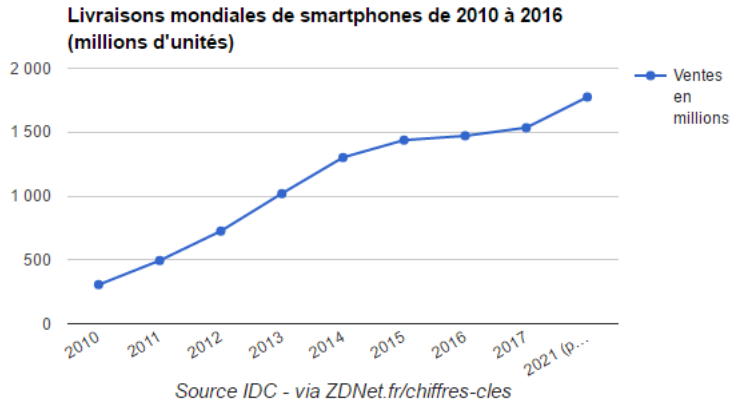


FIGURE 1.1 – [7]

L'émergence du Machine Learning

Il est important de se rendre compte que nous arrivons dans une technologie où même de simples actions de la vie de tous les jours peuvent donner à autrui des informations personnelles. Prenons l'exemple d'une grande surface ; en allant faire vos courses, vous ajoutez à leur base de données des informations sur vos besoins, vos habitudes d'achats. Grâce à cela, des analyses pourront être faites, permettant une démarche marketing ciblée, plus précise, allant même jusqu'à prédire certaines informations sur vous. C'est ici qu'est mise en avant l'avancée de l'intelligence artificielle, du big data, et surtout du machine learning. En effet, s'il n'existait pas d'outil capable de déchiffrer les données disponibles, il n'y aurait pas de menace à pareille échelle. Le machine learning, prenant rapidement son essor depuis quelques années, permet d'analyser énormément de base de données, et de prédire différents types de résultats. Selon l'encyclopédie illustrée

1. film germano-américain réalisé par Oliver Stone, sorti en 2016. Il revient sur les révélations faites par Edward Snowden sur la collecte illégale d'informations par la NSA sous prétexte de lutte antiterroriste.

du marketing [10], "*le terme de machine learning décrit un processus de fonctionnement d'un système d'intelligence artificielle par lequel le système est doté d'un système d'apprentissage*". C'est à dire que sur base de données réelles, le système apprendra et s'entrainera, pour ensuite tirer des conclusions et les appliquer sur des données futures. Margolaine Baratte [8] nous le décrit également comme étant "*la mise en place d'algorithmes ayant pour objectif d'obtenir une analyse prédictive à partir de données, dans un but précis. C'est en quelque sorte l'apprentissage par l'exemple*". Elle explique également que le machine learning ne sert non pas à trouver des liens de causalité mais plutôt à trouver des corrélations. On remarque par exemple que les bases de données de grandes surfaces ont décelé que les gens achetant des couches pour bébé sont plus enclins à acheter de la bière que la moyenne des consommateurs [33]. On remarque donc une corrélation entre "Achat de couches pour bébé" et "Achat de bières", sans pour autant savoir pourquoi (même si l'on peut imaginer que le père de famille ira moins souvent en bar et voudra donc avoir de la bière à proximité par exemple).

Avec cette phase d'**entrainement** sur une partie des données disponibles et cette seconde phase de **vérification**, à l'aide du reste des données disponibles, le machine learning permet aujourd'hui de mettre en pratique différents types d'analyses comme le montre la figure 1.2 :

- La classification : Permettant de classer de nouvelles données dans l'une ou l'autre classe prédéfinies (exemple : filtre spam, détection de fraude, classer des protéines selon leurs caractéristiques, etc) [38]
- Le clustering : Permettant de découper un ensemble d'observations en plusieurs sous ensembles appelés clusters tel que les observations d'un même cluster soient similaires en un sens [27]
- La régression : Permettant de prédire une valeur numérique et non plus une classe (exemple : la valeur d'un appartement, dépendant de son emplacement, sa superficie, etc) [28]
- ...

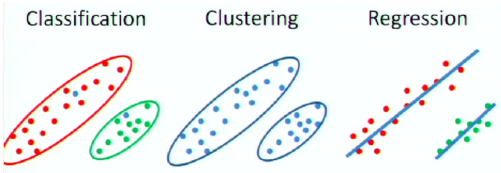


FIGURE 1.2 – [8]

Même si, comme le dit Loïc Knuchel [28], le machine learning n'est pas devenu un outil permettant de tout prédire, et qu'il nécessite des données "de qualité", c'est à dire desquelles on pourra tirer de l'information significative, on peut se rendre compte que la masse impressionnante de données collectées via la technologie d'aujourd'hui peut être utilisée à des fins précises. Cela s'avère d'ailleurs très utile dans beaucoup de domaines. Dans le marketing, les entreprises sauront quel produit proposer à qui, l'utilisateur de Netflix trouvera un avantage à se voir proposer des films qui lui correspondent personnellement. Ces avantages peuvent même aller jusqu'à aider la médecine : des scientifiques de Microsoft ont démontré qu'en analysant des échantillons de requêtes sur le moteur de recherches, ils étaient capables, dans certains cas, d'identifier qu'un utilisateur souffrait du cancer du pancréas, avant même qu'il n'ait reçu de diagnostic [34]. Cependant il demeure un côté sombre à toute cette évolution, dont nous parlons depuis le début de ce chapitre : qu'en est-il de notre anonymat ? Le machine learning pourrait-il aller au-delà de ce que les utilisateurs acceptent de partager ? Le machine learning reste-il dans le domaine de ce qui est autorisé par nos lois ? Et que se passerait-il si nos données tombaient entre les mains de personnes mal intentionnées, pourraient-elles en tirer profit ? Voilà plusieurs questions qui

donnent naissance à un sujet intéressant. En effet, on remarque déjà que l'avancée de l'intelligence artificielle remet en doute le droit de vie privée : Alex Lane [31] nous dit que Ph.D. candidate, Richard McPherson, Professor Vitaly Shmatikov et Reza Shokri ont analysés la précision avec laquelle des algorithmes basés sur le machine learning pouvaient identifier des gens brouillés (comme on peut le voir régulièrement à la télévision par exemple, afin de garder l'anonymat de la personne en question). Il s'avère que là où l'œil humain n'identifierait la personne qu'avec une précision de 0.19%, l'algorithme est capable de la reconnaître avec une précision de 71%, pouvant aller jusqu'à 82% si on lui donne cinq chances de l'identifier. Impressionnant, mais pas encore le plus interpellant. En effet, il est à noter que l'algorithme utilisé est créé avec une simplicité non négligeable, s'entraînant sur de simples échantillons obtenus via facebook par exemple ou sur des annuaires du personnel obtenus sur le web.

On peut vraiment se rendre compte que ce sujet est vaste, tant au niveau du type de données sur notre vie privée que l'on peut dévoiler que sur l'utilisation qu'il est possible d'en faire. Dès lors, nous avons trouvé impératif de limiter le domaine de notre analyse afin que celle-ci soit plus précise et le sujet traité plus spécifique.

La géolocalisation

Parmi toutes les données disponibles, ce mémoire visera principalement les données de géolocalisation. Citons [5] : "*pour la CNIL, les données de géolocalisation sont bien des « données à caractère personnel » selon la loi, dans la mesure où elles permettent une identification de l'utilisateur via le SSID (Service Set Identifier)*". Nous avons bien à faire à des bases de données récupérées en nombre via des applications, caméras, puces, GPS, et pouvant toucher à la vie privée de la population. Nous chercherons dans ce mémoire à analyser la quantité d'informations qu'une base de données sur la géolocalisation peut laisser transparaître. Il est important de se rendre compte que ce type de données est aujourd'hui presque "facile d'accès". On peut en recenser plusieurs causes :

1. L'évolution surprenante qu'a connu le domaine de la géolocalisation : au départ via l'observation du soleil et des étoiles, l'utilité de se localiser perdure depuis des siècles. Non seulement utile pour soi-même mais également pour communiquer, les moyens de localisation n'ont fait qu'évoluer. En passant par la fumée des feux pour donner sa position, la boussole et la triangulation radio, nous arrivons aujourd'hui aux GPS actuels permettant de se géolocaliser très facilement et avec une grande précision [2]. Enfin, avec l'arrivée des smart phones le GPS n'est plus un objet à part, il peut être intégré au téléphone et offre des services de géolocalisation portatifs intégrés [2].
2. Le nombre impressionnant d'outils nécessitant ce type d'informations pour fonctionner : donner sa géolocalisation permet par exemple d'obtenir des informations sur le trafic en temps réel, de dire dans quel endroit on se trouve pour prévenir ses amis et savoir où ceux-ci se trouvent, obtenir des informations sur les restaurants/cinémas à proximité de sa position, etc. Certaines chaines de fast-food donnent même des promotions et/ou articles gratuits si l'on partage notre position sur FourSquare [2], une application qui permet à l'utilisateur d'indiquer où il se trouve grâce à un système de géolocalisation et de recommander, ce faisant, des lieux de sorties. Il existe également des applications permettant aux parents de suivre la position du portable de leur enfant afin de ne pas s'inquiéter d'où il se trouve (AdoProtect [3] par exemple).
3. La négligence des gens quant aux dangers de dévoiler leurs coordonnées GPS : une étude faite par Colbert [15] démontre que les individus auxquels on pose des questions du type "Comment réagiriez-vous si les informations de votre position étaient obtenues, bien que

2. `https://fr.foursquare.com/` - consulté le 31/05 à 17h50
3. `http://www.adoprotect.com/` - consulté le 31/05 à 17h51

dans un certain contexte ?" donnent des réponses ne démontrant pas une inquiétude réelle. A tel point que l'auteur doute de la précision de ces résultats et ne les a même pas partagés. Une autre étude [30] a montré qu'en moyenne, sur 74 étudiants universitaires, le prix demandé pour donner leur géolocalisation serait entre 10 et 20 £ si c'est à des fins commerciales. Une valeur qui permet ici encore de montrer un intérêt peu développé pour celle-ci. Enfin, suite à une étude sur 16 participants utilisant des applications basées sur la localisation pendant 5 jours, Barkuus and Dey [9] disent trouver qu'en général, les gens ne sont pas vraiment concernés quant au sujet de leur confidentialité de localisation, lorsqu'ils utilisent ce type d'application.

Au vu de tout ce qui vient d'être dit, il semble assez évident que la mise en circulation de données touchant à la vie privée, sous forme de points géographiques se doit d'être protégée et surveillée, même si le consentement (peu éclairé) des utilisateurs permet de rester dans les limites de ce qu'autorise la loi. En effet, même si la géolocalisation a déjà pu sauver la vie d'un motard, retrouvé 4 jours après son accident grâce aux données précises stockées sur des serveurs Google [14], qu'elle a également déjà permis de retrouver les voleurs d'un Iphone grâce aux données GPS du téléphone [16] et qu'elle a sans doute rendu possible bien d'autres choses encore, continuons de garder en tête nos inquiétudes face à cette innovation. Car au-delà de ces bienfaits, la géolocalisation pourrait par exemple vous faire payer plus cher votre assurance vie si la firme de cette dernière savait que vous vous rendiez régulièrement au fastfood [45]. Mais encore, continuons de se souvenir que même pour des utilisateurs publiant des données de façon dite anonyme, des chercheurs ont pu trouver l'adresse postale précise de ceux-ci [22]. Il semble dès lors inutile de préciser que cela fera le bonheur des cambrioleurs qui disposeront dès lors des vos données GPS et sauront à quel créneau vous n'êtes pas à votre domicile... L'intimité de nos données personnelles demeure un de nos biens les plus précieux, et reste un des piliers de nos sociétés démocratiques, même si peu d'entre nous y attachent l'importance qu'elle mérite. Combien de fois n'entend-on pas dire « On peut tout savoir, puisque je n'ai rien à cacher ». C'est là ignorer profondément le fondement même de nos libertés défendues par les lois de nos démocraties.

Confidentialité de localisation (location privacy)

Selon A. J Blumberg et P. Eckersley[12], la "location privacy" est la capacité d'un individu de se déplacer dans un espace public sans que sa localisation ne soit systématiquement et secrètement enregistrée pour une utilisation ultérieure. Nous obtenons de Beresford et Stajano [11] une seconde définition spécifiant que la "location privacy" est la capacité d'empêcher d'autres personnes d'apprendre l'information actuelle ou passée de son emplacement. On peut en déduire comme point commun l'importance de garder confidentielles des informations concernant notre localisation.

Il est important de préciser que, toujours selon Beresford et Stajano [11], il s'agit plutôt ici de contrôle d'accès à cette information. C'est à dire qu'il ne faut pas nécessairement éviter brutalement toute utilisation ou enregistrement de nos coordonnées géographiques, mais qu'il faut être en mesure de les limiter, de les contrôler. En effet, afin de continuer à profiter des bienfaits de la géolocalisation cités ci-dessus, il semble intéressant de continuer à accepter de les dévoiler, mais dans une certaine mesure uniquement, et de façon contrôlée. Cette notion de contrôle est également appuyée par J. Krumm [30] qui précise qu'il faudrait pouvoir déterminer qui est en mesure de connaître ces données et donc qu'il est important de pouvoir les protéger.

Tout en restant dans cette notion de contrôle, Duckham et Kulik [20] parlent de la "location privacy" comme un type spécial d'information privée concernant le fait que les individus puissent déterminer eux-mêmes *quand*, *comment* et *dans quelles mesures*, les informations de leur géolocalisation sont communiquées aux autres. Cette définition est basée sur celle de Westin

[47] concernant la "privacy". En effet, il remarque que les gens accepteront généralement plus facilement de donner leur localisation si le :

- *Quand* correspond aux localisations passées plutôt que présentes ou futures
- *Comment* correspond à des requêtes et non pas à des envois de localisation automatiques
- *Dans quelles mesures* correspond à une zone globale telle une région plutôt qu'un point précis tel des coordonnées géographiques.

L'objectif serait donc de trouver un moyen de limiter les données géographiques dévoilées. Celles-ci ne devraient correspondre qu'aux données passées, signifiant qu'elles ne pourront pas permettre de prédire celles du futures (ce qui, de par l'émergence du machine learning et autres nouvelles techniques reste très discutable). Elles devraient notamment pouvoir être contrôlées quant à la façon dont ces données sont envoyées ; c'est à dire de pouvoir choisir quand celles-ci sont enregistrées (afin de permettre de ne pas dévoiler que l'on est à tel ou tel endroit si on ne le veut pas). Finalement, la précision de ces données doit également être limitée. Nous cherchons ainsi à minimiser l'information qu'on pourrait tirer de données géographiques sur les individus faisant partie de ces bases de données, tout en conservant l'utilisation et les bienfaits de la géolocalisation. Pour ce faire, il existe dans la littérature différentes techniques dont nous parlerons par la suite. Ces techniques permettent d'augmenter notre anonymat et notre niveau de vie privée aux dépens d'une perte de précision des données. Elles nous préservent des hackers et des fuites de données néfastes ou dangereuses pour la population.

Ces réflexions nous amènent à cette question, à laquelle il nous faut répondre pour pouvoir se lancer dans les analyses qui nous intéressent : Comment quantifier la location privacy ? En d'autres termes, comment pouvons-nous dire à quel point nous sommes identifiables géographiquement parlant ; ou encore, quelle quantité, quel type d'information une base de données dévoile-t-elle sur les individus qui y sont présents ? Différentes métriques présentées dans la littérature donnent réponse à ces questions. Il n'existe aujourd'hui pas de méthode systématique pour analyser et surtout quantifier les bases de données, ce qui rend le sujet plus vaste. Bien que le but premier de ce mémoire ne soit pas de comparer les différentes métriques de la littérature, nous en décrirons quelques-unes dans la section suivante afin de se rendre compte de leur diversité. Nous serons alors capable de mettre en avant les avantages de celles que nous avons choisies pour les analyses de ce mémoire.

1.2 État de l'art

Pouvoir quantifier la location privacy est nécessaire pour plusieurs types d'analyse. Par exemple afin de déterminer si tel ou tel mécanisme de défense est meilleur qu'un autre, pour analyser si une base de données a un haut ou bas niveau de location privacy, analyser la performance des différents types d'attaques possibles, etc. Selon Reza Shokri, Julien Freudiger, Murtuza Jadliwala, et Jean-Pierre Hubaux [40], les métriques de location privacy devraient être en mesure de capturer la quantité d'information qu'un adversaire a à propos des trajectoires ou des positions des utilisateurs. C'est à dire qu'elles doivent quantifier l'incapacité d'un adversaire à traquer les utilisateurs dans l'espace et dans le temps. Globalement on retrouve deux grands types de métriques : Les premières quantifient le niveau d'anonymat des utilisateurs dans la base de données en temps que telle. Elles définissent à quel point il est possible d'associer à une personne physique une des données de la base. Elles regroupent des métriques plus connues telles que le "k-anonymat" que nous présenterons dans cette section, la "l-diversity", etc [36] et des métriques moins connues comme par exemple une métrique basée sur le "clustering" dont nous parlerons également. Les secondes sont basées sur l'estimation de modèles obtenue à l'aide de la base de données. Elles quantifient alors le niveau de caractérisation des utilisateurs que l'on peut tirer de ces modèles. Dans ce second type de métriques on retrouve la "correctness", l'entropie

(aussi appelée "certainty"), l'information perçue, etc, dont nous parlerons aussi.
Cette section va donc nous permettre de présenter différentes métriques afin de voir les différences globales entre celles-ci et nous permettre d'en choisir une pour nos analyses. Il serait bien entendu intéressant de comparer par la suite les résultats obtenus à l'aide de l'une ou l'autre métrique mais ce point sort du cadre de notre mémoire.

k-anonymat

Le concept du k-anonymat a été introduit par Samarati [37] et Sweeny [44] et cherche à répondre à la question "Comment un détenteur de données peut-il les rendre publiques en garantissant que les individus contenus dans ces données ne puissent pas être identifiés, tout en gardant les données utiles ?". Des données bénéficieront d'une garantie de protection de k-anonymat si les informations de chaque personne contenues dans ces données sont non distinguables d'au moins $k-1$ autres personnes étant également présentes dans la base de données [40].

Si nous prenons par exemple des données médicales, Gedik, Bugra et Liu, Ling [24] expliquent que même si les noms des patients sont remplacés par des identifiants ou tout simplement supprimés, il reste néanmoins des ensembles de critères (nommés "quasi-identifier") qui peuvent permettre de distinguer une seule personne parmi les autres. Si nous prenons l'ensemble des critères comme étant l'ensemble {âge, lieu de domicile, sexe} et que nous utilisons d'autres bases de données publiques disponibles facilement, on pourrait alors mettre en évidence la personne correspondant à ces critères et obtenir les informations médicales de celle-ci. Dès lors la privacy n'est plus assurée. C'est ici que le k-anonymat entre en jeu puisqu'il évite que ce genre de problème n'arrive. Par exemple, dans la table de l'image 1.3, le 3-anonymat est assuré et on sera au moins non distinguable avec 2 autres personnes car l'ensemble {Activité, Age} ne permet de mettre en évidence que des groupes d'au moins 3 personnes.

Activité	Age	Diag
"M2"	[22, 23]	Grippe
"M2"	[22, 23]	Rhume
"M2"	[22, 23]	Rhume
"Etudiant"	[20, 21]	VIH
"Etudiant"	[20, 21]	Grippe
"Etudiant"	[20, 21]	Allergie
"Ens."	[24, 27]	Cancer
"Ens."	[24, 27]	Cancer
"Ens."	[24, 27]	Cancer

FIGURE 1.3 – 3-anonymat,[36]

Gruteser [25] étend ce concept à celui de la localisation et des "Llocation-based services" (LBS's), en ce sens qu'un individu est considéré comme étant k-anonyme au niveau de sa localisation si et seulement si l'information de localisation du téléphone d'un client à une LBS est non distinguable d'au moins $k-1$ informations de localisation venant de téléphones d'autres clients.

Métrique basée sur le clustering

Fischer, Lars and Katzenbeisser, Stefan and Eckert, Claudia [23] présentent une métrique basée sur le clustering. Plus précisément, elle se base sur une mesure de non-associativité. La non-associativité est définie comme étant l'incapacité d'un observateur de décider si certains éléments d'intérêt sont liés ou non. Dès lors, cela fait bien référence à la technique du clustering permettant de trouver des regroupements parmi des ensemble de points. La métrique est basée

sur le succès d'un attaquant à retrouver à quel utilisateur appartient telle ou telle donnée/tel ou tel groupe de données.

Du point de vue de l'attaquant, l'objectif sera ici de séparer les données en un nombre n de clusters sachant que chaque cluster correspond à un utilisateur (il y a donc n utilisateurs présents dans la base de données). Évidemment, l'attaquant ne connaît pas la division réelle des données, il va dès lors faire l'hypothèse que c'est l'une ou l'autre partition de manière probabiliste. Prenons par exemple un ensemble d'observations $\mathcal{O} = \{o_1, o_2, o_3, o_4, o_5\}$ à partitionner en 3 clusters pour 3 utilisateurs $\{u_1, u_2, u_3\}$. Prenons un ensemble de partitions possibles noté $\pi_\mathcal{O} = \{\pi_1, \pi_2, \pi_3\}$ où les partitions sont celles illustrées sur la figure 1.4. L'attaquant va définir une fonction de probabilité $P(\pi)$ pour $\pi \in \pi_\mathcal{O}$ définissant la probabilité que la partition π soit selon lui la bonne. Par exemple $P(\pi_1) = 0.5$, $P(\pi_2) = 0.2$, $P(\pi_3) = 0.3$.

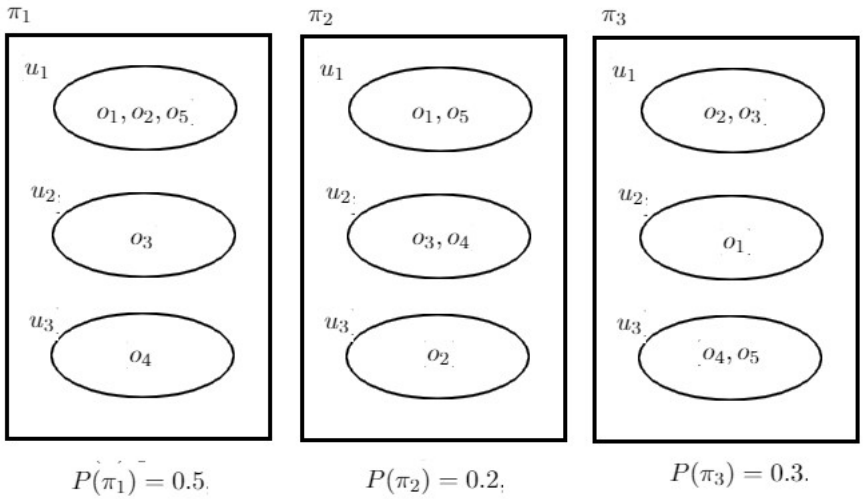


FIGURE 1.4 – Exemple de partitions possibles et probabilités liées à celles-ci obtenues par un attaquant

Pour mesurer la location privacy, il reste alors à comparer ces partitions avec celle qui est correcte. L'erreur entre un partitionnement π_i et la partition réelle est calculée sur base d'une mesure de dissemblance qui ne sera pas expliquée en détail ici. Si nous notons δ_π l'erreur entre la partition π et la partition réelle, la mesure de location privacy peut alors être définie comme suit :

$$\sum_{\pi \in \pi_\mathcal{O}} P(\pi) \cdot \delta_\pi$$

Si un attaquant donne une probabilité de 1 à la bonne partition π_{exacte} de sorte que $\delta_{\pi_{exacte}} = 0$, la mesure donnera bien une location privacy nulle, ce qui semble logique puisque les données ont été correctement attribuées.

La "correctness"

Une des métriques utilisée dans la littérature est celle de la "correctness", expliquée plus en détail par Shokri et al. [41]. Cette mesure sert à quantifier à quel point un attaquant obtient une solution correcte ou non, suivant ce qu'il cherche à faire. Prenons comme exemple un attaquant ayant comme objectif de pouvoir déterminer l'emplacement d'un utilisateur u à un temps t, grâce à une série d'observations o. La "correctness" sera définie comme l'erreur entre l'emplacement trouvé pour u au temps t, que nous noterons x et l'emplacement réel de cet utilisateur à ce

moment là, que nous noterons x_c. Dès lors, la valeur de "correctness" sera égale à

$$\sum_x \hat{Pr}(x|o) \cdot ||x - x_c||$$

avec $\hat{Pr}(x|o)$ la fonction probabiliste trouvée par l'attaquant cherchant à estimer le résultat x et $|| \cdot ||$ la norme. Il est à noter que cette formule comprenant une norme n'est utilisable que s'il existe une notion de distance entre x et x_c. Dans notre cas d'analyse par exemple, l'objectif est de retrouver à quel utilisateur appartient une observation o. Il n'y a dès lors pas de notion de distance entre 2 utilisateurs puisque soit on retrouve l'utilisateur correct, soit pas. C'est à dire que $||u - u_c|| = 1$ si $u = u_c$ et 0 sinon. La formule devient alors

$$\sum_u \hat{Pr}(u|o) \cdot ||u - u_c|| = \hat{Pr}(u_c|o)$$

avec u_c l'utilisateur correspondant à l'observation o.

Grace à cette métrique, on quantifiera la location privacy de la façon suivante : Au plus la correctness est élevée, au moins la location privacy n'est élevée.

L'entropie

Également énoncée par Shokri et al. [41] pour quantifier la "certainty", l'entropie sert à montrer à quel point la distribution obtenue par l'attaquant est uniforme/concentrée ou non. En se basant sur les notations décrites dans la "correctness", il s'agit de l'entropie de la fonction $\hat{Pr}(x|o)$. Avec l'entropie, nous serons en mesure de dire à quel point il est facile ou non de sélectionner un unique point de sortie x. Dans le cas de la location privacy il s'agit donc de dire si une observation sera, selon l'attaquant, fortement liée à un unique utilisateur ou si elle semble pouvoir appartenir à l'un ou l'autre de façon incertaine. Si la fonction est un Dirac par exemple, la certainty sera maximale puisque la sortie est unique. Si la fonction est très uniforme il sera plus difficile d'obtenir une réponse dite "certaine". Nous pouvons donc conclure qu'au plus l'entropie de la distribution $\hat{Pr}(x|o)$ est élevée au moins la "certainty" de l'adversaire ne l'est, car la distribution sera très uniforme. L'entropie se note H et se calcule comme suit :

$$\hat{H}(\mathcal{X}) = \sum_x \hat{Pr}(x|o) \cdot \log(\frac{1}{\hat{Pr}(x|o)})$$

Dans notre cas d'analyse où l'on cherche à attribuer une observation o à un utilisateur u, l'entropie s'écrit alors

$$\hat{H}(\mathcal{U}) = \sum_u \hat{Pr}(u|o) \cdot \log(\frac{1}{\hat{Pr}(u|o)})$$

L'information perçue

On retrouve dans l'article [42] une mesure de location privacy appelée "Information perçue". Les données présentent dans une base sont échantillonnées à partir de distributions réelles de variables aléatoire. L'information perçue va alors chercher à quantifier la capacité des données à représenter correctement ces distributions. Au plus les données sont représentatives des distributions réelles, au plus l'information perçue est élevée et au moins la location privacy l'est. En effet s'il est possible, à l'aide de la base, de retrouver les répartitions réelles des utilisateurs présents dans celle-ci, on sera en mesure de prédire leurs déplacements et donc de les identifier.

Cette dernière métrique sera celle que nous utiliserons au travers des analyses de ce mémoire, en parallèle avec le *taux de succès* présenté dans [43] et qui se base sur une idée similaire. Elles seront toutes deux expliquées plus en détail dans le chapitre 3 en mettant en avant les

avantages qu'elles présentend. Comme nous l'avons rapidement énoncé dans ce chapitre, le choix de la métrique de quantification n'est pas notre intérêt premier. La présentation des différentes métriques permet de se rendre compte de la diversité des manières de quantifier l'anonymat qui existent. Elle nous permettra d'argumenter brièvement dans le chapitre suivant le choix de la métrique de l'information perçue qui permet d'atteindre nos objectifs, au même titre que l'auraient pu d'autres métriques.

1.3 Objectifs

Il a déjà été prouvé dans la littérature que les données de géolocalisation peuvent donner beaucoup d'informations sur les utilisateurs et ce même avec très peu de données. De Montjoye Yves-Alexandre, Hidalgo César A., Verleysen Michel et Blondel Vincent D. [19] ont analysé 15 mois de données de géolocalisation de 1.5 millions d'individus et ont cherché à savoir si les déplacements, les traces des utilisateurs étaient uniques ou non. Il a été conclu qu'avec seulement quatre points choisis aléatoirement, c'est à dire quatre données de géolocalisation par utilisateur, il était possible de caractériser de façon unique les traces de 95% des utilisateurs. Il semble donc nécessaire de protéger les informations personnelles et privées des individus se retrouvant dans des bases de données de géolocalisation.

Dans le même état d'esprit, ce mémoire aura pour premier objectif de confirmer ces résultats en analysant un ensemble de traces d'utilisateurs présents sur la côte Est des États-Unis via une autre approche et une métrique en particulier. Nous analyserons ces données de localisation en tant qu'"attaquant" essayant d'obtenir des informations personnelles sur les individus. Par "attaquant", comprenons que nous aurons accès à une base de et que nous essayerons d'en tirer le plus d'information possible. L'objectif à atteindre est dès lors de définir à partir de combien de coordonnées géographiques observées dans la vie d'un des utilisateurs il est possible de le caractériser, de le définir dans la base de données et de prédire quels ensembles d'emplacements lui correspondent. Au vu des conclusions énoncées dans la paragraphe précédent, il semblerait que les résultats aient de grandes chances d'être alarmants. En conséquence, nous aurons pour second objectif d'augmenter le niveau de location privacy en utilisant des techniques de défenses telles que l'agrégation des positions ou l'ajout d'un bruit sur les observations. Nous pourrons alors analyser l'efficacité de ces techniques de défense. C'est à dire que nous regarderons à nouveau le nombre d'observations nécessaire pour identifier les utilisateurs et nous pourrons voir si les techniques de défense permettent d'augmenter le caractère privé des données.
Ces moyens de défense ayant de bonnes chances d'augmenter le niveau de location privacy, il est intéressant de comparer le degré de dénaturation qu'on inflige aux données avec l'anonymat qui en découle. Précisons que les jugements que l'on peut porter quant à l'utilité des données dénaturées sont purement subjectifs étant donné que cette utilité dépend de l'utilisation qu'on veut pouvoir en faire.

Comme nous l'avons précisé dans la section précédente, nous utiliserons une métrique particulière pour remplir ces objectifs : l'information perçue. Précisons d'emblée que ce choix n'est pas primordial et que d'autres métriques, similaires ou non, pourraient tout aussi bien convenir pour mener à bien les analyses envisagées. Néanmoins, l'une des particularités de la métrique choisie est d'utiliser les densités estimées sous formes de probabilités. Comme nous le verrons dans le chapitre suivant, il existe différentes méthodes permettant de déterminer des probabilités ou des pseudos-probabilités. Nous ferons alors une analyse comparative de plusieurs de ces méthodes sur des données simulées dans un premier temps et sur les données réelles brutes puis "protégées" dans un deuxième temps. Nous serons alors capable de deux choses : savoir quelle méthode est la plus efficace et dans quelle circonstances ; et déterminer le niveau de location privacy sur base de la meilleure forme d'attaque possible. En effet, un niveau de location privacy

basé sur une méthode d'attaque médiocre de sera pas représentatif du niveau réel.

1.4 Plan

La première partie sera consacrée à la description des données expérimentales et à l'introduction des notations qui seront utilisées tout au long de ce mémoire.

On présente ensuite les outils d'analyse en commençant par une description plus approfondie des métriques utilisées et de leur utilité. On y retrouve aussi la description des différentes méthodes de machine learning que nous utilisons pour obtenir les probabilités (ou pseudo-probabilités) utilisées dans la métrique choisie ainsi que des méthodes de protection des données envisagées. Finalement, on termine par une discussion concernant un problème qui survient quand on utilise cette métrique.

Une fois tous ces outils présentés on peut entamer les deux grosses parties de ce mémoire : l'analyse comparative des méthodes de machine learning dans le cadre de la location privacy sur des données simulées et l'analyse de la base de données réelles.
Dans la première partie nous passerons en revue chacune des méthodes utilisées et tacherons de les rendre les plus performantes possibles en optimisant leurs paramètres. Nous comparerons alors sur différentes bases de données simulées les performances de ces méthodes.
Dans la seconde partie nous utiliserons les méthodes qui se seront avérées les plus efficaces dans la première afin quantifier la location privacy tout d'abord dans les données réelles brutes, puis sur ces mêmes données après application d'un moyen de défense. Nous serons alors en mesure de répondre à nos objectifs et de pouvoir déterminer s'il existe des méthodes d'attaque plus performantes que d'autres, ce en fonction de la protection utilisée, ainsi que de savoir à quel point il est possible de sécurisées les bases de données de géolocalisation et à quel prix.

Chapitre 2

Données expérimentales et notations

Ce chapitre décrit en détail les données sur lesquelles se basent nos analyses, avec deux parties bien distinctes puisque les bases de données utilisées sont d'une part simulées et d'autre part réelles. Dans la première partie, nous expliquerons en quoi les données simulées sont utiles et comment elles sont générées. La seconde partie parlera des données réelles, expliquant d'où elles proviennent, ce qu'elles représentent et en quoi elles sont intéressantes. Un court résumé des notations sera alors présenté comme conclusion de ce chapitre.

Données simulées

Utilité : Les données simulées sont utilisées afin de pouvoir analyser en détail les résultats, de façon plus théorique. Elles nous permettent également d'avoir une connaissance parfaite de la base de données utilisée, c'est à dire connaitre la quantité théorique d'information qu'elle contient. Sur base de cette quantité connue d'information disponible, nous sommes alors en mesure de définir si nos techniques d'attaques sont ou non efficaces, dans le sens où la quantité d'information retirée est proche ou non de celle disponible.

Type et format des données : Cette base contient des données concernant un nombre défini d'utilisateurs, noté n et pouvant varier selon le type d'analyse visée. Chaque utilisateur est identifiable grâce à un identifiant unique, les identifiants étant définis comme allant de 1 à n. Ces utilisateurs sont destinés à se déplacer dans l'espace et à se retrouver à certains endroits notés o pour *observations*. Ces observations sont définies comme des coordonnées dans l'espace à deux dimensions $\mathbb{R}^2$ comme le sont les coordonnées géographiques $(latitude, longitude)$. Ici, les coordonnées ne représentent pas réellement des points géographiques en ce sens que nous ne nous soucions pas du point qu'elles représentent sur la carte. Ces observations sont liées à un utilisateur ayant visité cet endroit dans l'espace et sont au nombre de m observations par utilisateur. Ce nombre, similaire pour tous les utilisateurs, est également variable afin de pouvoir analyser le comportement d'un attaquant en fonction du nombre d'observations disponibles. Dès lors, la base de données est une matrice de dimensions $m \times 2 \times n$ qui correspondent respectivement au numéro de l'observation, la longitude ou latitude et à l'identifiant de l'utilisateur.

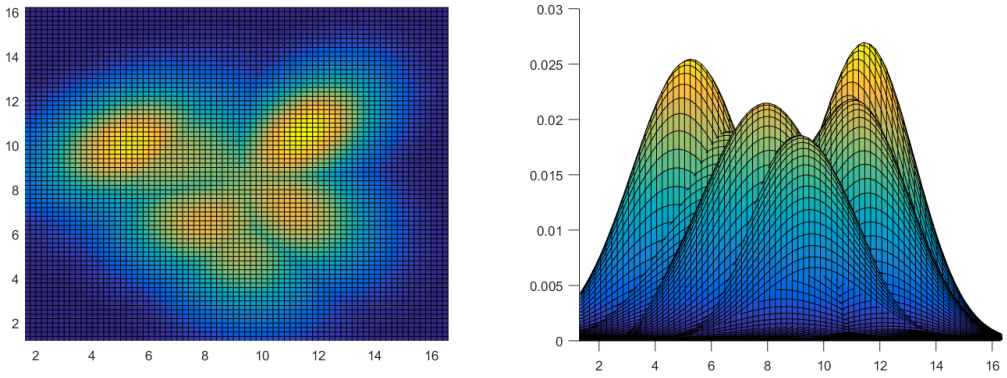
Comment sont générées les observations : Il faut tout d'abord choisir le type de distribution que vont suivre les observations des utilisateurs. Dans notre cas, nous avons choisi des mixtures de gaussiennes puisque cette hypothèse nous semble plausible dans la réalité. En effet, on peut imaginer que les utilisateurs sont centrés principalement sur deux zones : leur domicile et leur lieu de travail ; ces deux endroits seront les moyennes et donc les centres des gaussiennes. Les utilisateurs visitent également différents autres lieux plus ou moins éloignés de ces deux moyennes. Ces lieux sont visités avec une fréquence proportionnelle à la distance entre les centres des gaussiennes et les lieux en question. On peut aussi imaginer un nombre plus

important de centres en rajoutant par exemple une gaussienne centrée chez un membre de la famille proche, une maitresse, etc.

On choisit donc en premier lieu le nombre g de gaussiennes, que l'on fixe arbitrairement égal pour chaque utilisateur. Ensuite, on choisit aléatoirement g moyennes et g covariances liées à ces gaussiennes. Dans notre cas nous fixerons le nombre g de gaussiennes à 2. Les moyennes et variances sont aléatoires mais sont limitées à un certain intervalle défini. Enfin, on choisit aléatoirement la proportion de chaque gaussienne dans les distributions.

Une fois ces paramètres définis, les distributions le sont aussi et nous n'avons plus qu'à tirer un nombre m d'observations suivant la distribution de chaque utilisateur.

Il est important de préciser que durant l'entièreté des analyses basées sur des données simulées, nous utilisons un seul et unique ensemble d'observations que nous avons généré de la manière décrite ci-dessus. Ces observations sont tirées des gaussiennes visibles sur la figure 2.1. Afin de valider les résultats que nous obtiendrons nous feront également une comparaison entre ces derniers et ceux obtenus sur base d'observations issues de différentes gaussiennes simulées .



(a) vue aérienne (b) vue de profil

FIGURE 2.1 – Distributions simulées des utilisateurs (3 utilisateurs avec chacun une mixture à 2 gaussiennes)

Données réelles

Utilité : Ces données réelles vont permettre de comparer les résultats obtenus sur les données simulées avec ceux obtenus sur celles-ci. En effet, un attaquant sachant retrouver des utilisateurs simulés mais pas des utilisateurs réels n'est pas un bon attaquant. Dès lors les résultats obtenus sur base de données simulées ne sont intéressants que s'ils sont confirmés sur base de données réelles. Dans ces données aucune hypothèse n'est faite sur les déplacements des utilisateurs, qui n'agissent pas forcément selon une distribution prédéfinie, ce qui complique la tâche. Il est ainsi fort probable qu'on retrouve la présence de valeurs aberrantes ("outliers") dans des données réelles, c'est à dire d'observations concernant un individu auxquelles on ne s'attendrait pas (si celui-ci va exceptionnellement dans un magasin loin de chez lui par exemple). Nous pourrons avec ces données réelles nous rendre compte de la pertinence de nos résultats théoriques.

Type et format des données : Les données émanent de Stanford [32]. Elles ont été collectées via Brightkite, une application basée sur la localisation. Elle permet aux utilisateurs

de s'enregistrer à un endroit, un lieu et de voir quel autre utilisateur est proche de lui ou de savoir qui a déjà été là auparavant. Les données utilisées sont un sous ensemble de cette base de données, celles de la côte Est des États-Unis. On y trouve 135 utilisateurs liés à un identifiant et 216677 observations données sous forme de coordonnées géographique $(latitude, longitude)$. Ces 216677 observations sont chacune liée à un des identifiants mais ne sont pas réparties de façon égale entre ces derniers. Cette base de données sera retravaillée en fonction des analyses à effectuer c'est à dire qu'elle sera transformée dans le même format que celui des données simulées où chaque utilisateur a un nombre fixe m d'observations lui étant propres. Nous utiliserons pour cela m observations aléatoirement choisies parmi les observations des utilisateurs en disposant d'au moins ce nombre.

Notations

Nos bases de données sont donc composées d'un ensemble $\mathcal{U} = \{u_1, u_2, u_3, ..., u_n\}$ d'utilisateurs ; n dépendant de la base de données utilisée. Chaque utilisateur $u_i \in \mathcal{U}$ possède un ensemble $\mathcal{O}_{u_i} = \{o_1^{u_i}, o_2^{u_i}, o_3^{u_i}, ..., o_m^{u_i}\}$ d'observations ; m dépendant égalementde la base de données utilisée. Dans certains cas où il n'y a pas d'ambiguïté possible, celles-ci seront notées $\{o_1, o_2, o_3, ..., o_m\}$. L'ensemble des observations de tous les utilisateurs se note alors $\mathcal{O}$. Chaque utilisateur $u_i \in \mathcal{U}$ génère ses observations selon une fonction de densité $f_{u_i}(o)$, $o \in \mathbb{R}^2$ dans le cas continu ou une fonction de probabilité $p_{u_i}(o)$, $o \in \mathbb{Z}^2$ dans le cas discret. Un exemple de base de données est illustré par la table 2.1.

Table 2.1 – Exemple de base de données

	o_1		o_2		...		o_m	
	Lat	Long	Lat	Long	Lat	Long	Lat	Long
u_1	1.4	6.2	2	6			1.7	7.3
u_2	1.9	3	0.4	10			-0.6	3.5
...								
u_n	1.7	3.2	7.3	4			3	4.2

Chapitre 3

Outils d'analyse

Ce chapitre décrit les différents aspects techniques utilisés dans l'analyse des données des sections suivantes. Il se veut donc être une véritable boîte à outils permettant la compréhension de ce qui va suivre.

3.1 Métriques utilisées

La métrique principale que nous avons choisi d'utiliser dans nos analyses est celle de l'*information perçue*. Cette métrique a déjà été brièvement expliquée dans le premier chapitre mais elle sera expliquée en détail dans cette section. Précisons qu'à travers cet outil, les données seront dites utiles si elles représentent correctement les fonctions de distribution dont elles sont échantillonnées. D'autres métriques seront utilisées en parallèle de celle-ci : l'*information mutuelle* et le *taux de succès*, toutes deux expliquées dans cette section. Une fois ces termes définis nous montrerons ce que ces outils permettent d'obtenir comme type d'analyse et en quoi elles sont intéressantes.

Information Mutuelle

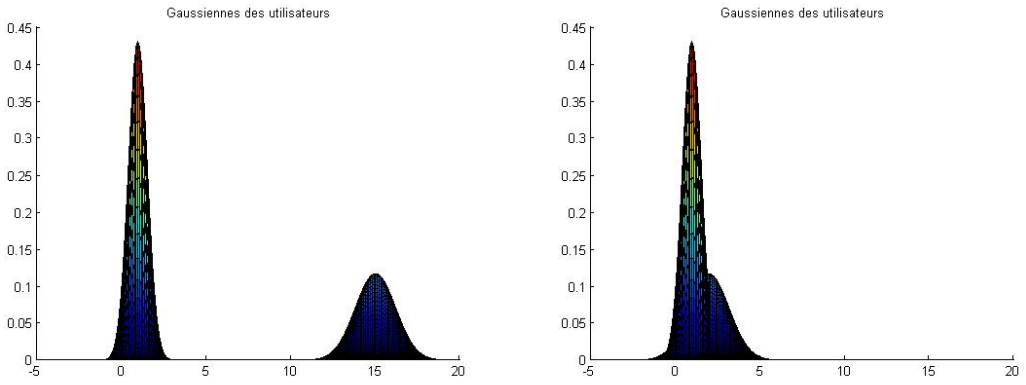
L'information mutuelle représente la quantité maximale d'information que l'on peut retirer des observations, compte tenu des distributions réelles que suivent les observations des utilisateurs. En utilisant les notations exposées dans le chapitre 2 c'est à dire n pour le nombre d'utilisateurs et m pour le nombre d'observations par utilisateur, l'information mutuelle se calcule comme suit (dans le cas discret et continu) :

$$MI(\mathcal{U}, \mathcal{O}) = H[\mathcal{U}] + \sum_{i=1}^{n} Pr[u_i] \sum_{j=1}^{m} p(o_j^{u_i}|u_i) \cdot \log_2(Pr[u_i|o_j^{u_i}])$$

$$MI(\mathcal{U}, \mathcal{O}) = H[\mathcal{U}] + \sum_{i=1}^{n} Pr[u_i] \int f(o^{u_i}|u_i) \cdot \log_2(Pr[u_i|o^{u_i}]) do^{u_i}$$

On retrouve dans cette formule l'entropie H dont il a déjà été question dans le chapitre 1. Dans la métrique basée sur l'entropie, on calculait l'entropie $\hat{\mathcal{U}}$ de la distribution estimée $\hat{p}(u|o)$ et au plus celle-ci était élevée, au plus la location privacy l'était. Ici l'entropie est calculée via la distribution a priori des utilisateurs et sert à définir la quantité maximale d'information que l'on pourrait retirer des données. Il n'est donc pas question de distribution estimée ici. Il est à noter que cette distribution sera dans le cadre de notre mémoire définie comme étant uniforme : $Pr[u] = 1/n$ puisque les utilisateurs ont tous le même nombre d'observations. Nous obtenons alors une entropie de $H[\mathcal{U}] = \log_2(n)$.

La somme additionnée à l'entropie dans la définition de l'information mutuelle sera un terme négatif ou au maximum nul puisque $Pr[u_i|o_j] \in [0, 1]$. Ce terme illustre globalement à quel point il est facile ou non de lier des observations à leur utilisateur, ce qui dépend des distributions réelles $f(o|u)/p(o|u)$. Au plus celles-ci seront séparées dans l'espace au plus on saura identifier les utilisateurs. A l'inverse, au plus celles-ci seront confondues au plus il sera difficile de les identifier et nous perdrons donc plus d'information. Dans la figure 3.1a par exemple les utilisateurs sont parfaitement reconnaissables et l'information mutuelle sera dès lors égale à l'entropie $H[\mathcal{U}]$. Dans la figure 3.1b par contre l'information mutuelle sera plus faible.



(a) $MI(\mathcal{U}, \mathcal{O}) = H[\mathcal{U}]$ (b) $MI(\mathcal{U}, \mathcal{O}) < H[\mathcal{U}]$

FIGURE 3.1 – Distributions réelles de 2 utilisateurs

Précisons tout de même que cet outil d'analyse est uniquement théorique : Dans le cas pratique, les distributions sont inconnues et cette métrique ne peut donc pas être calculée. Cependant, dans le cas des données simulées, ces distributions sont connues et on peut donc calculer l'information mutuelle sur base de celles-ci et de la formule de Bayes (donnée ici dans le cas continu) :

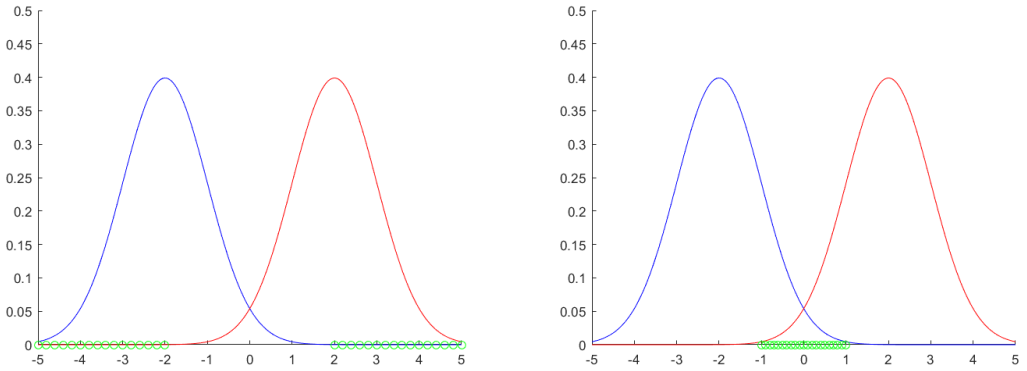
$$Pr[u|o] = \frac{f(o|u)}{\sum_{u^* \in \mathcal{U}} f(o|u^*)}$$

Information Mutuelle par sampling

Il est possible d'approximer l'information mutuelle au moyen d'une méthode dite de "sampling". Cette méthode consiste à employer la formule de l'information perçue qui sera expliquée ci-après, à cela près que l'on n'effectue pas de validation croisée (pas de séparation en deux sous groupes "learning|testing" des données puisque le modèle est connu). Le terme de validation croisée est également expliqué dans la suite de ce mémoire.

Cette méthode de sampling se base, sur les observations disponibles et plus précisément sur leur probabilité réelle plutôt que sur la distribution en entier, comme le fait l'information mutuelle. Cette métrique est donc "biaisée" par la position des observations. En effet, si celles-ci ne représentent pas parfaitement les distributions, on pourrait retirer plus ou moins d'information que ce que contiennent réellement les distributions. Prenons l'exemple de la figure 3.2. On pourrait avoir la chance de n'avoir aucune observation là où les distributions se superposent (3.2a). Étant donné que ce type d'observations ne fait pas diminuer l'information mutuelle par sampling, celle-ci serait alors supérieure à la valeur de l'information mutuelle. A contrario, si les observations étaient malheureusement situées uniquement là où les gaussiennes se chevauchent (3.2b), l'information mutuelle par sampling serait inférieure à la valeur de l'information mutuelle, étant donné que c'est précisément ces observations qui font diminuer l'information mutuelle.
Comme nous le verrons dans la section suivante, l'information perçue utilise elle aussi les

observations et non pas les distributions. On pourra donc retrouver ce même biais induit par les observations et parfois obtenir "plus d'information qu'il n'en existe".



(a) MI par sampling élevée (b) MI par sampling faible

FIGURE 3.2 – Répartitions potentielles des observations et influence sur l'information mutuelle par sampling. Les observations sont représentées en vert sur l'axe horizontal

Information Perçue

L'information mutuelle est, comme nous l'avons mentionné, très théorique. Elle ne pourra pas permettre une bonne analyse de la location privacy puisqu'elle nécessite de connaître la distribution réelle des utilisateurs. Dans un cas réel nous ne disposons que des données que nous cherchons à analyser et certainement pas des distributions qu'elles suivent. Il faut alors utiliser un autre outil d'analyse ne se basant que sur la base de données elle-même : l'information perçue.

L'information perçue représente la quantité d'information que l'on peut retirer des observations, compte tenu des distributions cette fois estimées que suivent les observations des utilisateurs. Son analyse permet d'avoir une idée de la qualité du modèle qu'on établit et surtout de la convergence de celui-ci en fonction du nombre d'observations disponibles. Alors que l'information mutuelle ne peut le faire, cet outil permettra de donner une mesure sur la quantité d'information qu'un attaquant, ne connaissant rien de plus que les données qu'il a à sa disposition, peut obtenir. C'est en ce point qu'il est intéressant puisqu'on sera dès lors capable de tirer des conclusions sur le niveau réel de location privacy des données.

Comme on vient de le souligner, dans le calcul de l'information perçue les distributions $p(o|u)$ (resp. $f(o|u)$ dans le cas continu) réelles ne sont pas connues. Il faudra donc les approximer afin d'obtenir une estimation de celles-ci que nous noterons $\hat{p}(o|u)$ (resp. $\hat{f}(o|u)$). Pour ce faire, nous retrouverons ici 2 phases connues dans l'usage du machine learning : une phase de création de modèle pour chaque utilisateur et une phase de test de celui-ci. Les données seront donc séparées en 2 parties. Sachant que nous disposons de m observations par utilisateur, l'une regroupera m_l données par utilisateur servant à créer un modèle noté nous le rappelons $\hat{p}(o|u)$ (resp. $\hat{f}(o|u)$) et l'autre regroupera m_t données par utilisateur servant à tester le modèle. L'ensemble des données d'apprentissage est noté $\mathcal{O}_l$ et les n données d'apprentissage de chaque utilisateur alors que l'ensemble des données de test est noté $\mathcal{O}_t$ et reprend l'ensemble des données de test de chaque utilisateur. Ces ensembles sont choisis de sorte que $\mathcal{O}_l \cap \mathcal{O}_t = \emptyset$ et seront définis à plusieurs reprises afin d'effectuer une validation croisée que nous expliquerons plus tard dans ce chapitre.

En utilisant les notations exposées dans le chapitre 2, l'information perçue se calcule comme

suit :

$$PI(\mathcal{U}, \mathcal{O}) = H[\mathcal{U}] + \sum_{i=1}^{n} Pr[u_i] \sum_{j=1}^{m_t} \frac{1}{m_t} \cdot \log_2(\hat{Pr}[u_i|o_j^{u_i}])$$

Comme pour l'information mutuelle, $\hat{Pr}[u_i|o_j^{u_i}]$ est calculé via la formule de Bayes, se servant ici des modèles estimés $\hat{p}(o|u)$ (resp. $\hat{f}(o|u)$).

La valeur de l'information perçue peut au maximum atteindre la valeur de l'information mutuelle. Dans ce cas cela signifie que les distributions ont été approximées de manière parfaite. Au plus les modèles se rapprochent des distributions réelles, c'est à dire au plus $\hat{p}(o|u)$ (resp. $\hat{f}(o|u)$) est proche de $p(o|u)$ (resp. $f(o|u)$), au plus l'information perçue tendra vers l'information mutuelle. Dans ce cas on sait retirer toute l'information présente dans la base de données et la location privacy est du niveau de celle évaluée grâce à l'information mutuelle (dès lors que celle-ci peut être calculée). En d'autres mots, au plus l'information perçue est grande, au plus on tire d'information de la base de données, au mieux on caractérise les utilisateurs et au moins la location privacy est élevée. Bien sûr, comme nous venons de le voir ce niveau dépend de la précision d'estimation de $\hat{p}(o|u)$ (resp. $\hat{f}(o|u)$). Une estimation de modèle médiocre nous rapportera une information perçue du même similaire, alors qu'il est peut être possible d'obtenir une estimation plus précise et donc une information perçue plus élevée. Il est dès lors utile de comparer plusieurs techniques d'approximation afin d'obtenir un degré de location privacy le plus représentatif possible.

Au contraire des métriques telles que le k-anonymat permettant de savoir si un utilisateur est distinguable ou non dans une certaine base de données, l'information perçue s'intéresse, à l'aide d'une certaine base de données, à la caractérisation des utilisateurs. Elle permet d'obtenir l'analyse des distributions estimées $\hat{Pr}[u|o]$ caractérisant chaque utilisateur et permettant de prédire par la suite des informations sur ceux-ci. Au plus l'utilisateur est bien caractérisé, au plus l'information perçue est élevée. Le plus gros avantage de cette métrique est l'existence d'une analyse de convergence liée au nombre de données utilisées pour estimer les distributions $\hat{Pr}[u|o]$.

La courbe de l'information perçue aura généralement l'allure de celle visible sur la figure 3.3. Elle contient deux informations essentielles :

1. Le niveau de convergence : celui-ci représente le niveau maximum d'information qu'on peut retirer des observations avec la méthode d'attaque qu'on utilise. Cette valeur permet donc de comparer les performances des différentes méthodes qu'on utilise et de tirer des conclusions en matière de location privacy.
2. La vitesse de convergence : celle-ci permet d'analyser l'influence du nombre d'observations utilisées sur la qualité du modèle. On peut donc se rendre compte du nombre d'observations dont on a besoin pour retirer un maximum d'information de la base de données.

C'est la présence de ces deux aspects qui nous a poussés à choisir cette métrique plutôt qu'une autre.

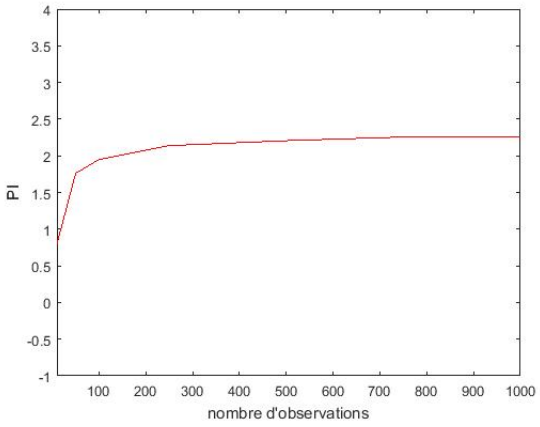


FIGURE 3.3 – Evolution de l'information perçue en fonction du nombre d'observations

Taux de succès

En parallèle avec l'information perçue, nous utiliserons le taux de succès (success rate ou SR) pour quantifier la location privacy. Celui-ci, comme son nom l'indique, va quantifier le succès d'une tâche. Dans le cadre de notre mémoire, la tâche est la suivante : identifier quel utilisateur a généré quelle observation. Le taux de succès représente donc dans ce cas précis le taux d'utilisateurs bien identifiés. Si nous voulons attribuer le bon utilisateur aux observations o d'un ensemble $\mathcal{O}$ de taille $|\mathcal{O}|$, le taux de succès est :

$$\begin{aligned} SR &= \frac{\#\ o \in \mathcal{O} \text{ attribuées au bon utilisateur}}{|\mathcal{O}|} \\ &= \frac{\sum_{o \in \mathcal{O}} Succes(o)}{|\mathcal{O}|} \end{aligned}$$

Avec $Succes(o)$ une fonction retournant 1 en cas de succès, c'est à dire si une observation o est "attribuée au bon utilisateur", et 0 sinon. En reprenant les notations définissant que o^{u_i} appartient à l'utilisateur u_i, cette fonction est définie comme suit :

$$\begin{cases} Succes(o^{u_i}) = 1 & \text{si } \hat{Pr}[u_i|o^{u_i}] > \hat{Pr}[u|o^{u_i}]\ \forall\ u \in \mathcal{U} \backslash \{u_i\} \\ Succes(o^{u_i}) = 0 & \text{sinon} \end{cases}$$

avec $\hat{Pr}[u|o]$ la distribution estimée par l'attaquant.

Dans le cadre de nos analyses, il est pertinent de pouvoir déterminer à partir de combien d'observations liées à un même utilisateur il est possible d'identifier celui-ci sur base des probabilités pour ces observations via le modèle estimé. Le taux de succès va nous permettre de le savoir. Nous utiliserons le même principe que celui défini ci-dessus à la différence que les observations ne serons plus testées indépendemment. Nous prendrons des groupements d'observations de taille g et nous essayerons de les attribuer au bon utilisateur. Puisqu'au plus g est grand, au plus nous avons d'information, il sera possible d'observer une corrélation entre la taille g du groupement et le taux de succès. Si nous avons un ensemble $\mathcal{O}^g = \{groupement_1, groupement_2, ...\}$ contenant des groupements de g observations, la formule devient

$$SR = \frac{\sum_{groupement \in \mathcal{O}^g} Succes(groupement)}{|\mathcal{O}^g|}$$

Sachant qu'un groupement est un ensemble de g observations, la fonction succès $Succes(groupement) = Succes(\{o_1, ..., o_g\})$ est définie comme suit :

$$\begin{cases} Succes(\{o_1^{u_i}, ..., o_g^{u_i}\}) = 1 & \text{si } \hat{Pr}[u_i|o_1^{u_i}] \cdot ... \cdot \hat{Pr}[u_i|o_g^{u_i}] > \hat{Pr}[u|o_1^{u_i}] \cdot ... \cdot \hat{Pr}[u|o_g^{u_i}] \ \forall \ u \in \mathcal{U} \backslash \{u_i\} \\ Succes(\{o_1^{u_i}, ..., o_g^{u_i}\}) = 0 & \text{sinon} \end{cases}$$

Sachant qu'il existe différentes manières de grouper g observations d'un ensemble (hormis si nous les groupons toutes ensemble ou une à une), nous faisons 10 permutations aléatoires de la matrice des observations et nous groupons les g premières observations ensembles, les g secondes ensembles, etc. Nous calculons alors les 10 taux de succès obtenus et le taux de succès final sera égal à la moyenne de ceux obtenus sur ces différents groupements. Nous obtenons ainsi une valeur plus précise et une courbe plus lisse pour le taux de succès.

Concrètement, le taux de succès s'exprime via trois dimensions :

1. Le nombre d'observations utilisées pour créer le modèle, celles dont on connait l'utilisateur qui les a générées
2. Le nombre d'observations de test, celles dont ne connait pas les utilisateurs qui les a générées et dont on tire des probabilités
3. Le taux de succès qu'on obtient sur les observations de la seconde dimensios via leurs probabilités a posteriori

Étant donné le temps et la mémoire que prendrait le calcul des courbes de taux de succès dans ces 3 dimensions, sans compter la difficulté de représentation de celui-ci sur papier, on propose de montrer les graphes pour seulement les deux derniers axes. On fixe donc le nombres d'observations servant à créer le modèle.
C'est ici qu'arrive également la meilleure justification de l'utilisation de l'information perçue. En effet, celle-ci représente la dépendance de la qualité du modèle, (traduite par l'information perçue) au nombre d'observations utilisées pour le créer.
Même si l'information perçue et le taux de succès ne s'expriment pas de la même manière, il existe des théorèmes montrant qu'une information perçue élevée entraine un taux de succès élevé [43]. En combinant les graphes de l'information perçue et du taux de succès selon les deux dimensions qu'on a choisies, on arrive à analyser l'influence des deux facteurs sur la qualité du modèle et en tirer les conclusions qui s'imposent quand à la location privacy.

Le taux de succès aura, dans nos analyses, une courbe généralement semblable à celle de la figure 3.4. Elle commence à la valeur du taux de succès obtenue avec une seule observation et converge vers la valeur maximale que le taux de succès peut atteindre.

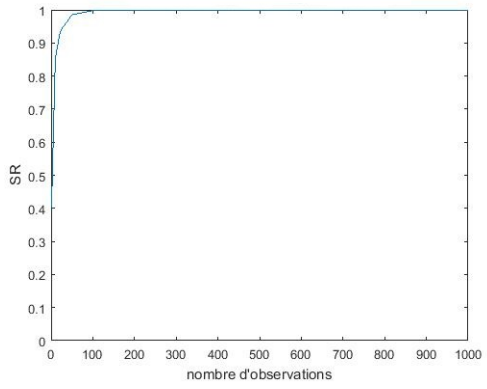


FIGURE 3.4 – Evolution du taux de succès en fonction du nombre d'observations

En conclusion, les graphes de taux de succès en 2D permettent d'une part de connaitre le meilleur taux de succès auquel on arrive pour un certain modèle, et d'autre part de savoir de combien d'observations de test on a besoin pour atteindre ce taux de succès.

3.2 Validation croisée

Comme nous l'avons vu au travers des métriques que nous venons de présenter, nos analyses se basent sur des estimations de modèles et de leur qualité. Le nombre de données disponibles pour estimer ces derniers est dans notre cas, souvent égal à 1000 par utilisateur. Une partie seulement de celles-ci sont utilisées pour la création en tant que telle du modèle. Cette partie des données est notée, comme dans les sections précédentes $\mathcal{O}_l$ et est de taille $n \cdot m_l$ avec n le nombre d'utilisateurs et m_l le nombre de données d'apprentissage par utilisateur. L'autre partie notée $\mathcal{O}_t$ sert à tester le modèle et est de taille $n \cdot m_t$. Les ensembles sont construits tels que $\mathcal{O}_l \cap \mathcal{O}_t = \emptyset$.

Étant donné le peu de données disponibles pour estimer le modèle et l'évaluer, nous utiliserons une validation croisée afin d'obtenir une approximation plus précise de celui-ci. Celle-ci consiste à séparer, pour chaque utilisateur ses m observations en k sous ensemble de taille $m_t = \frac{m}{k}$ comme on peut le voir sur la figure 3.5. Nous avons défini le nombre de sous ensembles égal à 10 arbitrairement dans nos analyses. Ce choix constitue le choix classique pour pour une validation croisée. On parle dans la littérature de "ten-fold validation croisée". Une fois ces groupes créés, nous prendrons les ensembles de test un à un et nous testerons le modèle créé sur base des $k-1$ autres sous-ensembles. Ainsi nous pourrons utiliser les mêmes données plusieurs fois en calculant des modèles sur base de configurations différentes des mêmes données. Si on a 1000 observations, on parvient donc à calculer $10 \cdot 100 = 1000$ probabilités issus de 10 modèles différents.

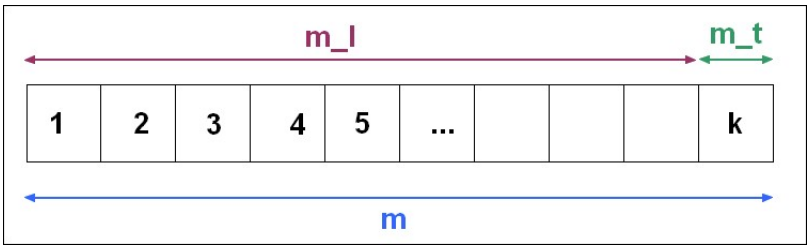


FIGURE 3.5 – Système de validation croisée pour les données d'un utilisateur

3.3 Méthodes d'attaque

Afin de calculer l'information perçue et le taux de succès il est, comme vu précédemment, nécessaire de disposer des probabilités $\hat{Pr}[u|o]$ associées aux observations.

L'usage classique est d'obtenir ces probabilités en estimant les fonctions de densité $\hat{p}(o|u)$ (resp. $\hat{f}(o|u)$ dans le cas continu). Plusieurs méthodes existent pour approximer ces densités. Dans cette section, nous présentons celles que nous avons choisies d'utiliser : le modèle discret en histogramme, l'estimation par noyau et l'algorithme de maximisation d'espérance supposant un modèle gaussien pour les distributions.

On propose aussi d'utiliser une autre approche pour obtenir non plus de véritables probabilités mais plutôt ce que nous appellerons des "pseudo-probabilités".
Cette approche utilise des classificateurs issus du machine learning. Dans leur utilisation premières, ceux-ci se basent sur un ensemble d'entrainement pour créer un modèle de classification

et pouvoir ensuite attribuer une classe aux observations de l'ensemble de test. On constate cependant que l'attribution d'une classe à une observation se base sur les probabilités a posteriori que l'observation appartienne à chaque classe [35]. Ces probabilités a posteriori sont fournies par MATLAB sous forme de scores dont la somme vaut bien 1, ce qui permet leur utilisation en tant que pseudo-probabilités. Nous insistons sur le fait que ces méthodes ne permettent pas d'obtenir des fonctions de densité mais reflètent bien le caractère prédictif de l'information perçue. Cependant, la définition formelle de cette information perçue requière l'utilisation de véritables probabilités, issues donc des distributions et de la formule de Bayes. Ainsi, par soucis de formalisme, on ne se permettra pas d'utiliser l'appellation "information perçue" pour la métrique découlant de ces scores mais plutôt "**Information perçue basée sur les scores**" (Score-Based Perceived Information) ou "**SBPI**".
Étant donné la similitude entre les courbes de PI et de SBPI, nous nous permettrons par la suite de comparer les valeurs de ces deux métriques comme si elles n'en faisaient qu'une. Nous ne démontrerons cependant pas leur "similarité".

Pareillement, pour le taux de succès, nous nous servirons de la même façon des probabilités que des pseudo-probabilités. Dans le cas de groupements d'observations, nous multiplierons donc les scores de celles-ci afin de deviner quel utilisateur les a générées.

Nous avons retenu la méthode des K plus proches voisins et des arbres de décision comme méthodes de classification pour obtenir ces scores.

Modèle discret

La première technique qu'on utilise est aussi la plus rudimentaire. Elle consiste à simplement discrétiser en bins le plan dans lequel se trouvent les utilisateurs. C'est à dire qu'une observation ne sera plus exprimée comme un point dans l'espace mais comme le bin contenant ce point. Il faut ensuite compter le nombre d'observations de chaque utilisateur qui se retrouve dans chaque bin et on peut alors en tirer une densité en histogramme afin d'obtenir les probabilités $\hat{p}(o|u)$. Par exemple, si un utilisateur u_i s'est retrouvé 10 fois dans un bin o_j sur ses m observations, $\hat{p}(o_j|u_i) = \frac{10}{m}$. Nous pouvons dès lors obtenir les probabilités $\hat{Pr}(u|o)$ par la règle de Bayes.

Afin de discrétiser le plan, nous choisissons un nombre de bins n_{bins} qui sera la discrétisation autant en longitude qu'en latitude. Dès lors nous obtenons un espace composé de n_{bins}^2 bins.

Modèle par Noyau

La deuxième technique que nous utiliserons est un outil très pratique : les "kernel density estimations" ou estimations par noyaux. Cette méthode nous permet, comme la précédente de ne faire aucune hypothèse sur les distributions des utilisateurs. Elle consiste à reconstituer la distribution en sommant une série de distributions appelées noyaux. Ces noyaux, par exemple gaussiens (c'est le choix que nous ferons dans ce document) sont centrés en chacune des observations dont on dispose. Ils ont de plus des paramètres à définir afin d'optimiser la qualité de l'estimation de la distribution qui en découle. Ces paramètres sont par exemple la covariance des noyaux dans le cas gaussien et sont constants pour chaque noyau.

Mathématiquement, dans le cas 2D, l'utilisation des kernels façonne une distribution : $\hat{f}(x, y)$ telle que [39]

$$\hat{f}(x,y) = \frac{1}{nh_xh_y} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h_x}\right) K\left(\frac{y - y_i}{h_y}\right)$$

avec n le nombre d'observations et h_x, h_y les "bandwidths" en x et en y. Cette formule se simplifie

en

$$\hat{f}(x,y) = \frac{1}{n}\sum_{i=1}^{n} K_{h_x}(x - x_i)\, K_{h_y}(y - y_i)$$

Cette dernière formule inclut la valeur de la bandwidth dans le kernel. Dans notre cas, avec les kernels gaussiens, la bandwidth correspond à l'écart type et on a donc

$$K_{h_x}(x - x_i) = \mathcal{N}(x_i, h_x^2) = \frac{1}{h_x\sqrt{2\pi}} \exp\left(\frac{(x - x_i)^2}{2h_x^2}\right)$$

Les estimations par noyaux sont en fait une généralisation du modèle discret. En effet, dans le modèle en histogramme, chaque observation pèse uniquement à l'endroit où elle se trouve (dans le bin où elle se trouve, précisément). Alors que dans le cas des kernels, chaque observation a une influence sur l'endroit où elle se trouve mais aussi sur son voisinage plus ou moins proche. De plus, les noyaux fonctionnent en espace continu puisqu'on n'utilise pas de bin mais bien la position exacte de chaque observation.

On comprend donc que la somme des noyaux (pondérée afin de garder une fonction de densité correcte) puisse représenter correctement la distribution des utilisateurs et nous permettre d'en tirer des probabilités.

Modèle par maximisation d'espérance

La dernière technique d'estimation des distributions que l'on utilise se base sur l'hypothèse que les distributions $\hat{p}(o|u)$ (resp. $\hat{f}(o|u)$) sont des gaussiennes ou des mixtures de gaussiennes.

En faisant cette hypothèse, nous pouvons utiliser un algorithme spécifique qui vise à estimer les paramètres des distributions gaussiennes. Les paramètres à estimer sont les moyennes, covariances et proportions. Celles-ci sont définies à l'aide de l'algorithme "Expectation maximization" (EM). Il nécessite cependant de connaitre le nombre de gaussiennes que l'on veut lier à chaque utilisateur.

En quelques mots, l'algorithme EM est un algorithme itératif comportant 2 types de phases répétées en alternance :

1. des phase où l'espérance de vraisemblance est évaluée en tenant compte des derniers paramètres calculés
2. des phases où l'on estime le maximum de vraisemblance des paramètres grâce à la vraisemblance de la phase 1

Cet algorithme ne sera pas expliqué plus en détails car nous avons utilisé la fonction *fitgmdist* implémentée dans MATLAB.

Méthode des k plus proches voisins

La première méthode de classification qu'on utilise est celle des K plus proches voisins. Cette méthode utilise la division des données en deux sous groupes (un ensemble de données d'apprentissage $\mathcal{O}_l$ et un ensemble de données de test $\mathcal{O}_t$). Ici les données d'apprentissage ne sont pas utilisées pour créer un modèle en tant que tel. Concrètement, les observations de cet ensemble sont simplement utilisés telles quelles, comme "références" pour les données de test. Précisons que dans notre cas, les classes correspondent aux utilisateurs et la classe correcte à prédire pour une observation est l'utilisateur qui l'a générée.

Afin de prédire la classe d'une observation de $\mathcal{O}_t$, le principe du classificateur "K-nearest neighbours" est de se baser sur la classe des K observations de $\mathcal{O}_l$ les plus proches de l'observation

en question. Par plus proches nous entendons plus proches selon la distance euclidienne dans un espace à deux dimensions. L'idée est que si une observation se trouve entourée de K observations appartenant à l'utilisateur u_i, il y a de grandes chances qu'elle appartienne à u_i également.

Comme on vient de le dire, l'objectif premier du KNN est de prédire une classe, mais la fonction MATLAB permet d'obtenir les pseudo-probabilités pour chaque classe (via un "score"). Ces pseudos-probabilités seront attribuées de la façon suivante : une observation aura une probabilité d'appartenir à la classe u proportionnelle au ratio d'observations de la classe u dans les K plus proches voisins. On utilise donc par la suite ces scores, ces pseudos-probabilités comme $\hat{Pr}(u|o)$.

Méthode des arbres de décision

On expérimente maintenant un autre classificateur qui ne nécessite pas de calculer la distribution $\hat{Pr}(o|u)$ des utilisateurs : les "decision trees" ou arbres de décision.
Ce processus consiste à créer un arbre pour caractériser les observations en fonction de critères basés sur la nature des observations (leurs "features" donc). Ainsi, la racine contient toutes les observations et chaque noeud contient celles qui satisfont les critères de tous les embranchements jusqu'à ce noeud. Plus on descend dans l'arbre, plus on fixe de critères sur les observations et plus on réduit donc leur nombre.

Si on laisse l'algorithme créer l'arbre qui colle au mieux aux observations, on risque d'avoir de l'overfitting, c'est à dire que le modèle sera très bon pour les observations sur lesquelles on crée le modèle ($\mathcal{O}_l$), mais il risquerait d'être mauvais pour les observations de test ($\mathcal{O}_t$). Afin d'éviter cela, on doit limiter la croissance de l'arbre en fonction d'un critère qu'on choisit. L'algorithme utilisé par MATLAB est l'algorithme *CART* [13] Celui-ci suit une série de règles pour optimiser une fonction. En particulier, l'algorithme choisit les critères de split de sorte que le nombre d'observations dans un nœud enfant ne soit pas plus petit qu'un certain paramètre : "MinLeafSize"(MLS). L'augmentation de ce paramètre permet de pouvoir faire une meilleure généralisation et d'avoir des résultats plus cohérents sur un ensemble de test (au détriment de la qualité de la classification de l'ensemble d'apprentissage). On peut également tailler l'arbre pour qu'il ne dépasse pas une certaine profondeur (niveau). Ce processus est appelé "pruning" et est effectué après avoir créé entièrement l'arbre. On peut ainsi à nouveau assurer une meilleure généralisation et diminuer l'overfitting.

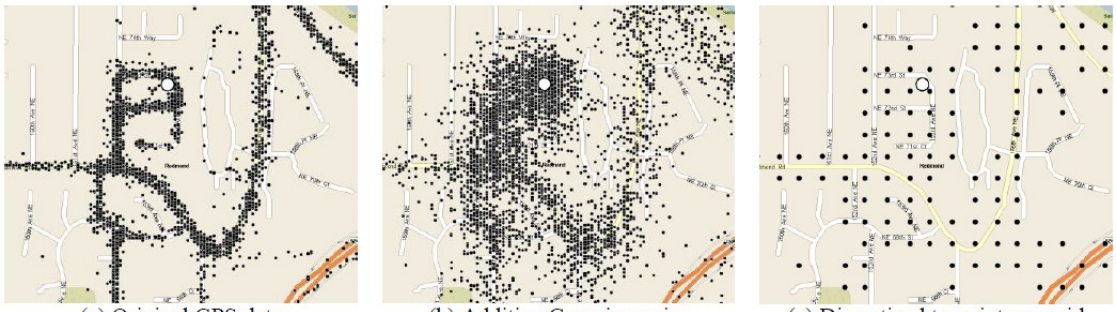
Dans notre cas, les conditions à chaque embranchement vont prendre la forme $latitude < k$, $latitude > k$, $longitude < k$,$longitude > k$. Plus on descend dans l'arbre et plus on aura de restrictions quant aux localisations des observations.

3.4 Méthodes de défense

Nous utiliserons deux moyens de défense pour tenter de réduire l'information perçue et le taux de succès obtenus dans l'analyse de location privacy d'une base de données.

Le premier, repris par Shokri et al. [41] sous le terme de "reducing precision" ou de "merging regions" est une méthode d'agrégation de l'espace. Celle-ci, comme son nom l'indique réduira la précision des données. Elles ne seront par exemple plus accessibles au kilomètre près mais seulement à 10km près. Dans notre analyse, nous diviserons l'espace en bins. Toutes les données se trouvant dans un même bin feront alors référence au même point sur la carte : le centre du bin. Avec ce principe les données contiendront a priori moins d'information et donc entraîneront une diminution d'information perçue et de taux de succès, en fonction du nombre de bins choisi.

Le second moyen de défense, cité par Krumm [30], est celui de la dégradation spatiale. Il est également repris par Shokri et al. [41] sous le nom de "perturbation/adding noise". En quelques mots cela consiste à dégrader l'exactitude des données, en ajoutant par exemple un bruit gaussien. Nos données seront alors dites bruitées et on peut analyser à quel point la dégradation des données doit être élevée pour obtenir le taux de succès maximum souhaité. Par exemple, Krumm [29] montre que pour des données GPS, même avec un bruit gaussien ayant une déviation standard de 1 km, on peut encore retrouver certains utilisateurs. Visuellement, ces moyens de défenses se présentent comme sur la figue 3.6.



(a) Original GPS data (b) Additive Gaussian noise (c) Discretized to points on grid

FIGURE 3.6 – Différents moyens de défense : ajouter du bruit ou discrétiser l'espace. [29]

3.5 Problème des outliers

Les données que nous utilisons pour calculer l'information perçue et le taux de succès contiennent des observations que l'on peut qualifier d'outliers. Nous entendons par là qu'elles se situent dans des endroits que l'utilisateur qui les a générées ne fréquente que très rarement et qui sont isolées de ses localisations habituelles. Le problème qui en découle est que suivant la méthode qu'on utilise pour calculer les probabilités ou pseudo-probabilités $\hat{Pr}[u|o = outlier]$ associées à ces outliers, celles-ci peuvent prendre une valeur nulle.
Suivant la méthode qu'on utilise, le taux de probabilités nulles varie mais globalement, celui-ci diminue lorsqu'on augmente le nombre d'observations et tend à devenir nul, comme on le voit sur la figure 3.7.

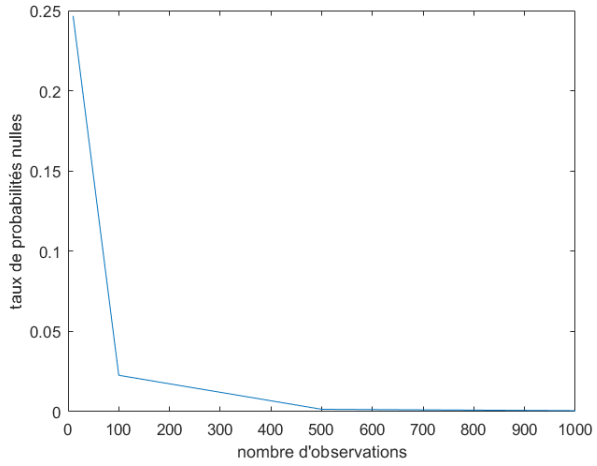


FIGURE 3.7 – Evolution du taux de probabilités nulles en fonction du nombre d'observations utilisées (modèle par KNN)

Il s'avère que ces valeurs nulles soulèvent un problème, tant au niveau de l'information perçue qu'au niveau du taux de succès. Nous présentons tout d'abord la nature du problème et y apportons ensuite notre solution.

- Problème dans l'information perçue :
  La définition de l'information perçue vue précédemment dans ce chapitre utilise un logarithme en base 2 de $\hat{Pr}[u|o]$. Si une seule de ces probabilités est nulle, son logarithme sera infiniment négatif et l'information perçue le sera également. Le problème est que ce résultat ne reflète pas correctement la qualité du modèle qu'on évalue. En effet, si le modèle permet d'expliquer toutes les autres observations, on veut que notre métrique soit cohérente et qu'elle soit donc élevée, même si elle doit logiquement être moins élevée que dans le cas où le modèle permet d'expliquer la totalité des observations.
- Problème dans le taux de succès :
  Ce problème se retrouve également dans le taux de succès mais de manière un peu différente. Prenons le cas où nous groupons les probabilités par le nombre g d'observations par utilisateur. Les probabilités liées aux g observations seront multipliées entre elles et la présence d'une probabilité nulle rendra ce produit nul également. De nouveau, si toutes les probabilités sont égales à 1 grâce à un modèle parfait pour toutes les autres observations, il sera tout de même impossible de remonter le produit à une valeur non nulle. Le taux de succès sera alors médiocre pour cause d'une exception, d'un outlier. Illustrons ceci avec la comparaison des probabilités liées à deux utilisateurs. Supposons un ensemble d'observations $\{o_1, o_2, o_3, o_4, o_5\}$. Les probabilités que ces observations appartiennent à l'utilisateur u_1 et u_2 sont données sur la table 3.1. Il semble assez évident, au vu des probabilités, que ces observations appartiennent à l'utilisateur u_1. Or, si nous acceptons les valeurs nulles telles qu'elles le sont, la multiplication des probabilités respectives à chaque utilisateur donnera deux valeurs nulles et on ne pourrait donc rien conclure quant à l'utilisateur qui a effectivement généré ces observations. La fonction *Succes* comme définie précédemment prendrait alors la valeur 0.

TABLE 3.1 – Probabilités que les observations o_i, $i \in [1, 5]$ appartiennent à l'utilisateur u_1 et u_2

	u_1	u_2
o_1	1	0
o_2	0	1
o_3	1	0
o_4	1	0
o_5	1	0

Solution proposée

Afin de réduire l'influence des probabilités nulles, nous utiliserons l'hypothèse que tout individu a une probabilité, très faible certes, mais non nulle de visiter tout endroit dans le monde. Autrement dit, chaque utilisateur a une probabilité non nulle d'être lié à chaque observation. Cette hypothèse nous semble d'autant plus plausible avec des données comme les nôtres, liées à des individus possédant un smartphone et utilisant une application sur celui-ci. En effet il est fort probable que cet ensemble d'utilisateurs ne contienne pas de personnes âgées ou de bébés qui, eux, ne se déplaceraient pas énormément. Notre hypothèse revient donc à remplacer les probabilités $\hat{Pr}[u|o]$ initialement nulles ou trop faibles par une probabilité faible mais non nulle .

Pour mettre en place cette hypothèse, il faut répondre à deux questions :

1. Quel est le seuil s à partir duquel on considère une probabilité comme trop faible ?
2. Par quelle valeur v remplacer une telle probabilité

Une solution serait évidemment de choisir $s = v$ ce qui simplifierait les choses. Ce n'est pourtant pas le choix que l'on fait. Choisir $v < s$ n'aurait pas de sens, on considère donc en toute généralite des choix respectant l'inégalité $v \geq s$.

Pour la première question, on choisit de ne considérer que les probabilités nulles comme trop faibles. Autrement dit, on accepte une probabilité inférieure à v pour peu qu'elle ne soit pas nulle. Ce choix est motivé par plusieurs remarques :

- Les pseudo-probabilités nulles obtenues par les méthodes de classification (KNN et arbres de décisions) et par le modèle en histogramme sont liées directement au nombre d'observations qu'on utilise pour créer le modèle. Étant donné qu'on ne dépasse pas 10000 observations pour nos analyses, la probabilité non nulle la plus faible qu'on puisse rencontrer est de 10^{-4}. Choisir $v < 10^{-4}$ empêche donc de trouver d'autres valeurs que 0 pour les probabilités inférieures à s.
- Les probabilités issues directement des distributions estimées (par noyau et par maximisation d'espérance) sont inversement proportionnelles à la distance entre les observations (à faible probabilité) et les zones à densité élevée. Par ailleurs, il semble cohérent que la probabilité pour un utilisateur de visiter un nouvel endroit soit dépendante de la distance entre cet endroit et son domicile ou son lieu de travail. On préfère donc garder la dépendance par rapport à ce lieu que de fixer une probabilité arbitraire pour les observations à faible probabilité. On considère donc que la dépendance ne perd son sens que pour les lieux vraiment trop éloignés, c'est à dire précisément quand la probabilité est nulle (ou en tout cas quand MATLAB la considère comme telle). Ces lieux sont donc considérés comme ayant simplement une probabilité d'être visitée constante et non dépendante du lieu de vie.

Précisons tout de même que ce changement devrait normalement impacter toutes les probabilités. En effet, si nous n'acceptons pas de probabilité nulle, il ne peut pas y avoir de probabilité égale à 1 non plus par exemple. Cependant, dans notre cas, nous ne changerons que les probabilités nulles étant donné que ce sont les seules qui ont un impact significatif sur l'information perçue et le taux de succès. En effet, pour la PI :

$$\log(0) \ll \log(\epsilon)$$

pour ϵ petit mais

$$log(1 - \epsilon) \simeq \log(1)$$

et pour le SR :

$$a \cdot 0 = b \cdot 0$$

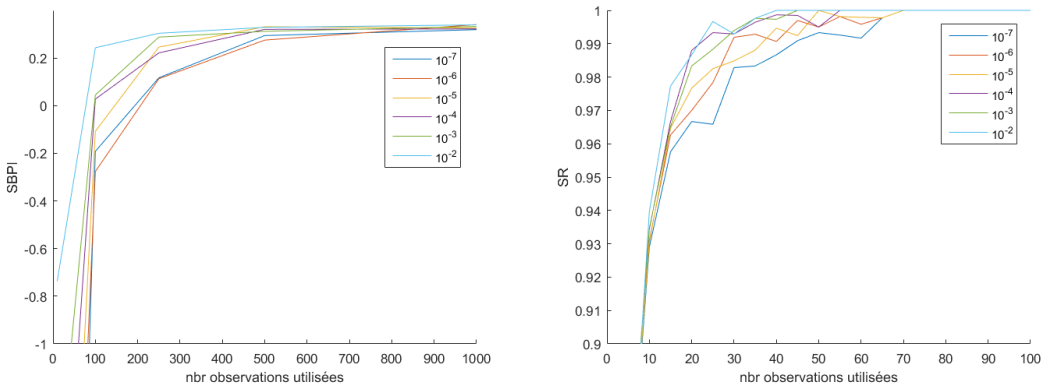
et ce même pour $a \ll b$ mais

$$\max(a \cdot 1, b \cdot 1) = \max(a \cdot (1 - \epsilon), b \cdot (1 - \epsilon))$$

pour a et b quelconques.

En pratique, la valeur de v qu'on choisit est utilisée de la manière suivante : on remplace par v toutes les probabilités $\hat{Pr}[u_i|o_j^{u_k}]\ \forall i, j, k : \hat{Pr}[u_i|o_j^{u_k}] = 0$. Par contre, on ne se sert pas de toutes ces probabilités dans le calcul de la PI. En effet, dans cette dernière, on utilise seulement les probabilités $\hat{Pr}[u_i|o_j^{u_k}]\ \forall i, j, k : i = k$. En français, la formule de l'information perçue ne conserve

que les probabilités associées à l'utilisateur qui a généré chaque observation.
Pour la seconde question, il faut analyser l'impact de la valeur de v sur l'information perçue et le taux de succès.



(a) SBPI (b) SR

FIGURE 3.8 – Influence de la valeur v par laquelle on remplace les probabilités nulles sur la SBPI et le SR (modèle par KNN obtenu avec 1000 observations)

Pour la SBPI on remarque que la valeur de v a surtout une influence pour des nombres faibles d'observations étant donné que c'est pour ces valeurs qu'on a le plus de probabilités nulles. Celles-ci viennent de la mauvaise qualité du modèle pour des nombres faibles d'observations. On vérifie sans surprise qu'au plus la valeur de v est élevée au plus la valeur de l'information l'est également étant donné que la valeur de v intervient explicitement dans la formule de la PI à la place des probabilités nulles. (Dans le cas de la figure 3.8a, le fait que les courbes de SBPI soient confondues après 500 observations vient du fait qu'avec au moins autant d'observations, le modèle est suffisamment performant pour éviter les probabilités nulles. Si ce taux est nul, la PI sera exacte et elle représentera parfaitement le modèle alors que s'il reste des probabilités nulles, on acceptera que notre modèle n'explique pas la totalité des observations et soit donc imparfait, pour les outliers précisément. Pour ces données on triche donc légèrement sur la formule pour masquer l'imperfection du modèle et conserver une caractérisation plus réaliste de celui-ci. On gardera tout de même a l'esprit que la subjectivité de la valeur de v et son absence de lien avec le modèle diminue l'interprétabilité de la valeur de la PI.

Pour le taux de succès, l'influence de v est moins nette. Même si globalement plus sa valeur est élevée, meilleur sera le taux de succès, il y a quand même des entrelacements entre les courbes. Ceux-ci s'expliquent par le fait que même si remplacer une probabilité $\hat{Pr}[u_i|o_j^{u_k}]$ nulle par v est favorable au taux de succès lorsque $i = k$, c'est l'inverse pour les cas où $i \neq k$. Suivant le nombre de fois où chacun des cas apparaissent, l'une ou l'autre valeur de v peut donc être préférable. On se rend donc compte qu'il n'y a aucune "triche" dans le calcul du SR. On applique le même traitement à toutes les probabilités et pas seulement à certaines d'entre elles. On choisit simplement d'utiliser différemment nos probabilités de sorte à maximiser le taux de succès global des méthodes même si dans certains cas, modifier les probabilités nulles peut légèrement diminuer le taux de succès de l'une ou l'autre méthode.
Malgré cela, les différences entre les différentes courbes sont minimes. Sur la figure 5.7, la plus grande ne dépasse pas quatre centièmes et celle-ci tend à devenir nulle quand on multiplie assez d'observations.

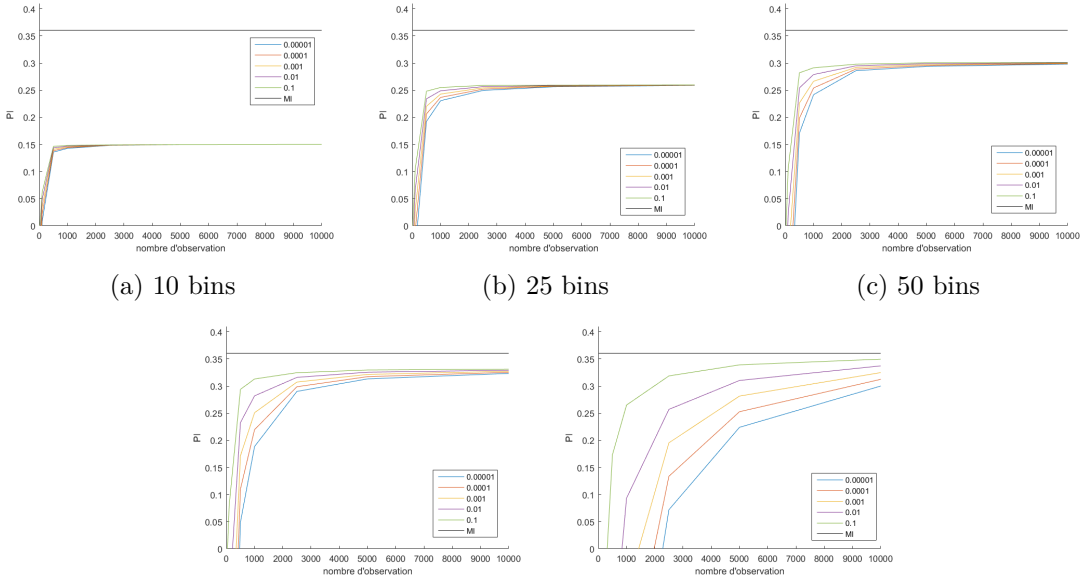
Finalement on choisit d'utiliser une valeur de v égale à 10^{-5}. Cette valeur fournit des résultats corrects pour le taux de succès comme sur l'information perçue. De plus, cette valeur permet

d'éviter d'avoir à remplacer des valeurs inférieures mais non nulles par v dans le cas des modèles par histogramme, KNN et arbres de décision, comme on l'a vu au début de la section.

On peut se rendre compte que cette valeur a du sens avec le raisonnement suivant : sur 10 ans on peut imaginer que notre quotidien se résume à deux endroits "maison" et "boulot" en prenant une surface de ville moyenne cela correspond à 25 km^2 visités quotidiennement. On peut se dire que sur un an on part facilement 10 fois à un nouvel endroit, pour un week-end, des vacances ou le boulot en donnant à ces endroits la même superficie que notre quotidien, soit 25km^2.
La terre a une superficie émergée de 149400000 km^2. Ce qui fait presque 6 millions d'endroits visitables (que l'on suppose équiprobable dans ce cas). Si on part une fois, on a une probabilité de 1.7 10^{-7} d'aller à chaque endroit. En le faisant 10 fois sur l'année pendant 10 ans donc 100 fois, cela donne 1.7 10^{-5} comme probabilité de visiter chaque endroit.

Le cas particulier de l'histogramme

Le cas de l'histogramme est un peu plus délicat que les autres modèles générant des pseudo-probabilités nulles. En effet, avec celui-ci, augmenter le nombre de bins entraine une augmentation du taux d'apparition des probabilités nulles. Si celles-ci surviennent à une fréquence non négligeable, la valeur de v choisie change significativement les résultats et nous ne pouvons alors plus réellement nous y fier.
Les graphes de la figure 3.9 nous permettent de nous rendre compte de l'impact de la valeur de v sur la PI en fonction du nombre de bins utilisés.



(a) 10 bins (b) 25 bins (c) 50 bins

(d) 100 bins (e) 500 bins

FIGURE 3.9 – Comparaison des PI avec différentes valeurs de v pour différents nombres de bins

Comme attendu, on voit qu'augmenter le nombre de bins augmente la qualité du modèle et donc le niveau de convergence général de la PI.
Pour 10 bins, les courbes pour différentes valeurs de v convergent vers des valeurs de PI extrêmement proches avec seulement 1000 observations. On peut donc dans ce cas là conclure qu'on a une PI quasi exacte. En effet, pour un nombre très faible de bins, le risque d'avoir une probabilité nulle devient vite très faible quand on augmente le nombre d'observations du modèle, et le peu de fois où cela arrive n'est pas suffisant pour changer considérablement la PI.
Si on divise l'espace en plus de bins, la convergence des courbes avec différents v nécessite de plus en plus d'observations. Ainsi, avec 50 bins, il faut 5000 observations pour que les courbes convergent et qu'on puisse se fier à la valeur de la PI (un peu inférieure à 0.3). Avec 100 et 500

bins, 10000 observations ne suffisent pas à faire converger les courbes vers une même valeur. On ne peut donc pas vraiment se fier aux valeurs de PI obtenues étant donné qu'on n'a pas de moyen de savoir quelle valeur de p donne le niveau de PI le plus acceptable.

Globalement, on comprend que pour avoir une PI suffisamment précise avec un modèle en histogramme, il faut se restreindre à utiliser peu de bins, et donc viser un niveau de PI assez faible.

Chapitre 4

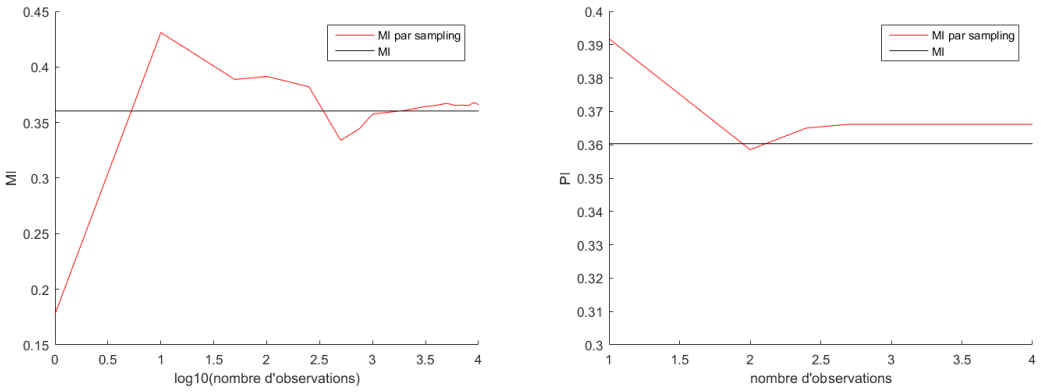
Analyse des données simulées

Ce chapitre analyse l'influence des méthodes d'attaque utilisées sur l'information perçue et le taux de succès. Afin de savoir si elles sont efficaces, nous calculerons dans un premier temps l'information mutuelle puisque celle-ci peut être ici calculée. Ainsi nous saurons si toute l'information disponible est perçue par les méthodes d'attaque. Nous affûterons alors ces méthodes afin de les rendre les plus performantes possibles individuellement et surtout de les comparer entre elles. Nous pourrons alors utiliser les plus performantes parmi celles envisageables en pratique sur de plus grandes bases de données (la contrainte principale étant celle du temps de calcul). Nous pourrons alors nous servir de celles-ci pour l'analyse sur les données réelles.

4.1 Information mutuelle

On peut calculer la MI par sampling de plusieurs façons différentes. Sur la figure 4.1a, on la calcule comme la moyenne de l'information perçue sur 10 ensembles de 10000 observations. On voit qu'elle oscille autour de la MI réelle et que les oscillations s'amenuisent en augmentant le nombre d'observations.

Sur la figure 4.1b par contre, on ne considère que les 10000 observations qu'on utilise par la suite. On se met donc dans la situation où notre base de donnée comporte seulement ces observations. Afin de lisser la courbe de l'information mutuelle par sampling, on utilise la même technique que pour l'information perçue, c'est à dire que pour chaque nombre n d'observations pour lesquels on calcule la MI, on fait une moyenne sur les $\frac{10000}{n}$ ensembles de n observations. On voit dans notre cas que les 10000 observations sont légèrement biaisées de sorte que la MI par sampling est légèrement supérieure à la MI réelle (quelque six millièmes seulement). On peut donc s'attendre à ce que l'information perçue puisse dépasser l'information mutuelle, mais cependant pas l'information mutuelle par sampling.



(a) Avec 10 ensembles d'observations (b) Avec les observations utilisées

FIGURE 4.1 – Convergence de la MI par sampling vers la MI réelle

4.2 Optimisation des méthodes d'attaque

Maintenant que nous savons la quantité maximale à retirer des données, nous pouvons optimiser les différentes techniques d'attaque.

4.2.1 Modèle discret

Pour ce modèle nous allons discrétiser l'espace en un nombre de bins défini. La première question qui apparait est celle du choix de ce nombre de bins. Celui-ci est en effet capital pour l'estimation de la PI étant donné que l'augmentation du nombre de bins améliore la qualité du modèle. Malheureusement, pour la raison évoquée à la fin du chapitre précédent, on est contraint de se limiter à un nombre de bins relativement faible, afin de garder des résultats cohérents pour l'information perçue.
Nous constatons de plus que le nombre de bins augmente légèrement le temps pris par l'algorithme comme on le voit sur la figure 4.2. Le temps n'est cependant par rédhibitoire pour cette méthode étant donné que celui-ci reste très court même avec beaucoup de bins et beaucoup d'observations. Il n'intervient donc pas dans le choix du nombre de bins utilisés et constitue au contraire un point fort de cette méthode.

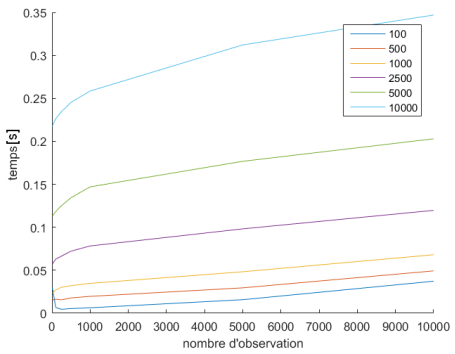


FIGURE 4.2 – Comparaison des temps de calcul de la PI avec modèle en histogramme pour différents nombres de bins

En prenant les deux remarques précédentes en considération, on propose d'utiliser 50 bins, ce qui donne un niveau de 3 et permet la comparaison avec les autres modèles.

4.2.2 Modèle par Noyau

On utilise maintenant l'estimation par noyau (kernel density estimation) pour approximer les distributions. Hormis le choix du noyau que l'on ne discute pas ici et qu'on choisit gaussien, ce qui constitue le choix le plus populaire, le paramètre principal pour cette méthode est la covariance que l'on donne à chaque noyau , aussi appelée "bandwidth". On constatera dans la suite que celle-ci a une grande importance, puisque différents choix pour celle-ci mènent à des PI fort différentes.

3 choix sont possibles pour le type de covariance utilisée :

- $\begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}$: identité multipliée par un scalaire, ne tient pas compte des différences de dispersion entre les deux coordonnées (cfr figure 4.3(1))
- $\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$: matrice diagonale, tient compte des différences de dispersion mais pas de l'orientation des distributions (cfr figure 4.3(2))
- $\begin{bmatrix} a & c \\ c & b \end{bmatrix}$: matrice symétrique semi-définie positive, permet des lissages dans toutes les directions (cfr figure 4.3(3))

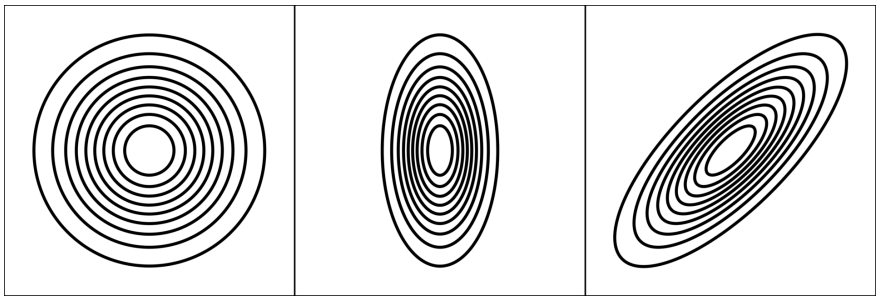
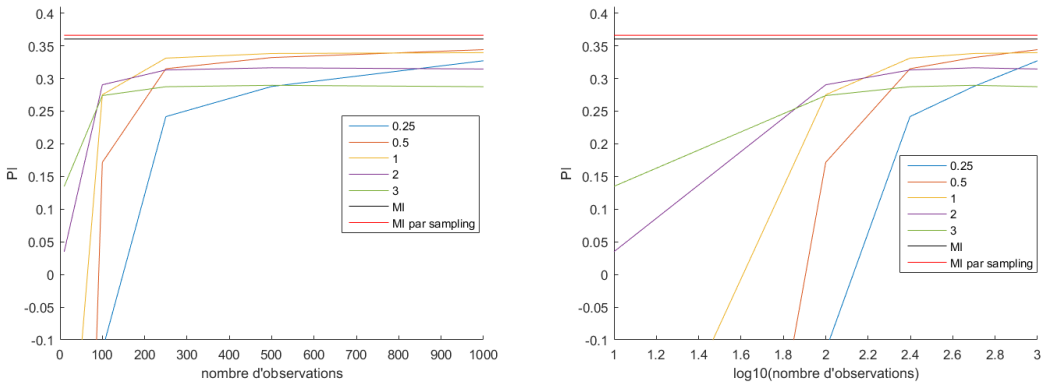


FIGURE 4.3 – Influence de la covariance sur les noyaux gaussiens ([3])

Les covariances les plus utilisées sont les deux premiers types, pour des raisons de simplicité de calcul. Cependant, des recherches montrent que l'utilisation de covariances non diagonales peut améliorer la qualité de l'estimation des densités [46],[21].
Pour des raisons de complexité temporelle, on se limite à 1000 observations pour l'utilisation des kernels. En effet, pour évaluer la valeur de la densité en un point (en une observation donc), il est nécessaire de recalculer la somme des contributions de chaque observation ayant servi à créer le modèle et donc d'évaluer chaque gaussienne au point correspondant à l'observation.

Covariance multiple de l'identité

On teste d'abord des covariances constantes, multiples de l'identité, on choisit arbitrairement les coefficients 0.25, 0.5, 1, 2 et 3.



(a) PI(échelle linéaire) (b) PI(échelle logarithmique)

FIGURE 4.4 – Evolution de la PI avec le modèle construit par kernel pour des covariances multiples de l'identité

On remarque sur la figure 4.4 que pour des valeurs supérieures à 1, la PI a déjà convergé avec 1000 observations et celle-ci n'atteint pas la valeur de la MI, on n'a donc pas un modèle parfait. Par contre avec ces valeurs, la PI atteint assez vite un niveau positif et relativement acceptable. A contrario, des valeurs inférieures à 1 fournissent une PI qui n'a pas encore convergé avec 1000 observations. La valeur de la PI avec 0.25 est inférieure à celle avec 1 mais est encore en nette augmentation. De la même façon, la courbe avec 0.5 continue d'augmenter mais moins vite qu'avec 0.25 et présente une valeur similaire à celle avec $a = 1$ pour 1000 observations. D'autre part, la convergence avec ces petites valeurs prend plus de temps qu'avec des valeurs supérieures à 1.
On pourrait donc espérer qu'en augmentant le nombre d'observations, les petites valeurs fassent converger la PI vers la valeur de la MI mais ce n'est pas garanti et ce pourrait très bien ne pas être le cas. A nouveau, on ne testera pas cette hypothèse ici étant donné le temps pris par l'utilisation des kernels.
Puisque la meilleure valeur dépend du nombre d'observations, on comprend qu'on ne peut pas obtenir de résultats optimaux avec des covariances constantes.

Covariance diagonale obtenue par la règle de Scott

Un meilleur choix nous est donné par la "Silverman's rule of thumb" dans le cas 1D [17], [26] :

$$\hat{h}_{opt} = 0.9 \min\left(\hat{\sigma}; \frac{\hat{q}_3 - \hat{q}_1}{1.349}\right) n^{-\frac{1}{5}}$$

$\hat{h}_{opt}$ correspond à l'écart type du kernel gaussien, $\hat{\sigma}$ à la variance des observations pour l'utilisateur, $\hat{q}_3$ et $\hat{q}_1$ aux troisième et premier quartiles calculés pour les observations de l'utilisateur et n au nombre d'observations pour l'utilisateur.
Dans le cas multidimensionnel, cette règle est adaptée par Scott (1992) [26] :

$$\hat{h}_j = n^{-\frac{1}{d+4}} \hat{\sigma}_j$$

avec $\hat{h}_j$ l'écart type du kernel gaussien pour la composante j, d le nombre de dimensions et $\hat{\sigma}_j$ l'écart type de la coordonnée j.
D'une part, cette formule prend en considération le nombre d'observations dans le modèle. D'autre part, elle permet de choisir des valeurs différentes pour la variance de chacune des deux coordonnées, ce qui est plus raisonnable dans le cas où la distribution est plus allongée dans une direction que dans une autre. Nous avons testé cette méthode et calculé l'information perçue obtenue avec celle-ci afin de pouvoir la comparer avec les résultats précédents.

Covariance pleine (extension de Scott)

Malgré qu'il n'y ait pas moyen de dériver une règle similaire pour une matrice de covariance pleine, Härdle et al. [26] proposent de se baser sur les observations précédentes pour choisir d'utiliser une covariance qui dépend de la même façon du nombre d'observations et qui est multiple de la covariance des données par un exposant un demi :

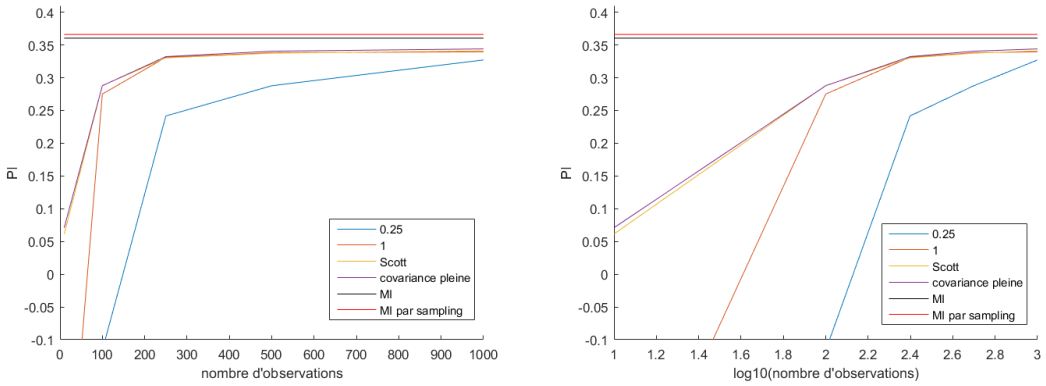
$$\hat{H} = n^{-\frac{1}{d+4}} \hat{\Sigma}^{\frac{1}{2}}$$

De nouveau, nous avons calculé l'information perçue obtenue à l'aide de cette règle afin de comparer les résultats.

Comparaison des résultats pour le kernel

On compare ici les quatre techniques les plus prometteuses qu'on ait testées jusqu'à maintenant (cfr figure 4.5) :

- covariance valant un quart de l'identité
- covariance identité
- covariance diagonale obtenue par la formule de Scott
- covariance pleine (Scott)



(a) PI (échelle linéaire) (b) PI (échelle logarithmique)

FIGURE 4.5 – Comparaison des PI avec kernels et différents types de bandwidth

On s'aperçoit qu'inclure des termes non diagonaux dans la covariance n'a qu'un intérêt minime et ce seulement pour un nombre faible d'observations. Les deux courbes avec les formules de Scott diagonale et non diagonale sont quasiment confondues alors que la covariance des données est bien non diagonale.
On voit aussi que ces deux courbes sont très proches de celle pour une covariance identité à partir de 100 observations et que les calculs de la formule de Scott n'apportent au final pas une grande amélioration par rapport au choix "canonique" à part pour des petits nombres d'observations.

A nouveau, nous avons également observé le temps de calcul de cette méthode afin de comparer la complexité temporelle du Kernel avec celle des autres méthodes. On peut analyser cet aspect à l'aide de la figure 4.6.

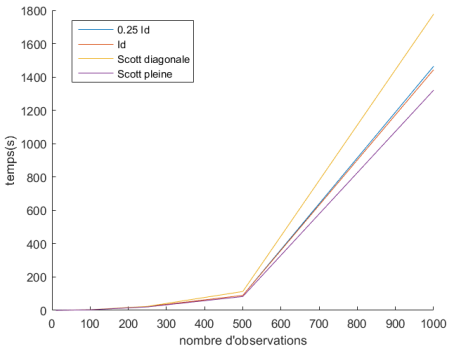


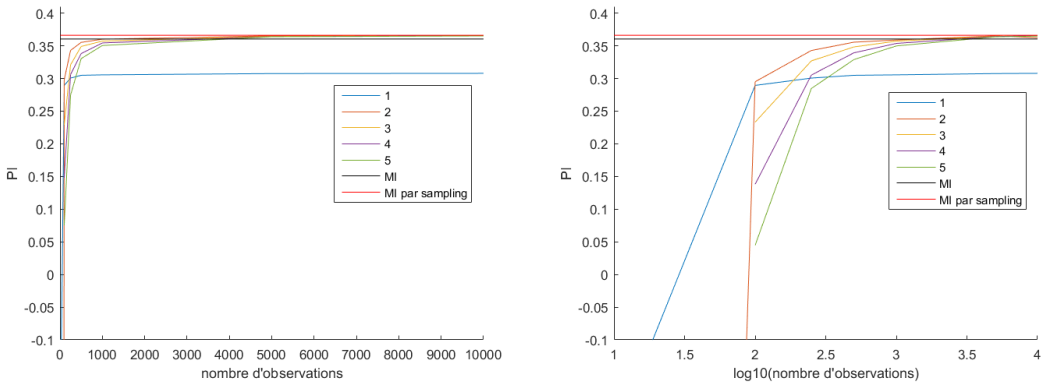
FIGURE 4.6 – Durée de calcul de la PI avec kernel

Les résultats temporels sont relativement surprenants. Alors que la PI est très similaire pour la règle de Scott avec une matrice diagonale ou pleine, le temps pris par l'algorithme est plus grand avec une matrice diagonale. La différence est assez significative puisque pour 10000 observations, le temps augmente de 1300 à 1800 secondes. Avec une covariance multiple de l'identité, le temps est intermédiaire et d'environ 1400 secondes.
Le temps augmente exponentiellement et on comprend qu'on peut difficilement tester le modèle par noyau pour des valeurs plus grandes que 1000 observations.
On réalise également que pour plus d'utilisateurs, le temps d'exécution de l'algorithme va considérablement augmenter et que cette méthode risque d'être inutilisable en pratique dû à son temps d'exécution exorbitant.

4.2.3 Modèle par maximisation d'espérance

Comme expliqué dans les outils d'analyse, ce modèle se base sur l'hypothèse que les distributions sont des gaussiennes ou des mixtures de gaussiennes, afin de pouvoir utiliser l'algorithme "Expectation maximization" (EM).

L'utilisation de cet algorithme nécessite de lui fournir en argument le nombre q de gaussiennes a priori dont il faut optimiser les paramètres, c'est à dire le nombre de gaussiennes supposées par utilisateur. Nous ferons l'hypothèse que celui-ci est égal pour chaque utilisateur.
Il faut bien réaliser que dans la plupart des cas, on ne connait pas la valeur q^* du nombre de gaussiennes dont sont issues les observations. En toute généralité, les distributions ne sont même pas forcément gaussiennes et il n'y a donc pas moyen de connaitre la valeur optimale de q. Il est donc important de savoir l'impact qu'aura un mauvais choix de ce nombre sur la PI. On la calcule pour des valeurs de q allant de 1 à 5.



(a) PI (échelle linéaire)

(b) PI(échelle logarithmique)

FIGURE 4.7 – Évolution de la PI avec le modèle construit par EM pour différentes valeurs de q

Le premier résultat qui saute aux yeux sur la figure 4.7 est que si on passe en argument un nombre q de gaussiennes au moins égal au nombre correct q^* (2,3,4 ou 5 donc), la valeur de la PI converge vers la MI. Par contre, avec un nombre trop petit (1), la PI converge vers un niveau plus bas que la MI.
En regardant les valeurs exactes, on constate que plus on augmente q par rapport à q^*, moins la PI est élevée, même si la différence n'est que de l'ordre du dix-millième. En particulier, pour le bon nombre de gaussiennes a priori, la différence entre la MI par sampling et la PI avec 1000 observations n'est que de six millièmes seulement. Avec 10000, la PI dépasse même la MI réelle mais reste bornée par la MI par sampling. Ensuite, on constate logiquement que plus on suppose de gaussiennes, plus il faut de données pour en calculer les paramètres. Ainsi, avec peu d'observations, la courbe avec 1 gaussienne donne une meilleure valeur de PI qu'avec plus de gaussiennes. De la même façon, même si la valeur de convergence est quasi similaire avec des nombres q suffisants, la vitesse de convergence est supérieure avec un minimum de gaussiennes a priori. On conclut donc que supposer un trop grand nombre de gaussiennes pour s'assurer d'une valeur de PI élevée a un impact sur la vitesse de convergence de cette PI dans le cas où on surestime ce nombre. Il faut donc bien choisir ce paramètre.

On peut expliquer la convergence de l'algorithme pour différentes valeurs de q en regardant les valeurs prises par les différentes moyennes, covariances et proportions. On constate que l'algorithme peut soit fixer la proportion d'une gaussienne à une valeur très faible pour en diminuer l'impact et se ramener au bon nombre de gaussiennes, soit définir deux gaussiennes très similaires et donc avoir le nombre de gaussiennes réellement différentes souhaité tout en gardant une vraisemblance optimale.

Ce modèle est donc très prometteur, au moins dans le cas où on connait l'aspect général des distributions.

Nous analysons ensuite le temps de calcul de l'algorithme à l'aide de la figure 4.8.

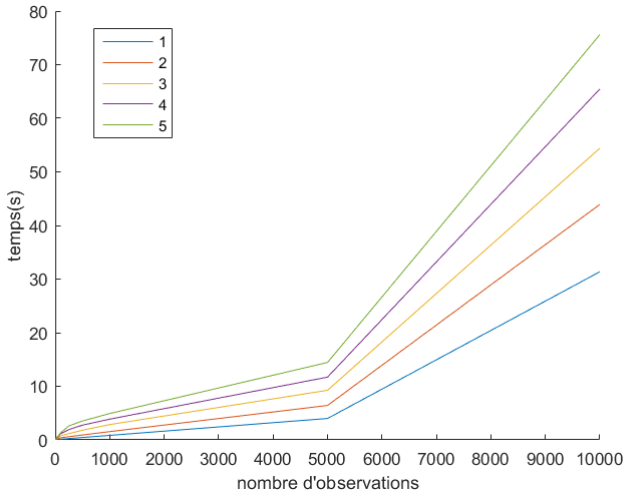


FIGURE 4.8 – Durée de calcul de la PI avec EM pour différents nombres de gaussiennes a priori

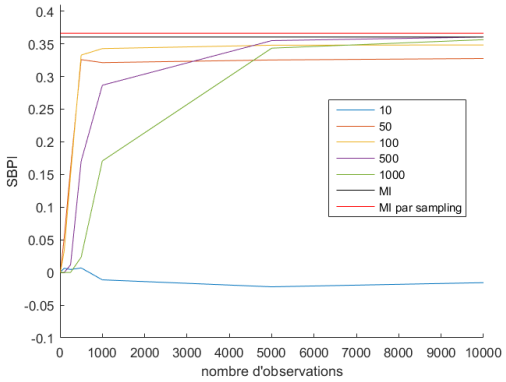
Sans surprise, le temps pris pour évaluer l'information perçue augmente avec le nombre de gaussiennes à ajuster. L'incrément induit par l'ajout d'une gaussienne est à peu près constant. On voit également que le temps augmente exponentiellement avec le nombre d'observations. Cependant il reste raisonnable et on ne dépasse pas 80 secondes même avec 10000 observations et 5 gaussiennes. L'algorithme est donc assez rapide et on peut donc envisager de l'utiliser pour des bases de données contenant plus d'utilisateurs

4.2.4 Méthode des K plus proches voisins

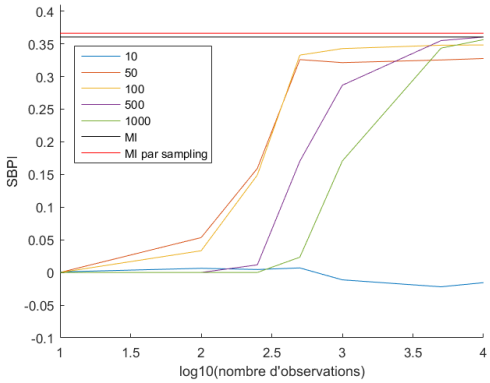
La première méthode de classification qu'on utilise est celle des K plus proches voisins (K-nearest neighbours). Le paramètre principal de cette méthode est la valeur de K correspondant au nombre des plus proches voisins pris en compte. Il est alors intéressant de discuter du choix de la valeur de ce paramètre. Dans cette section, on essayera d'abord d'utiliser différentes valeurs constantes avant d'utiliser la validation croisée pour choisir la meilleure valeur pour chaque taille de sous ensemble sur laquelle on calcule la SBPI.

K constant

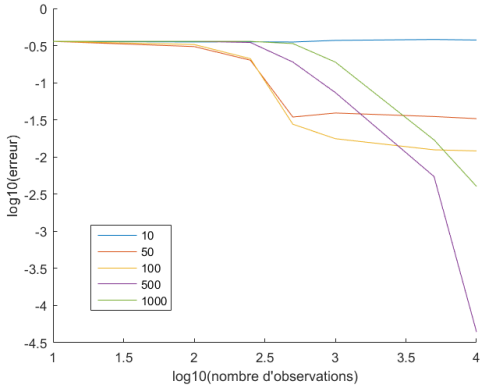
On essaye d'abord des valeurs de K constantes pour tout nombre d'observations. On compare les SBPI obtenues pour K prenant les valeurs 10, 50, 100, 500 et 1000.



(a) SBPI (échelle linéaire)



(b) SBPI (échelle logarithmique)



(c) Erreur logarithmique

FIGURE 4.9 – Evolution de la SBPI et de l'erreur de SBPI avec KNN et différentes valeur de K

On remarque d'abord sur les figures 4.9 que parmi les valeurs qu'on a testées, seule 10 donne de très mauvais résultats, il s'agit donc juste d'une valeur trop faible.
Ensuite, avec 50, les résultats sont déjà bien meilleurs : la SBPI converge mais vers une valeur un peu en dessous de celle de la MI, on a une erreur de trois centièmes.
Avec 100, la convergence se fait vers un niveau légèrement meilleur et on a une erreur d'à peine plus d'un centième.
En augmentant encore K à 500, la SBPI augmente encore. On pourrait d'ailleurs croire qu'elle converge vers la MI étant donné que sur la figure 4.9c, l'erreur ne fait que diminuer. Il se trouve cependant qu'elle dépasse même la valeur correcte. La différence avec la MI par sampling est quant à elle de quelques millièmes seulement.
Les dernières constatations pourraient nous mener à croire qu'augmenter la valeur de K améliore encore la valeur de la SBPI mais il n'en est rien. En effet, avec 1000 observations, la SBPI est un peu moins élevée qu'avec seulement 500.
Globalement, on remarque qu'une fois de plus la valeur optimale de K n'est pas la même pour tous les nombres d'observations. En effet, plus K est élevé et plus la SBPI prend du temps à converger. Ceci nous amène à penser qu'une validation croisée sur ce paramètre pourrait s'avérer être intéressante.

K crossvalidé

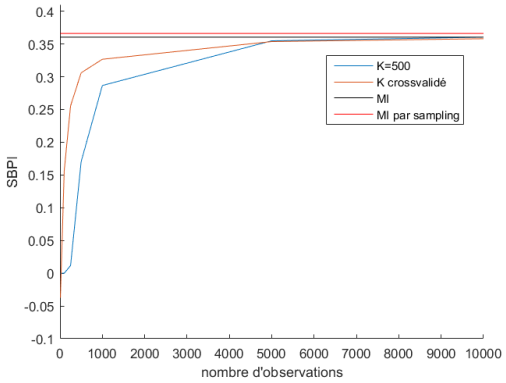
Pour la validation croisée, on partitionne les observations d'entrainement $\mathcal{O}_l$ en 5 "folds". On crée alternativement le modèle sur 4 de ces 5 ensembles et on évalue l'erreur quadratique moyenne (mse) sur le $5^{ème}$ ensemble pour 20 valeurs de K espacées logarithmiquement entre 3 et la taille de l'échantillon utilisé pour la création du modèle. On utilise ensuite la valeur de K

donnant la meilleure erreur pour calculer l'information perçue sur les observations de test ($\mathcal{O}_t$).

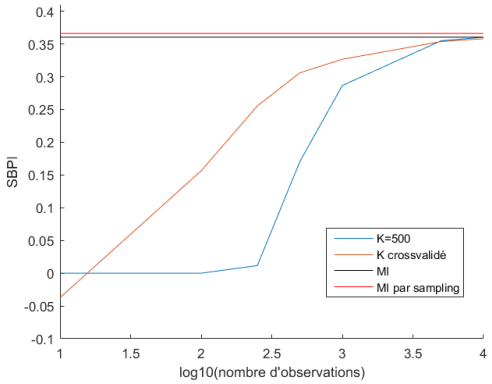
Comparaison des résultats pour le KNN

On compare à nouveau les différentes valeurs de K intéressantes pour le KNN :

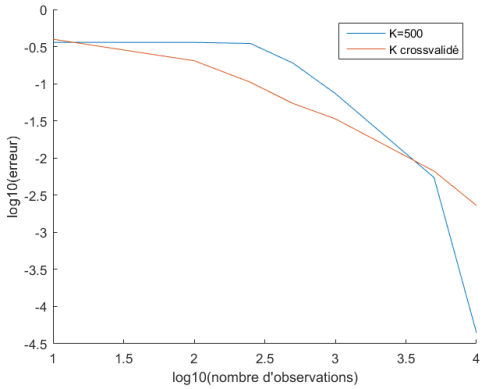
- $K = 500$
- K crosvalidé



(a) SBPI (échelle linéaire)



(b) SBPI (échelle logarithmique)



(c) Erreur logarithmique

FIGURE 4.10 – Comparaison des erreurs de SBPI et erreurs de SBPI avec KNN pour différents paramètres K

On confirme bien avec la figure 4.10 que la validation croisée apporte quelque chose par rapport à un choix constant de K. Avec au moins 5000 observations, la différence entre les 2 courbes est tellement mince que l'intérêt ne semble pas énorme. Par contre, pour moins d'observations, la SBPI est bien meilleure avec le K choisi par validation croisée qu'avec le K fixé très grand. Par exemple, avec 500 observations, l'erreur est de six centièmes seulement alors qu'avec $K = 500$, elle était d'environ deux dixièmes.

On est étonné de voir qu'une valeur de 500 pour K donne une aussi bonne valeur de MI, et surtout que la validation croisée ne permette pas d'atteindre cette valeur. On retrouve pourtant dans les valeurs testées une valeur proche de 500 qui pourrait être choisie comme meilleure valeur et donner une SBPI plus élevée. En testant une validation croisée avec K entre 450 et 550, on n'arrive pourtant pas à obtenir une SBPI aussi élevée (même si la différence n'est que de deux dix-millièmes). L'explication qu'on donne à cela est que dans la validation croisée, la sélection du K optimal se fait sur quatre cinquièmes des observations servant à créer le modèle ($\frac{4}{5}m_l$). Par contre, le modèle utilisé pour le calcul de la SBPI comporte m_l observations. Pour un

grand nombre d'observations, cette différence a tout de même de l'importance et c'est ainsi que $K = 500$ est optimal pour m_l observations mais pas pour $\frac{4}{5}m_l$.
Précisons quand même qu'étant donné l'ordre de cette différence, on peut considérer qu'effectuer une validation croisée mène à un résultat tout à fait cohérent et on retient cette procédure comme optimale pour le choix du paramètre K dans le KNN.

De nouveau, nous analysons l'aspect temporel du calcul de la SBPI à l'aide de la méthode des K plus proches voisins. Celui-ci est repris sur la figure .

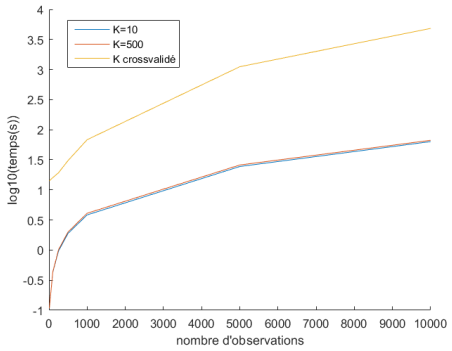


FIGURE 4.11 – Durée de calcul de la PI avec KNN

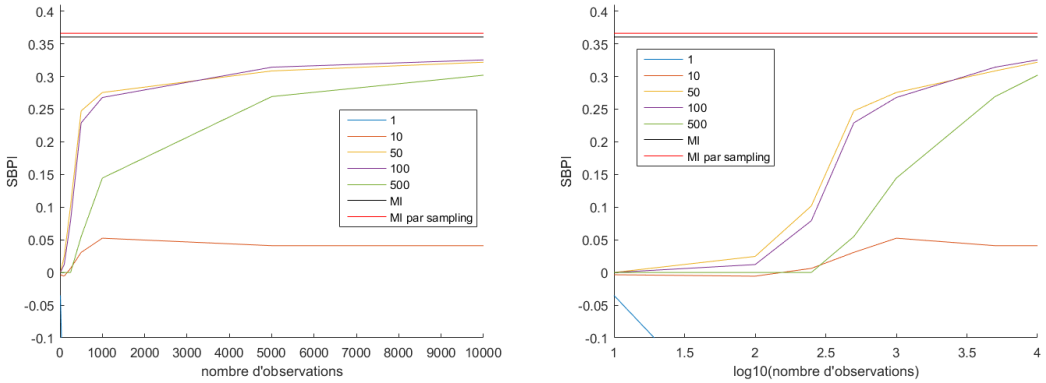
On constate que le temps de calcul ne varie pas beaucoup avec le nombre de voisins dans le modèle. La différence de temps entre $K = 10$ et $K = 500$ n'est que de 3 secondes avec 10000 observations.
Par contre, la validation croisée prend énormément de temps ! Pour la SBPI avec 10000 observations, le temps nécessaire est d'environ 4800 secondes, soit environ 1heure 20 minutes. Et notre modèle ne comporte que 3 utilisateurs. Pour 1000 observations par contre il ne faut que environ une minute ce qui est plus raisonnable.
La rapidité relative de cette méthode laisse présager qu'on puisse l'utiliser sur des bases de données contenant un nombre raisonnable d'observations, même pour un certain nombre d'utilisateurs.

4.2.5 Méthode des arbres de décision

Afin de pouvoir comparer les résultats avec une autre méthode de classification, on choisit de considérer les arbres de décision pour caractériser les distributions et calculer nos métriques.

MLS constant

On essaye d'abord de fixer la taille minimum des feuilles (MLS) à différents niveaux pour observer l'évolution de la SBPI. On choisit de tester pour 1, 10, 50, 100 et 500.



(a) SBPI (échelle linéaire) (b) SBPI (échelle logarithmique)

FIGURE 4.12 – Comparaison des SBPI et erreur de SBPI avec decision tree et différents MLS constants

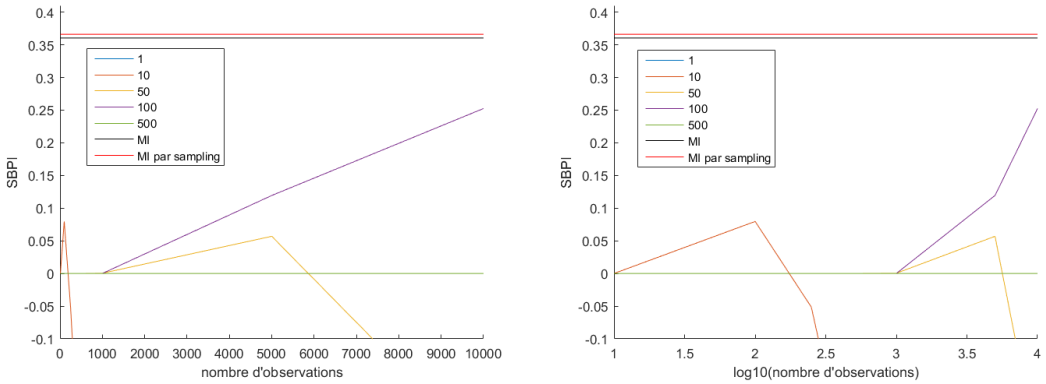
On remarque sur la figure 4.12 que les valeurs 1 et 10 ne sont clairement pas suffisantes. 500 semble un peu trop élevé et fournit des résultats moins bons qu'avec 50 et 100. Ces deux valeurs donnent une SBPI très similaire qui est cependant un peu en dessous de la valeur de la MI. L'erreur est d'environ quatre centièmes.
Avant 1000 observations, il est préférable de choisir 50 puis après on préférera 1000. A nouveau la meilleure courbe dépend du nombre d'observations mais on voit que des valeurs raisonnables fournissent tout de même une SBPI similaire et plutôt bonne et cela pour tout nombre d'observations.

MLS crossvalidé

A nouveau il semble utile d'utiliser la validation croisée pour déterminer la taille minimale des feuilles. On sépare encore en 5 folds et on teste 20 valeurs de MLS entre 10 et 100 étant donné qu'on remarque qu'une plus grande valeur n'est jamais bénéfique.

Niveaux de pruning constants

Un autre procédure consiste à choisir le nombre de niveaux à tailler dans l'arbre. On effectue donc les mêmes analyses sur ce paramètre au lieu de la taille des feuilles.



(a) SBPI (échelle linéaire) (b) SBPI (échelle logarithmique)

FIGURE 4.13 – Comparaison des SBPI et erreur de SBPI avec decision tree et niveaux de pruning constants

On a des résultats très différents en fonction du niveau qu'on choisit (4.13). On voit qu'avec 500 on a taillé tout l'arbre. La SBPI est donc nulle. Avec 1 et 10 l'erreur est énorme, tout comme avec 50 où elle ne fait qu'augmenter. La seule valeur qui permet d'avoir une courbe raisonnable est 100 et on a quand même une erreur de l'ordre du dixième avec 10000 observations. On comprend donc qu'il faut choisir très précisément le niveau de pruning pour avoir des résultats valables.

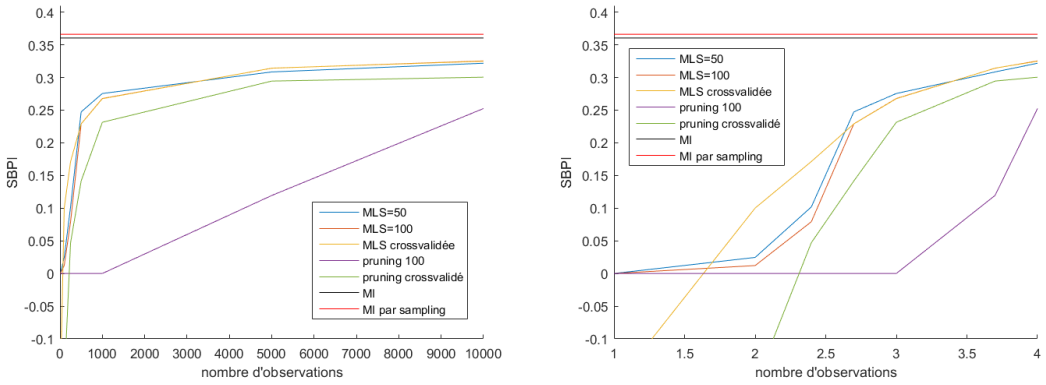
Niveau de pruning crossvalidé

Etant donné la sensibilité de la SBPI par rapport au niveau de pruning, on se permet de tester tous les niveaux possibles entre 1 et la taille maximale de l'arbre (moins de 500 comme on l'a vu).

Comparaison des modèles avec les decision trees

On compare les cinq techniques les plus prometteuses qu'on ait testées jusqu'à maintenant avec les arbres de décision :

- MLS=50
- MLS=100
- MLS crossvalidé
- niveau de pruning = 100
- niveau de pruning crossvalidé



(a) SBPI (échelle linéaire)

(b) SBPI (échelle logarithmique)

FIGURE 4.14 – Comparaison des SBPI et erreurs de sbPI avec decision tree

La première constatation est que n'importe quel niveau de pruning fournit des résultats moins bons que ceux obtenus en choisissant la taille maximale des feuilles. La meilleure valeur qu'on obtient est de seulement 0.3. On peut donc oublier cette méthode pour les arbres de décision.
On voit ensuite que la validation croisée n'apporte pas grand chose par rapport à une taille minimale de feuilles raisonnable. Le niveau de convergence est à peu près similaire et le niveau de SBPI est juste un peu meilleur jusque 500 observations.

On regarde ensuite l'aspect temporel pour les différents facteurs de croissance de l'arbre. Les résultats des figures 4.15 montrent que les méthodes les plus rapides sont celles impliquant des arbres de petite taille. Ainsi, si on ne fait pas de split de moins de 100 observations ou si on taille 100 niveaux de l'arbre, cet arbre n'a que peu d'étages et il ne faut donc que peu de temps pour obtenir les probabilités lors du calcul de la SBPI. Ce temps n'est que d'une dizaine de secondes même pour 10000 observations.
Les courbes pour MLS = 1 et niveau de pruning = 1 donnent des arbres très proches qui sont

aussi très profonds, il prennent un temps similaire et qui est de l'ordre d'une centaine de secondes pour 10000 observations.
La validation croisée prend encore un certain temps, on remarque que pour peu d'observations, sélectionner la taille des feuilles prend plus de temps que de sélectionner la bonne profondeur pour l'arbre mais pour des grands nombres d'observations, la tendance s'inverse et il est plus rapide de sélectionner la taille des feuilles. On peut expliquer cela par le fait que beaucoup d'observations entraînent un arbre plus profond et donc plus de niveaux de pruning à évaluer (étant donné qu'on les évalue tous). La validation croisée prend quelques 300 secondes pour la taille des feuilles avec 10000 observations et quelques 900 secondes pour la profondeur de l'arbre.

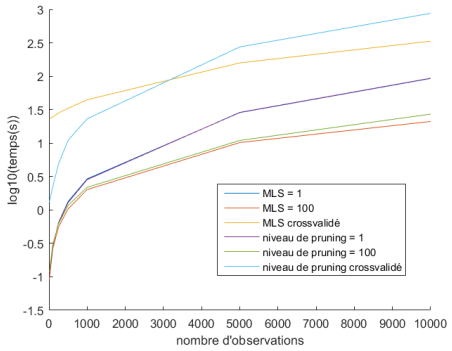
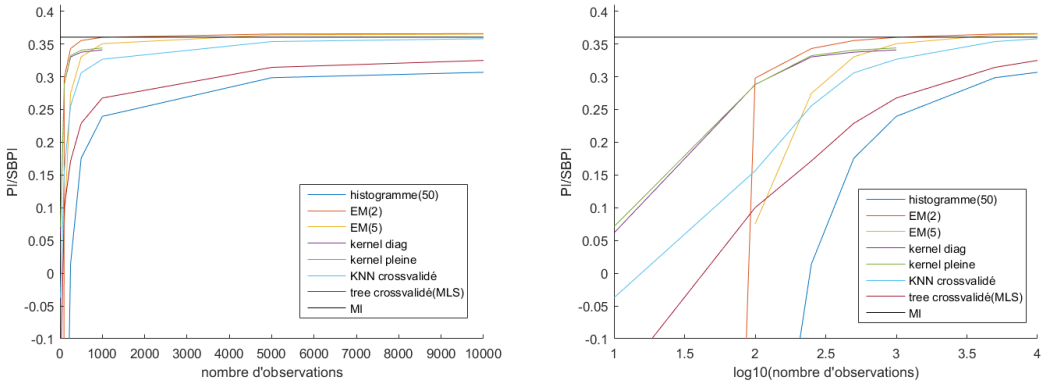


FIGURE 4.15 – Durée de calcul de la SBPI avec les arbres de décision

4.3 Comparaison des différentes méthodes

A présent qu'on a testé différents paramètres pour les méthodes servant à l'évaluation de nos métriques, on compare dans cette section les résultats qu'on a obtenus avec celles-ci pour les meilleurs choix de paramètres qu'on a testés. Ces analyses ont également été faites pour des autres bases de données simulées afin de confirmer les conclusions faites ici. Ces analyses se retrouvent dans l'annexe A.



(a) PI/SBPI (échelle linéaire) (b) PI/SBPI (échelle logarithmique)

FIGURE 4.16 – Comparaison des PI/SBPI avec les différents modèles

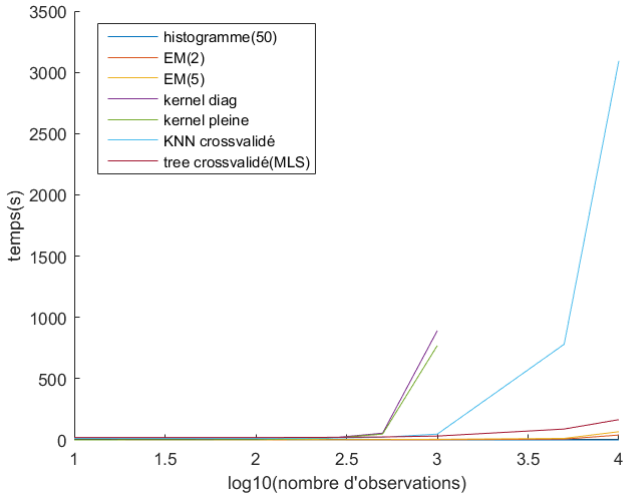


FIGURE 4.17 – Comparaison des temps de calcul de la PI/SBPI avec différents modèles

TABLE 4.1 – Ratio PI/MI (par sampling) et temps de calcul de la PI en fonction des méthodes et du nombre d'observations. La meilleure valeur de PI pour chaque nombre d'observations est indiquée en vert et les temps supérieurs à 1000 secondes sont indiqués en rouge

	50 observations		100 observations		1000 observations		10000 observations	
modèle	% PI	temps	% PI	temps	% PI	temps	% PI	temps
histogramme(50 bins)	0	1	0	1	65.4	1	83.78	1
kernel(0.25)	0	4	0	15	89.34	1464	/	/
kernel(1)	51.15	4	75.13	15	92.73	1443	/	/
kernel(Scott diagonale)	63.61	4	78.69	16	93.12	4853	/	/
kernel(Scott pleine)	62.2	4	78.6	14	93.95	3607	/	/
EM(2)	42.01	1	81.38	1	98.36	4	99.91	120
EM(5)	0	1	20.56	2	95.72	13	99.7	206
KNN(100)	0.57	1	9.08	1	93.6	4	95.12	65
KNN(500)	0	1	0	1	78.28	4	98.43	67
KNN(crossvalidé)	0	36	42.71	14	89.22	68	97.79	4833
tree(MLS= 50)	0	1	6.71	1	75.23	2	87.89	23
tree(MLS= 100)	0	1	3.31	1	73.15	2	88.84	21
tree(MLS crossvalidé)	0	22	27.35	21	73.07	45	88.74	335
tree(pruning = 100)	0	1	0	1	0	2	68.96	27
tree(pruning crossvalidé)	0	2	0	2	63.21	23	82.06	875

Avec peu d'observations, le kernel est très bon et rapide. La règle de Scott donne de bons résultats, que ce soit avec une covariance pleine ou diagonale. Ce modèle obtient de très bonnes performances quand on augmente le nombre d'observations également. Le problème est qu'il prend énormément de temps quand on considère beaucoup d'observations étant donné qu'obtenir la valeur de la densité en une observation nécessite de recalculer la somme de chaque noyau en ce point.

L'algorithme EM quant à lui nécessite plus d'observations pour fournir des résultats cohérents et donc une PI convenable. Avec beaucoup d'observations (à partir de 1000), la connaissance du modèle a priori (mixture de gaussiennes) fournit pourtant une information perçue encore plus grande qu'avec le modèle par noyau si on utilise un algorithme adapté (EM). On constate

que même avec un nombre trop grand de gaussiennes, la PI reste meilleure qu'avec le kernel étant donné que l'algorithme EM est capable de se ramener au nombre correct de gaussiennes en utilisant des "artifices". Le temps pris par l'algorithme est raisonnable ce qui permet de l'utiliser même pour de grands nombres d'observations en pouvant se permettre de surestimer le nombre de gaussiennes. Cependant cette méthode fonctionne bien pour ces données simulées car on connait le type de distribution et qu'il s'agit précisément de mixtures de gaussiennes. Dans le cas de données réelles, elle ne fonctionnera donc certainement pas aussi bien.

Le KNN fournit des résultats variables pour peu d'observations mais il se défend bien pour de plus grands modèles. La validation croisée prend évidemment du temps mais avec une valeur raisonnable de K, la durée de l'algorithme est très faible (2 à 3 fois plus faible que l'EM et largement plus faible que le kernel) et la SBPI atteint un très bon niveau. Dans le cas où on ne fait pas d'hypothèse sur la distribution et où on a beaucoup d'observations, c'est le classificateur qui semble préférable.

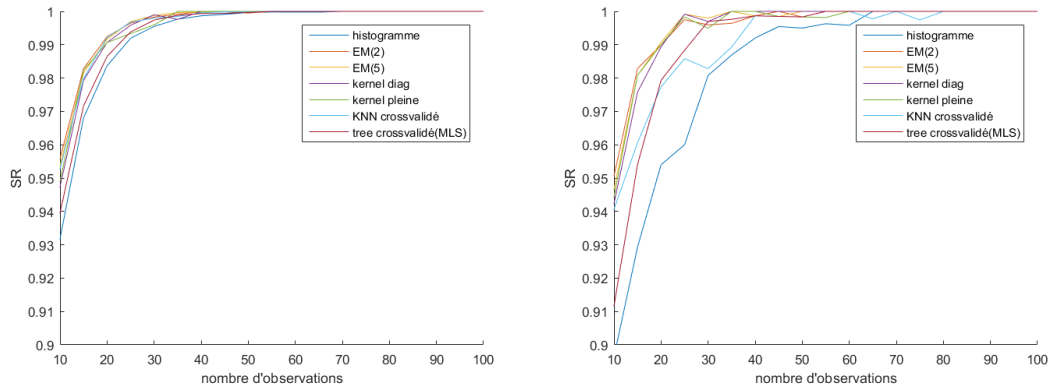
Les arbres de décisions ne fournissent quand à eux pas de résultats très encourageants. Ils nécessitent un grand nombre d'observations et même comme ça restent derrière le KNN. La validation croisée est cependant beaucoup plus rapide que pour ce dernier et cela peut constituer un bon avantage.

Le modèle en histogramme n'atteint pas une aussi bonne valeur de PI que les autres classificateurs, mais ce niveau reste élevé. De plus, son extrême rapidité permet sont utilisation sur de très grandes bases de données, ce qui n'est pas négligeable.

Globalement, tous les modèles qu'on a testés sur des données simulées permettent de caractériser correctement les utilisateurs. Le moins bon étant le modèle en histogramme qui parvient tout de même à obtenir deux tiers de la valeur de PI avec 1000 observations. Les arbres de décisions sont un peu plus performants avec n'importe quel nombre d'observations puis vient le KNN. Les modèles qui estiment les distributions en continu sont les plus performants même s'ils ont l'inconvénient d'être soit extrêmement lents soit de faire une hypothèse sur l'aspect des distributions et peuvent donc obtenir des résultats très variables. On conclut que toutes les méthodes testées permettent d'obtenir un modèle correct des utilisateurs.

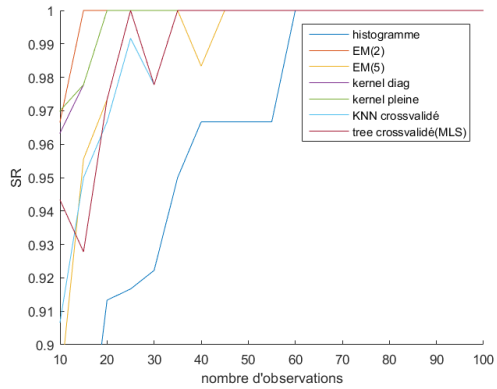
Taux de succès

Une fois les résultats analysés en ce qui concerne l'information perçue, analysons maintenant les courbes de taux de succès basées sur un modèle estimé à l'aide de 10000, 1000, 100 et 20 observations.

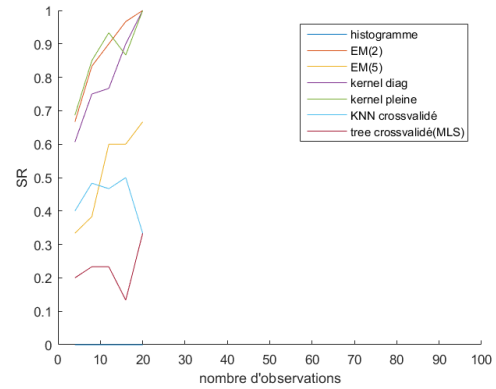


(a) Modèle avec 10000 observations

(b) Modèle avec 1000 observations



(c) Modèle avec 100 observations



(d) Modèle avec 20 observations

FIGURE 4.18 – Comparaison des SR avec les différents modèles et différents nombres d'observations

La conclusion principale à tirer des graphes sur les figures 4.18 est qu'avec un modèle basé sur au moins 100 observations, il suffit d'utiliser une cinquantaine d'observations de test pour avoir un succès garanti avec n'importe quel modèle. Dans ces conditions, on comprend que la location privacy est extrêmement peu élevée étant donné qu'on considère 100 observations comme largement faciles à obtenir.

On confirme également que le modèle par histogramme avec 50 bins est nettement moins bon que les autres et qu'il faut donc plus d'observations pour le faire fonctionner, même si il atteint lui aussi un taux de succès maximal.

La figure 4.18d confirme qu'il faut néanmoins un nombre minimum d'observations pour construire un modèle utilisable avec certaines méthodes, même si d'autres méthodes fournissent déjà de bons résultats avec seulement 20 observations, ce qui prouve une fois de plus l'absence de protection des utilisateurs de cette base de données simulées.

Conclusion

Dans cette partie d'analyse des résultats basée sur des données simulées, nous avons tout d'abord calculé de manière exacte et approchée l'information mutuelle, qu'il était dans ce cas possible d'obtenir grâce aux distributions réelles connues. Nous avons par la suite calculé l'information perçue et le taux de succès à l'aide de 3 estimateurs de fonction de densité différents : le modèle discret, le modèle par noyau et le modèle par maximisation d'espérance. Deux autres techniques de classification cette fois ont aussi été testées et évaluées à l'aide d'une information

perçue basée sur des scores et du taux de succès. Il nous semble assez évidemment de conclure que toutes les techniques d'attaques utilisées apportent de bons résultats autant au niveau de l'information perçue que du taux de succès. D'une part le niveau d'information perçue atteint à peu près l'information totale disponible dans les données, et ce peu importe la technique d'attaque. D'autre part on remarque que toutes permettent de retrouver avec un taux de succès de 1 à quel utilisateur appartiennent un ensemble d'observations dès lors que cet ensemble contiennent plus ou moins 20 observations, ce qui n'est pas énorme. Les courbes du taux de succès nous montrent par ailleurs qu'il n'est même pas nécessaire d'avoir un ensemble de 20 observations pour déterminer l'utilisateur lié à ces observations, si l'on accepte un taux de succès légèrement inférieur à 1.

Chapitre 5

Analyse des données réelles

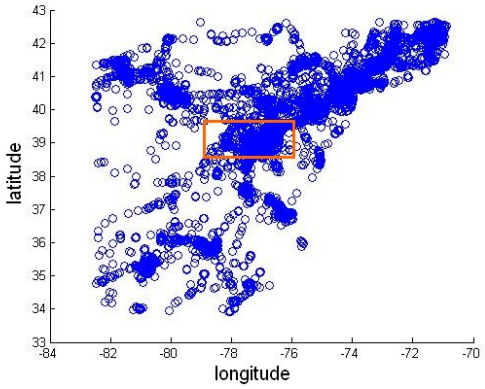
Maintenant que nous avons rendu les méthodes d'attaque les plus performantes possible et que nous avons réalisé notre étude comparative de celles-ci, nous pouvons passer à l'analyse des données réelles. Nous reprenons sur la figure 5.1b les données que nous allons analyser. Celles-ci ont été enregistrées sur la côte Est des Etats-Unis que l'on peut voir sur la figure 5.1a. Étant donné le temps d'exécution non négligeable de certaines méthodes d'attaque utilisées, nous rognerons l'espace sur lequel nous étudions les données afin d'obtenir un plus petit nombre d'utilisateurs présents dans cet espace. Nous prendrons la zone qui, à vue d'oeil, contient le plus d'observations afin d'obtenir un problème le moins "trivial" possible. L'espace conservé est observable sur la figure 5.1c et correspond à la ville de Washington. Seuls les utilisateurs pour lesquels on dispose d'au moins 1000 observations dans cet espace sont conservés et nous obtenons alors 27 utilisateurs dont nous ne gardons pour chacun que 1000 observations.

Dans ce chapitre nous allons tout d'abord analyser la base de données réelles brutes, c'est à dire démunie de tout moyen de défense. Nous analyserons l'information perçue et le taux de succès obtenus sur base de celles-ci. Ensuite nous chercherons à diminuer les valeurs atteintes par ces métriques à l'aide de moyens de défense puisqu'il est fort probable que les résultats obtenus sur les données soient alarmants, notamment au vu des conclusions du chapitre précédent. Pour réduire la capacité d'un attaquant à retrouver les utilisateurs nous testerons 2 méthodes de défense que nous avons présentées dans le chapitre 3 : **l'agrégation de l'espace** et **le bruitage de données**. Pour rappel, la première consiste à diviser l'espace en un nombre de bins défini et d'ainsi réduire la précision des données. Chaque observation contenue dans un bin prendra alors les coordonnées du centre de ce bin. La seconde méthode de défense consiste à ajouter un bruit gaussien aux données afin que celles-ci perdent à nouveau en précision. Nous analyserons l'efficacité de ces moyens de défense et parlerons de la quantité de précision qu'il est nécessaire de retirer afin de réellement diminuer le taux de succès de l'attaquant.

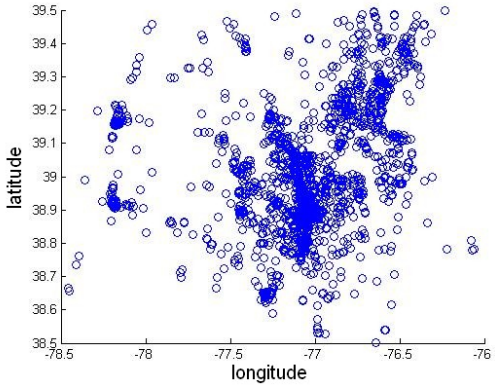
Pour faire ces analyses, nous utiliserons 3 méthodes développées précédemment : la méthode basée sur le modèle discret, celle des K plus proches voisins et celle de maximisation d'espérance. Puisque toutes les méthodes d'attaque testées dans le chapitre 4 obtiennent un succès global, nous avons choisi de garder seulement 3 de ces méthodes afin ne pas trop nous répéter. Il nous semblait cependant intéressant au niveau des analyses de garder au moins une technique liée à l'estimation d'un modèle et une liée à la classification. Le modèle discret et celui de maximisation d'espérance ont été préférés au Kernel car, bien que nous aurions préféré l'utiliser, il mettait beaucoup trop de temps à s'exécuter. La méthode des K plus proches voisins a quant à elle été préférée à celle des arbres de décision car les résultats liés à la première sont plus convaincants que ceux liés à la seconde, tant au niveau de l'information perçue basée sur les scores que du taux de succès. L'utilisation des ces différents d'outils nous permettra de comparer leur efficacité face à différentes méthodes de défense et de déterminer plus précisément le niveau de location

privacy des données étudiées.

(a) Côte Est des Etats-Unis



(b) Données originales



(c) Données rognées

FIGURE 5.1 – Données analysées

5.1 Données brutes

Pour les trois méthodes que nous avons choisi de reprendre dans ce chapitre, nous analyserons sous forme de tableau l'information perçue et le taux de succès. Afin d'avoir une estimation de la vitesse de convergence de l'information perçue et donc de la qualité du modèle, nous reprendrons les valeurs de celle-ci pour un nombre d'observations égal à 100 et 1000. Nous analyserons alors le taux de succès obtenu avec les probabilités ou les scores générés sur base des 1000 observations par utilisateurs. De nouveau, pour avoir une estimation de la vitesse de convergence du taux de succès et donc de l'influence du nombre d'observations de test, nous reprendrons les valeurs de ce taux pour 1, 100 et 1000 observations. Une autre façon intéressante d'analyser les données est de fixer un taux de succès souhaité et de regarder le nombre d'observations nécessaires pour atteindre cette valeur. Nous fixerons alors une valeur de taux de succès demandée égal à 0.7 ce qui nous semble suffisamment élevé dans certains contextes que pour caractériser un individu.

Ensuite, nous ferons les mêmes analyses sur le taux de succès obtenu cette fois avec un modèle basé sur seulement 100 observations par utilisateur. Nous pourrons alors voir s'il est nécessaire d'avoir accès à un nombre aussi élevé que 1000 observations pour construire le modèle ou si 100 sont suffisantes.

Regardons tout d'abord les résultats obtenus pour un modèle construit sur base de 1000 observations. Les valeurs obtenues pour les différentes méthodes d'attaque sont reprises dans le

tableau 5.1.

TABLE 5.1 – Analyse de l'information perçue pour un modèle basé sur 100 et 1000 observations, du taux de succès pour 1, 100 et 1000 observations avec modèle basé sur 1000 observations et du taux de probabilités nulles

DONNEES BRUTES	PI/SBPI		SR			Taux probas nulles
modèle	100 obs	1000 obs	1 obs	100 obs	1000 obs	
histogramme (25 bins)	1.43	1.6	0.27	0.98	1	0.00074
histogramme (100 bins)	1.9	2.26	0.4	0.99	1	0.0026
histogramme (400 bins)	2.2	2.73	0.53	1	1	0.0079
EM (5 gauss)	1.56	3.14	0.71	0.97	0.97	0
EM (10 gauss)	0.12	3.38	0.78	0.99	1	0
EM (20 gauss)	−2.33	3.54	0.8	0.99	1	0
KNN	2.21	3.15	0.86	1	1	0.095

Avec un modèle basé sur 1000 observations, les trois méthodes permettent d'atteindre un succès maximal pour 1000 observations de test et presque autant pour seulement 100. Il est impressionnant de voir que même avec une seule observation, le taux de succès avec EM va jusque 0.8 et peut même monter jusque 0.86 pour le KNN. L'histogramme quant à lui ne dépasse pas 0.53. Ces résultats montrent qu'il est aisé d'identifier le propriétaire des observations, et ce même pour un nombre très peu élevé d'observations.

Au niveau de la vitesse de convergence de l'information perçue, on remarque facilement que celle de l'EM est relativement faible, et ce au plus que le nombre de gaussiennes est élevé. La vitesse de convergence du KNN, bien que considérablement plus élevée que celle de l'EM, est elle même plus faible que celle de l'histogramme. Pour un nombre d'observations disponibles plus petit que 1000, on peut donc penser que la maximisation d'espérance serait probablement moins performante que le KNN lui même étant probablement moins performant que l'histogramme. Cette hypothèse se devra d'être vérifiée, puisqu'il est possible que l'information perçue calculée sur 100 observations soit surévaluée pour cause d'un taux de probabilités nulles trop élevé.

On regarde ensuite le nombre d'informations de test nécessaires pour obtenir un taux de succès arbitraire de 0.7 qu'on considère comme le seuil pour avoir un taux de succès raisonnable. On analyse cela grâce au tableau 5.2

TABLE 5.2 – Analyse du nombre d'observations nécessaires pour obtenir un taux de succès,calculé à l'aide d'un modèle estimé sur 1000 observations, de 0.7

DONNEES BRUTES	
modèle	# d'observations nécessaires
histogramme (25 bins)	10
histogramme (100 bins)	1
histogramme (400 bins)	1
EM (5 gauss)	1
EM (10 gauss)	1
EM (20 gauss)	1
KNN	1

A part pour l'histogramme utilisant 25 bins, tous les modèles ne nécessitent qu'une observation pour obtenir le taux de succès fixé comme seuil. De façon imagée, cela correspond à devoir

suivre une fois chaque utilisateur pour pouvoir les distinguer avec une probabilité de 70% si on a auparavant construit un modèle pour chacun d'eux avec 1000 observations.

Nous analysons ensuite le tableau 5.3 reprenant l'évolution du taux de succès basé sur un modèle utilisant maintenant seulement 100 observations. A l'aide de celui-ci nous pourrons discuter de l'hypothèse faite à l'instant et voir si le modèle peut être correct pour un nombre si peu conséquent d'observations.

TABLE 5.3 – Analyse : de l'information perçue pour un modèle basée sur 100 observations, du taux de succès pour 1 et 100 observations avec modèle basé sur 100 observations, du taux de probabilités nulles

DONNEES BRUTES	PI/SBPI	SR		Taux probas nulles
modèle	100 obs	1 obs	100 obs	
histogramme (25 bins)	1.43	0.02	0.66	0.016
histogramme (100 bins)	1.9	0.04	0.81	0.03
histogramme (400 bins)	2.2	0.05	0.91	0.05
EM (5 gauss)	1.56	0.07	0.94	0.00074
EM (10 gauss)	0.12	0.08	0.91	0.001
EM (20 gauss)	−2.33	0.08	0.86	0.003
KNN	2.21	0.81	1	0.15

Avec un modèle basé sur seulement 100 observations, les résultats sont évidemment moins bons du point de vue des attaquants. Dans ce cas, seul le KNN atteint un succès maximal pour 100 observations, et celui-ci atteint même un taux de 0.81 pour 1 observation, ce qui est énorme au vu des autres résultats. Cependant, la présence d'une fraction plus que conséquente d'outliers nous indique qu'il y a un grand nombre d'observations que le modèle n'est pas en mesure d'expliquer et qu'il est probable que les valeurs obtenues soient alors irréalistes. Nous analysons donc les valeurs obtenues par la méthode de l'histogramme et celle de l'EM. Au niveau du taux de succès, on arrive avec les deux méthodes à une valeur de l'ordre de 0.9 pour 100 observations. Les valeurs d'information perçue sont pourtant beaucoup plus élevées dans le cas des histogrammes que dans celui de l'EM. L'explication vient du taux de probabilités nulles trop élevé dans le cas des histogrammes. Il s'avère donc que la valeur de l'information perçue a été surévaluée et que celle-ci est visiblement beaucoup plus sensible à ce taux de probabilités nulles que le taux de succès. Pour revenir sur l'hypothèse faite précédemment quant à la vitesse de convergence plus lente de l'EM, les résultats sont en fait a priori trop faussés pour à la PI avec peu d'observations que pour pouvoir conclure quoi que ce soit. Nous n'observerons donc plus la vitesse de convergence par la suite et nous analyserons le taux de succès basé sur une PI, et donc un modèle, pour 1000 observations.

On effectue de nouveau la même analyse que précédemment pour savoir le nombre d'observations nécessaires pour obtenir un taux de succès de 70% mais cette fois sur un modèle construit sur 100 observations. Les résultats se trouvent dans le tableau 5.4.

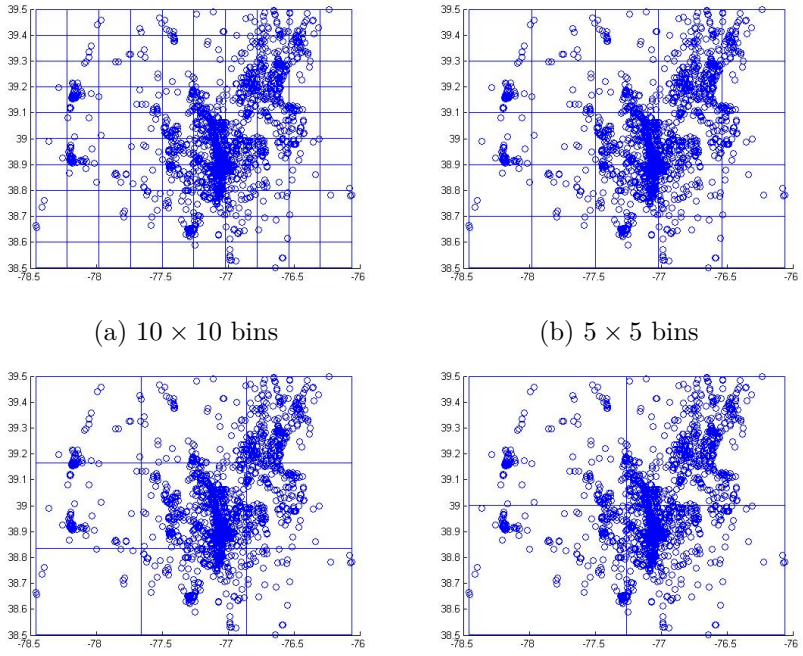
TABLE 5.4 – Analyse du nombre d'observations nécessaires pour obtenir un taux de succès,calculé à l'aide d'un modèle estimé sur 100 observations, de 0.7

DONNEES BRUTES	
modèle	# d'observations nécessaires
histogramme (25 bins)	> 1000
histogramme (100 bins)	50
histogramme (400 bins)	30
EM (5 gauss)	21
EM (10 gauss)	21
EM (20 gauss)	21
KNN	1

Si on en croit nos valeurs, le meilleur modèle, c'est à dire celui obtenu par KNN permettrait d'identifier le bon utilisateur à 70% avec à nouveau une seule observation. A nouveaux les outliers nous empêchent de croire cette valeur et dans les autres cas, plus crédibles, il faut entre 20 et 30 observations pour obtenir ce taux seuil. Ce nombre est en accord avec la qualité moins bonne du modèle étant donné qu'il est créé sur moins d'observations.

5.2 Données agrégées

Afin de diminuer la possibilité de caractériser les utilisateurs de notre base de données, l'agrégation de l'espace est utilisée comme premier moyen de défense. Pour analyser l'efficacité de celle-ci et observer comment les résultats de l'attaquant varient face à ces changements, nous séparerons l'espace des données en un nombre variable de bins. Nous diviserons tout d'abord l'espace en 10×10 bins, ce qui semble être encore représentatif des points d'origine, ensuite en 5×5 bins, 3×3 bins et enfin 2×2 bins. Les découpes dans l'espace se font comme sur la figure 5.2. Nous analyserons l'évolution de l'information perçue et du taux de succès pour un modèle construit à l'aide des 1000 observations. Ces analyses seront faites pour les quatre valeurs d'agrégation, et ce pour chacune des 3 méthodes choisies : modèle discret, KNN et modèle par maximisation d'espérance. Le modèle discret sera utilisé avec 25 bins pour obtenir un taux de probabilités nulles le moins élevé possible. La maximisation d'espérance sera quant à elle utilisée avec 10 gaussiennes car les résultats sur les données brutes sont plus élevés qu'avec seulement 5 gaussiennes et car le temps de calcul est non négligeable pour 20 gaussiennes. Au vu des résultats détaillés disponibles en annexe, l'agrégation en 100 bins n'augmente pas beaucoup la location privacy. Nous ne reprendrons alors pas ici les résultats liés à cette agrégation.



(a) 10×10 bins (b) 5×5 bins

(c) 3×3 bins (d) 2×2 bins

FIGURE 5.2 – Agrégation de l'espace en différents nombre de bins

Regardons alors sur le tableau 5.5 le niveau de location privacy qu'il est possible de gagner à l'aide d'une agrégation en 4, 9, et 25 bins.

TABLE 5.5 – Analyse pour différents nombre de bins de l'information perçue pour un modèle basé sur 1000 observations, du taux de succès pour 1000 observations avec modèle basé sur 1000 observations, du taux de probabilités nulles

DONNEES AGREGEES	PI/SBPI (1000 obs)			SR (1000 obs)			Taux probas nulles		
modèle	4 bins	9 bins	25 bins	4 bins	9 bins	25 bins	4 bins	9 bins	25 bins
histogramme (25 bins)	1.01	1.16	1.6	0.85	0.85	1	0.0001	0.0005	0.00074
EM (10 gauss)	< -1	< -1	< -1	0.13	0.28	0.41	0	0	0
KNN	0.81	1.06	1.31	0.4	0.72	0.84	0.0057	0.028	0.092

Bien qu'avec une agrégation de 25 bins la méthode de défense permette de réduire le taux de succès à 0.84 pour le KNN et même 0.41 pour la maximisation d'espérance, elle n'empêche pas l'histogramme d'atteindre un taux de succès unitaire. On peut donc dans ce cas précis identifier à nouveau tous les utilisateurs. Dans le cas d'une agrégation en 4 ou 9 bins, on peut réduire ce taux à 0.85 mais pas en dessous. On tire donc encore beaucoup d'information sur les utilisateurs malgré une agrégation de l'espace très élevée. On peut alors conclure que même si l'agrégation permet de réduire le niveau autant d'information perçue que de taux de succès dans le cas du KNN et de l'EM, elle ne permet pas globalement d'augmenter considérablement le niveau de location privacy. En effet, le modèle en histogramme garde des valeurs toujours plus élevées que les autres méthodes d'attaque.

Etant donné que le modèle en histogramme utilise précisément la même méthode d'attaque que celle de défense, il est logique qu'il ne soit pas affecté outre mesure par cette technique de défense et qu'il conserve de bons résultats.
Au contraire, les deux autres modèles sont plus affectés par l'agrégation des données. Le KNN est contraint d'utiliser comme voisins des observations ayant toutes les mêmes coordonnées

puisqu'elles sont probablement dans un même bin. Elles se trouvent donc toutes à la même distance que lui et il ne peut alors plus en trouver K "plus proches". On comprend donc que les résultats ne soient pas particulièrement bons.
L'algorithme EM quant à lui est contraint d'essayer d'ajuster des gaussiennes avec seulement 4, 9 ou 25 points différents. Difficile dans ce cas pour lui de calculer des moyennes et des covariances adaptées avec si peu de variabilité dans les données.

On s'attache ensuite à nouveau au nombre d'observations qui permettent d'obtenir un taux de succès de 70%, pour nos données agrégées cette fois. Cette analyse peut se faire grâce à la table 5.6

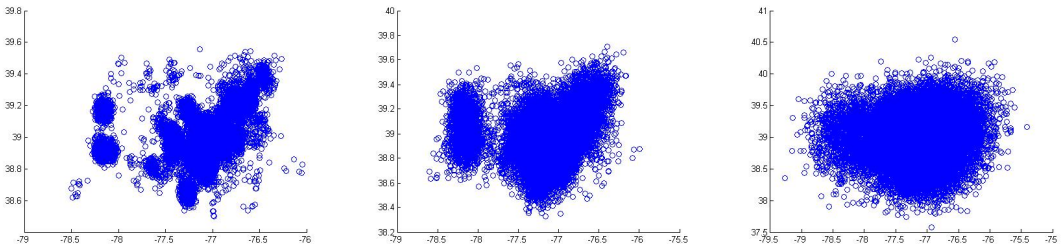
TABLE 5.6 – Analyse du nombre d'observations nécessaires pour obtenir un taux de succès, calculé à l'aide d'un modèle estimé sur 1000 observations, de 0.7

DONNEES AGREGEES	# d'observations nécessaires		
modèle	4 bins	9 bins	25 bins
histogramme (25 bins)	275	80	10
EM (10 gauss)	> 1000	> 1000	17
KNN	> 1000	600	9

Pour le meilleur modèle, c'est à dire l'histogramme, on a quand même besoin de 275 observations pour obtenir le taux de succès voulu lorsqu'on agrège très fortement les observations. On considère que cette valeur est tout de même assez élevée. Pour 9 bins il n'en faut par contre que 80 ce qui est plus raisonnable alors que les autres méthodes requièrent elles trop d'observations. On remarque sans surprise que le modèle avec 25 bins est suffisamment bon pour qu'une vingtaine d'observations soient largement suffisantes à l'identification des utilisateurs avec le taux recherché.

5.3 Données bruitées

Une autre technique de défense est d'ajouter du bruit aux données. Dans notre cas nous allons bruiter les données en ajoutant une petite valeur en latitude et en longitude : un bruit gaussien de variance v. Nous ferons varier cette variance et analyserons les résultats obtenus selon les différentes méthodes d'attaque. Nous utiliserons trois valeurs de v : 0.001, 0.01 et 0.1, ce qui nous donnera les données visibles sur la figure 5.3. A priori au plus le bruit est élevé au moins les données sont représentatives de ce qu'elles étaient à la base et au moins l'information perçue et le taux de succès ne devraient être élevés.



(a) $v = 0.001$ (b) $v = 0.01$ (c) $v = 0.1$

FIGURE 5.3 – Bruitage des données avec des bruits v différents

Le tableau 5.7 reprend à nouveau la méthode en histogramme utilisant 25 bins et la maximisation d'espérance pour 10 gaussiennes. Les raisons de ce choix sont les mêmes que pour les données agrégées.

TABLE 5.7 – Analyse pour différents niveaux de bruit de l'information perçue pour un modèle basé sur 1000 observations, du taux de succès pour 1000 observations avec modèle basé sur 1000 observations, du taux de probabilités nulles

DONNEES BRUITEES	PI/SBPI (1000 obs)			SR (1000 obs)			Taux probas nulles		
modèle	$v = 0.1$	0.01	0.001	$v = 0.1$	0.01	0.001	$v = 0.1$	0.01	0.001
histogramme (25 bins)	0.34	0.84	1.31	0.66	0.89	1	0.02	0.02	0.001
EM (10 gauss)	0.41	1.27	2.28	0.7	0.85	0.96	0	0	0
KNN	0.36	1.19	1.7	0.3	0.85	0.89	0	0.009	0.047

Pour les données bruitées, les résultats sont un peu plus encourageants puisque avec un bruit de 0.1, on peut faire chuter le taux de succès à une valeur maximale de 0.7. Cette valeur reste tout de même considérable et prouve encore une fois qu'il est difficile d'obtenir niveau élevé de location privacy. Les résultats montrent qu'il est légèrement préférable d'utiliser la maximisation d'espérance lorsque le bruit est élevé. Globalement les 3 méthodes fournissent des résultats comparables à part le KNN dont le taux de succès chute à 0.3 dans le cas d'un bruit élevé.

On remarque tout de même que le modèle en histogramme produit un taux d'outliers de un sur cinquante ce qui est relativement élevé et pourrait donc nous faire croire que les résultats liés à celui-ci sont en fait faussés. On se fiera donc plutôt aux résultats de l'EM, malheureusement tout aussi convaincants qui confirment donc que le niveau de location privacy est peu satisfaisant.

On analyse à nouveau le nombre d'observations nécessaires à l'obtention d'un taux de succès de 70%, pour les données bruitées cette fois. On se base sur le tableau 5.8.

TABLE 5.8 – Analyse du nombre d'observations nécessaires pour obtenir un taux de succès,calculé à l'aide d'un modèle estimé sur 1000 observations, de 0.7

DONNEES BRUITEES	# d'observations nécessaires		
modèle	$v = 0.1$	0.01	0.001
histogramme (25 bins)	> 1000	49	13
EM (10 gauss)	1000	19	3
KNN	> 1000	600	9

Avec un bruit de 0.1, il faut énormément d'observations pour obtenir ce taux de succès. Avec 0.01, l'histogramme et le modèle par EM permettent néanmoins d'atteindre cette valeur avec 50 et 20 observations ce qui est facilement envisageable. Enfin, avec le plus petit bruit analysé, les trois méthodes ne nécessitent que peu d'observations pour atteindre les 70% de taux de succès.

On conclut alors que les résultats d'attaque restent dans l'ensemble très bons et que cette méthode de défense ne protège pas énormément les données compte tenu de la dénaturation que le bruit leur inflige. Augmenter davantage ce bruit pour diminuer la caractérisation qu'on peut faire des utilisateurs modifierait encore plus les données et les rendrait encore plus inutilisables, ce qui n'est pas forcément souhaitable. On souligne tout de même la capacité de cette méthode à augmenter considérablement le nombre d'observations nécessaires pour atteindre un taux de 0.7. Malgré cela, on se rend donc à nouveau compte de la difficulté de rendre non identifiables les utilisateurs présents dans les bases de données.

Conclusion

Au vu de ce chapitre, il est assez évident que les résultats obtenus sur base de données réelles confirment ce qui est dit dans la littérature et que nous avons pu obtenir dans le cas de données

simulées : les techniques d'attaque permettent de retrouver des utilisateurs avec une facilité marquante lorsque celles-ci ne sont pas protégées. Dans leur état brut, les données disponibles dans la base étudiée permettent une bonne caractérisation des utilisateurs présents dans celle-ci. On a montré que peu importe la méthode d'attaque utilisée, l'information perçue atteignait une valeur élevée, de même que le taux de succès atteignant souvent même une valeur maximale. A titre d'exemple, si on crée le modèle sur 1000 observations, une seule est nécessaire pour obtenir un taux de succès de 70%.

En appliquant les mécanismes de défense aux données, il est possible d'augmenter le niveau de location privacy mais pas de manière marquante, et ce aux dépens d'une perte considérable de précision des données.

Avec l'agrégation des données, on arrive à diminuer le taux de succès à une valeur de 0.85, obtenu avec la méthode d'attaque en histogramme, mais seulement en dégradant abusivement les données. C'est à dire en utilisant 9 ou 4 bins. Cependant, cette valeur de 0.85 nécessite un grand nombre d'observations. Ainsi, même pour 70%, il faut tout de même 275 observations, ce qui n'est pas si évident à obtenir.

Avec l'application d'un bruit de variance supérieure à 0.001, on arrive aussi à diminuer un le taux de succès : jusque 0.7 avec une variance de 0.1 et 0.89 avec une variance de 0.01. Les niveaux de taux de succès restent donc tout de même élevés alors qu'à nouveau, les observations ne ressemblent plus aux données d'origine. Par contre, ce taux de succès nécessite un grand nombre d'observations pour être obtenu : 1000. Cela donne quand même un petit intérêt à cette méthode.

Dans l'ensemble, les données brutes permettent une caractérisation bien trop précise des utilisateurs et les mécanismes de défenses permettent difficilement de pallier à cela. Si certains de ceux-ci sont efficaces contre l'une ou l'autre méthode, il reste toujours des méthodes d'attaque capables d'obtenir un taux de succès toujours plus grand que 0.7. Cependant, obtenir cette valeur requière l'utilisation d'un grand nombre d'observations qui peuvent se révéler compliquées à obtenir. Un autre aspect à prendre en compte est que si ces mécanismes augmentent la location privacy, ils le font au prix d'une dénaturation excessive des données. Il n'est donc même pas garanti qu'on puisse encore s'en servir dans un cas concret, ce qui enlèverait l'intérêt de conserver de telles données de géolocalisation.

Conclusion

Au départ de ce travail, nous avons observé qu'avec l'augmentation croissante du nombre de données récoltées par les nouvelles technologies, le droit d'être anonyme était aujourd'hui remis question. En effet, ces données sont stockées au quotidien dans des bases de données, et parfois à notre insu. Il s'avère que divers exemples démontrent qu'il est possible d'utiliser ces informations et d'avoir alors atteinte à la vie privée des individus. Notamment à l'aide d'observations sur les déplacements géographiques. Il est donc important de se rendre compte du niveau de location privacy que présentent les bases de données et d'essayer de l'augmenter afin de se prémunir des atteintes à notre vie privée.

Etant donné qu'il n'existe pas de méthode systématique pour analyser et surtout quantifier l'anonymat dans les bases de données, nous avons remarqué la diversité des outils et plus précisément des métriques permettant d'y arriver. Nous avons choisi d'utiliser l'information perçue afin de voir les choses sous un aspect relativement novateur puisqu'il donne une idée du niveau de caractérisation des utilisateurs et non de distinguabilité. Le taux de succès nous permettait alors d'analyser la capacité à identifier l'individu ayant généré une ou un ensemble d'observations.

Puisque les métriques choisies nécessitent l'utilisation d'une méthode d'attaque spécifique, nous avons réalisé une étude comparative faite sur une base de données simulées. L'intérêt de celles-ci étant d'avoir accès aux distributions réelles des utilisateurs. Nous avons alors pu tester plusieurs méthodes d'attaque, étant ici des méthodes de modélisation et de classification : la méthode utilisant un modèle discret, la méthode utilisant le modèle par noyau, la méthode utilisant la maximisation d'espérance, la méthode des K plus proches voisins et enfin celle des arbres de décision. A l'aide des ces différentes attaques, nous avons pu déceler les avantages et les inconvénients de chacune des méthodes. La méthode utilisant le modèle par noyau et celle utilisant la maximisation d'espérance sont celles donnant les meilleurs résultats. Le modèle par noyau est avantageux car il ne fait aucune hypothèse quant à la forme des distributions à estimer. Il a cependant un temps de calcul conséquent ce qui pose problème pour des données de grandes dimensions. La méthode de maximisation d'espérance a quant à elle un temps d'exécution acceptable mais fait l'hypothèse d'un modèle gaussien. Elle dépend donc d'un paramètre pouvant modifier les performances de la méthode si nous ne connaissons pas a priori le nombre de gaussiennes ou si les véritables distributions sont trop éloignées d'une mixture de gaussiennes. La méthode utilisant le modèle discret montre des résultats un peu moins élevés, mais a quant à elle un aspect temporel très avantageux qui facilite réellement l'analyse de données à échelle réelle. Les résultats d'information perçue lui étant liés sont par contre souvent biaisés pour un nombre de bins trop élevé et/ou un nombre d'observations analyses trop faible. En effet, ces paramètres entraînent un taux de probabilités nulles non négligeable. Concernant les méthodes de classification c'est à dire celle des K plus proches voisins et celle des arbres de décision, elles sont toutes deux comparables bien que les résultats liés au K plus proches voisins sont légèrement plus élevés. L'aspect temporel est ici aussi non négligeable bien que plus abordable que pour le modèle par Noyau. Globalement, toutes les méthodes utilisées sont dans un même ordre de grandeur quant au résultats obtenus, bien que l'une ou l'autre ait ses avantages et ses inconvénients. Toutes permettent, dans le cas de nos données simulées, de bien repérer quel

utilisateur est lié à quelles observations, et ce avec une précision surprenante. Il est évident que ces tests n'ont été réalisés que sur des bases de données simulées spécifiques, elles ne nous permettent donc pas de conclure en toute généralité que les méthodes d'attaque seront performantes à ce point sur toutes les bases de données, mais elles nous laissent la crainte que ce soit bien possible.

Dès lors que nous savions que nos méthodes d'attaque permettaient l'identification des utilisateurs sur des données simulées, nous pouvions analyser une base de données réelles. Celle-ci est issue d'une application pour smart phones. Nous avons dans un premier temps cherché à découvrir la quantité d'information disponible sur les utilisateurs. Au vu de l'étude comparative faite sur les méthodes d'attaque, nous avons pu choisir les méthodes qui seraient a priori les plus efficaces pour analyser la base de données. Nous utilisons donc la méthode avec modèle discret, la maximisation d'espérance et les K plus proches voisins. Les résultats sont comme attendus, effrayants : les utilisateurs sont presque toujours identifiables et peuvent être identifer avec une probabilité de 70% avec seulement 1 observation (pour un modèle estimé à l'aide de 1000 observations). Notre analyse se porte alors inévitablement sur la mise en place de deux systèmes de défense afin d'augmenter le niveau de location privacy : l'agrégation et le bruitage des données. A nouveau ici, les résultats sont plus qu'interpellants. Malgré une diminution de précision des données modifiant totalement leur disposition spatiale, que ce soit pour l'agrégation ou le bruitage des données, on ne peut jamais augmenter considérablement la location privacy. Avec la connaissance de 1000 observations, le modèle en histogramme étant très performant face à l'agrégation, donne un taux de succès de 0.85 pour un nombre de bins de seulement 4. Pour le bruitage de données, avec 1000 observations à nouveau, la maximisation d'espérance atteint un taux de succès de 0.7. On peut cependant noter le point positif étant que ces méthodes de défense augmentent considérablement le nombre d'observations de test nécessaire afin d'obtenir un taux de succès si bon. Il faut par exemple 275 observations pour identifier un utilisateur avec une probabilité de 0.7 dans des données agrégées en 4 bins. Et ce nombre va jusque 1000 observations pour des données bruitées avec une variance de 0.1.

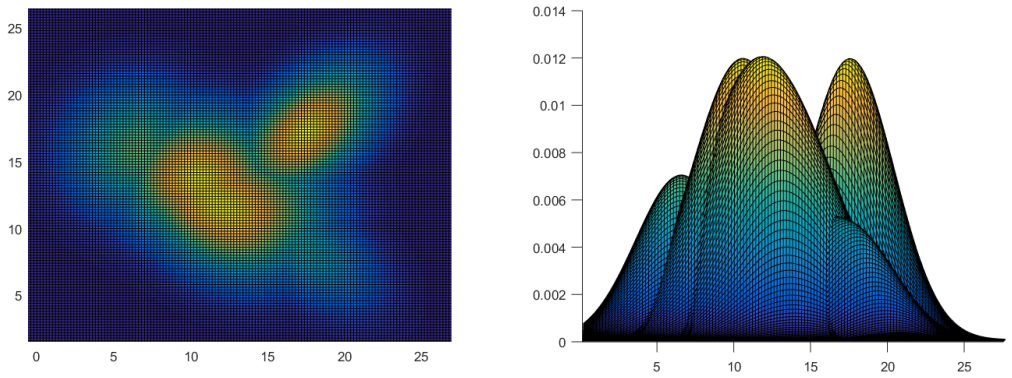
Au vu des différents types d'attaques qu'il est possible d'utiliser, il est d'autant plus difficile de protéger correctement une base de données. En effet, il faut se protéger contre tout type d'attaque, puisque là où une méthode de défense est utile contre une méthode d'attaque, elle peut s'avérer être peu performante contre d'autres. On remarque par exemple que l'agrégation diminue fortement les résultats pour l'attaque par maximisation d'espérance et par KNN, mais ne contre pas énormément celle en histogramme. Au contraire, le bruitage de données est plus performant contre le modèle en histogramme que contre la maximisation d'espérance. On pourrait donc envisager d'utiliser simultanément plusieurs types de protections afin de contrecarrer tous les types d'attaques potentielles. Malheureusement, qui dit ajouter des protections dit altérer les données et donc risquer de les rendre inexploitables. Il y a donc un compromis à trouver.

Les recherches semblent actives afin de trouver une solution à ce manque d'anonymat. Pierre Deville propose par exemple l'idée de ne garder en mémoire non plus les données de chaque utilisateur et les endroits où ils sont présents mais uniquement le nombre d'utilisateurs présents à un certain endroit. Avec ce système il semblerait que d'une part, les données ne sont plus sujettes au manque d'anonymat et d'autre part, elles gardent leur intérêt. En effet, on pourra par exemple toujours savoir quel endroit est fortement fréquenté par la population et donc nécessitera plus de réseau. Malgré les solutions tendant à être imaginées afin de supprimer ce problème de location privacy, l'humanité semble tout de même avoir encore fort à faire pour arriver à concilier l'abondance des données et l'envie de les utiliser, avec le besoin de vie privée et les lois émergeant à cet effet.

Annexe A

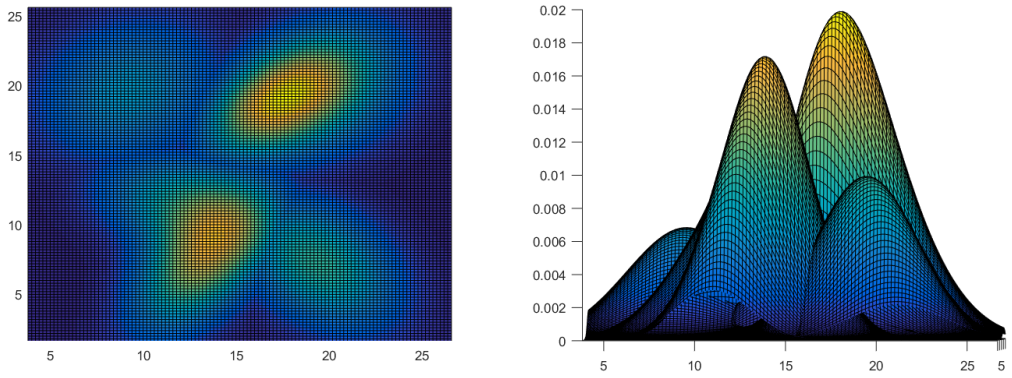
Validation des résultats théoriques

Dans ce chapitre, nous avons tout le temps utilisé la même distribution et les mêmes observations. On pourrait donc se demander si les résultats sont consistants où s'ils sont liés à notre cas particulier et donc biaisés.
Pour éclaircir ce problème, on compare avec deux autres distributions aléatoires contenant elles aussi 3 utilisateurs avec chacun 2 gaussiennes. Les moyennes, covariances et proportions de chaque gaussiennes sont par contre bien différentes de celles des distributions utilisées dans ce chapitre. Les distributions sont illustrées sur les figures A.1 et A.2.



(a) Vue du haut (b) Vue de profil

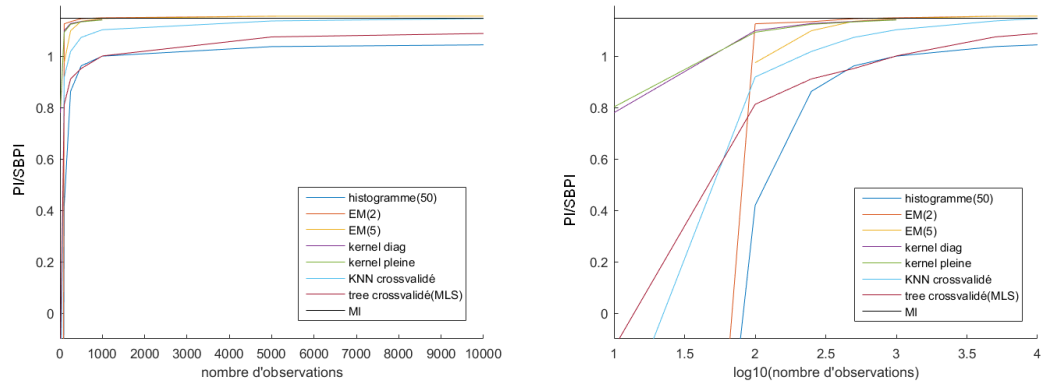
FIGURE A.1 – Distributions réelles des utilisateurs pour le deuxième jeu de paramètres



(a) Vue du haut (b) Vue de profil

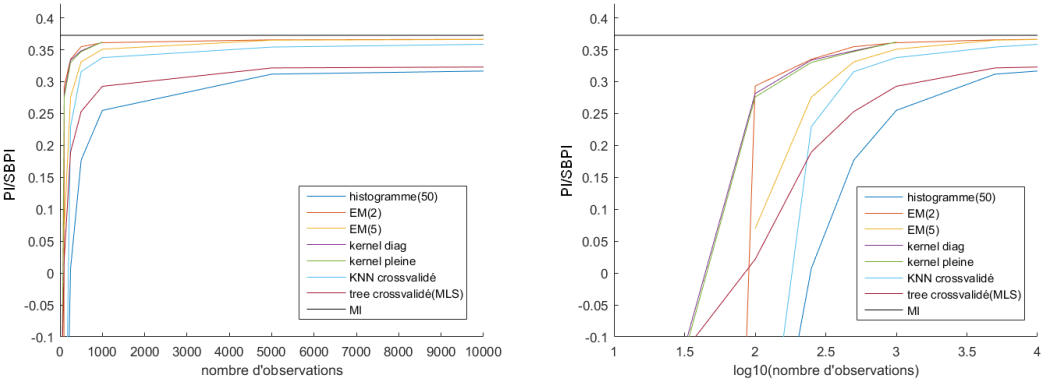
FIGURE A.2 – Distributions réelles des utilisateurs pour le troisième jeu de paramètres

Les distributions de la figure A.1 sont fort groupées et présentent des covariances plus grandes que celles étudiées dan ce chapitre. Celles de la figureA.2 quant à elles sont fort écartées et on toujours des grandes covariances. On confirme donc que les 3 distributions étudiées sont fort différentes et couvrent une large gamme parmi celles possibles en gardant le même nombre d'utilisateurs et le même nombre de gaussiennes.



(a) PI/SBPI (échelle linéaire) (b) PI/SBPI (échelle logarithmique)

FIGURE A.3 – Comparaison des PI/SBPI avec les différents modèles(2)



(a) PI/SBPI (échelle linéaire) (b) PI/SBPI (échelle logarithmique)

FIGURE A.4 – Comparaison des PI/SBPI avec les différents modèles(3)

On constate que les observations qu'on avait faites sont valables pour les 3 séries de distributions et d'observations. Dans certains cas l'une ou l'autre méthode est un peu meilleure qu'une autre sur l'un des graphes, par exemple sur les figures A.3, le modèle par histogramme dépasse le modèle par arbre de décision autour de 1000 observations alors qu'avec les distributions de ce chapitre, le choix des arbres de décision était tout le temps meilleur (4.16). Cependant, ce ne sont que des différences ponctuelles ou présentes seulement pour peu d'observations. On considère donc que nos conclusions du chapitre 4 restent valables pour des distributions gaussiennes et des observations générales.

Annexe B

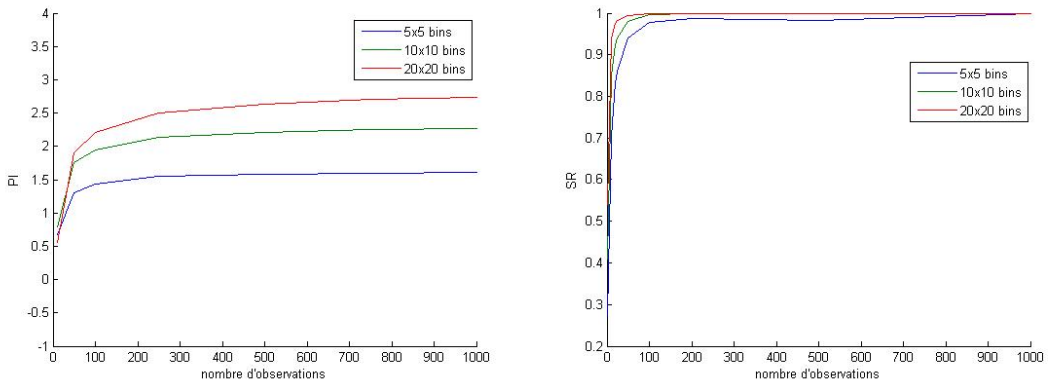
Analyse complète des méthodes sur les données réelles

B.1 Données brutes

Pour les trois méthodes que nous avons choisi de reprendre ici, nous analyserons l'évolution de l'information perçue et du taux de succès. Nous analyserons le taux de succès obtenu avec les probabilités ou les scores liés aux 1000 observations des utilisateurs et également celui obtenu avec 100 observations des utilisateurs. Nous pourrons alors voir s'il est nécessaire d'avoir accès à un nombre aussi élevé que 1000 observations ou si 100 d'entre elles sont suffisantes pour obtenir des résultats corrects.

Modèle discret

Pour le modèle discret, nous avons déjà remarqué que le nombre de bins pris en compte par l'attaquant afin de caractériser les utilisateurs change les résultats obtenus, tant au niveau de l'information perçue qu'au niveau du taux de succès. Nous savons également qu'au plus le nombre de bins pris en compte est élevé, au plus l'hypothèse sur les probabilités nulles modifie les résultats. Nous allons donc observer l'évolution de l'information perçue et du taux de succès pour des valeurs de 5×5, 10×10 et 20×20 bins.



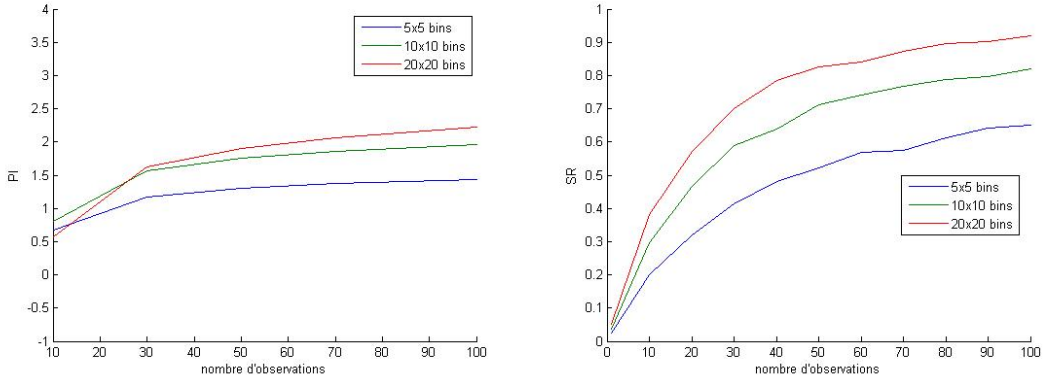
(a) Evolution de la PI en fonction du nombre d'observations

(b) Evolution du SR en fonction du nombre d'observations

FIGURE B.1 – Analyse des résultats en fonction du nombre de bins utilisés par le modèle en histogrammes

On observe, comme dans le cas des données simulées, qu'au plus le nombre de bins utilisés est élevé, au plus l'information perçue converge vers une valeur élevée, mais elle converge moins vite. On remarque que le taux de succès converge vers 1 peu importe le nombre de bins utilisés mais atteint ce résultat plus rapidement si le nombre de bins est plus grand et donc si l'information perçue avec 1000 observations est plus grande. Le taux de probabilités nulles est ici respectivement de 0.00074, 0.0026 et 0.0079 pour 5×5, 10×10 et 20×20 bins. Ces résultats sont donc suffisamment dignes de confiance.

Nous analysons ensuite le taux de succès basé sur un modèle en histogramme se servant de non plus 1000 mais seulement 100 observations. Le modèle étant normalement moins bon, le taux de succès devrait avoir une courbe évoluant moins vite et moins haut. Il est intéressant de savoir si cette intuition est juste et à quel point le taux de succès serait moins bon.



(a) Evolution de la PI en fonction du nombre d'observations

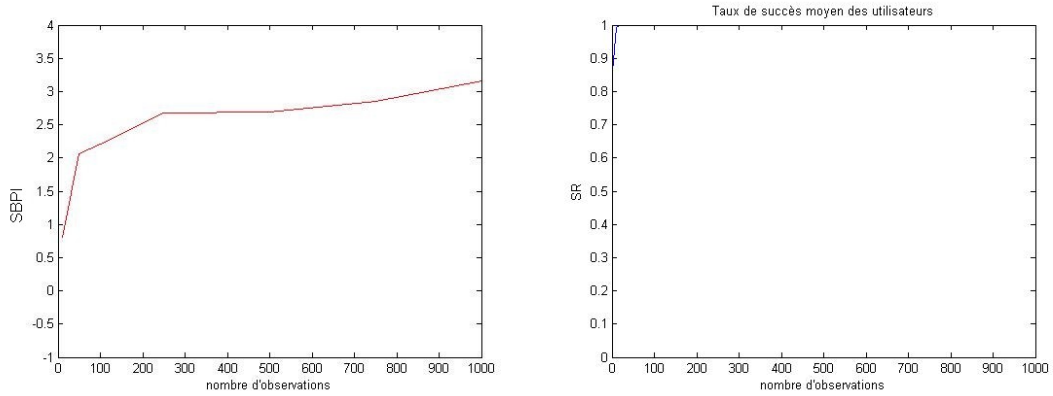
(b) Evolution du SR en fonction du nombre d'observations

FIGURE B.2 – Analyse des résultats en fonction du nombre de bins utilisés par le modèle en histogramme

On remarque tout d'abord que notre conclusion selon laquelle la convergence de l'information perçue est plus lente avec un nombre de bins plus élevé est vérifiée. Cela se remarque avec la courbe de PI pour 20×20 bins qui est en dessous de celles des autres pour peu d'observations disponibles. Concernant le taux de succès, celui-ci commence à une valeur très faible si nous ne prenons qu'une observation. En effet il est à une valeur de 0.025, 0.037, 0.05 pour 5×5, 10×10 et 20×20 bins respectivement. Ce taux est plus ou moins du même ordre de grandeur que $1/n = 1/27 = 0.037$ qui serait obtenu en moyenne si nous avions choisi aléatoirement 1 utilisateur parmi les 27 pour chaque observation. Afin d'obtenir un résultat plus ou moins correct il faut dans ce cas utiliser le plus d'observations possible. Cependant, comme attendu, même avec les 100 observations groupées, le taux de succès reste beaucoup plus bas que celui obtenu si on groupait 100 observations avec un modèle basé sur 1000 observations. On obtient une différence de 0.18 pour 5×5 bins. Le taux de probabilités nulles est à nouveau ici acceptable et est de 0.016, 0.03 et 0.05 pour 5×5, 10×10 et 20×20 bins respectivement.

Méthode des k plus proches voisins

Nous allons effectuer les mêmes analyses que dans la section précédente cette fois à l'aide de la méthode des K plus proches voisins. Commençons par observer l'évolution de l'information perçue et du taux de succès obtenus sur le modèle basé sur les 1000 observations détenues.



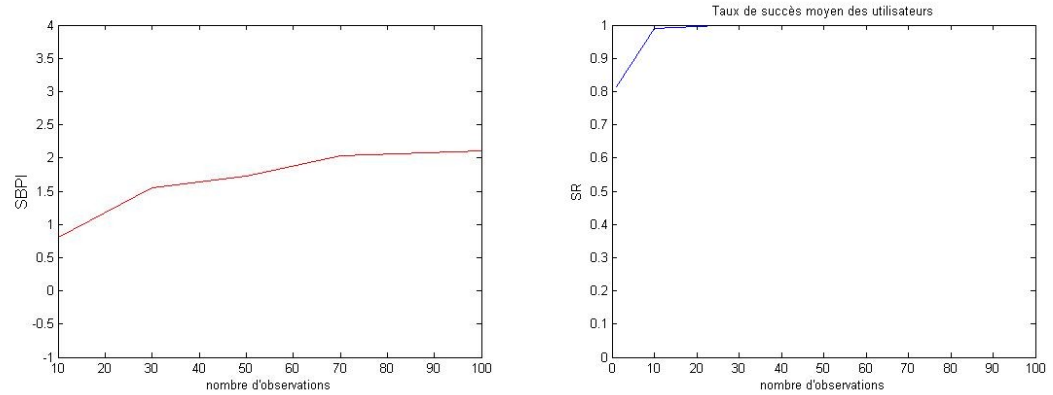
(a) Evolution de la SBPI en fonction du nombre d'observations

(b) Evolution du SR en fonction du nombre d'observations

FIGURE B.3 – Analyse des résultats sur les données réelles pour la méthode du KNN

On remarque qu'à l'aide de la méthode des K plus proches voisins, les résultats sont très bons autant pour l'information perçue basée sur les score que pour le taux de succès. En effet la SBPI monte jusque plus ou moins 3 et surtout, le taux de succès atteint déjà une valeur de 1 pour seulement 25 observations. Pour une seule observation, le taux de succès atteint déjà 0.86. La méthode du KNN semble dès lors efficace. Il est à noter que son temps d'exécution est cependant beaucoup plus élevé et pourrait poser problème pour des données de taille beaucoup plus grande comme le sont généralement les bases de données. De plus, le taux de probabilités nulles est de 0.095 ce qui n'est non négligeable.

Comme pour la méthode utilisant le modèle discret, nous avons également voulu analyser le taux de succès afin de déterminer si celui-ci était calculé sur base d'un modèle de classification d'uniquement 100 observations et non plus 1000.



(a) Evolution de la SBPI en fonction du nombre d'observations

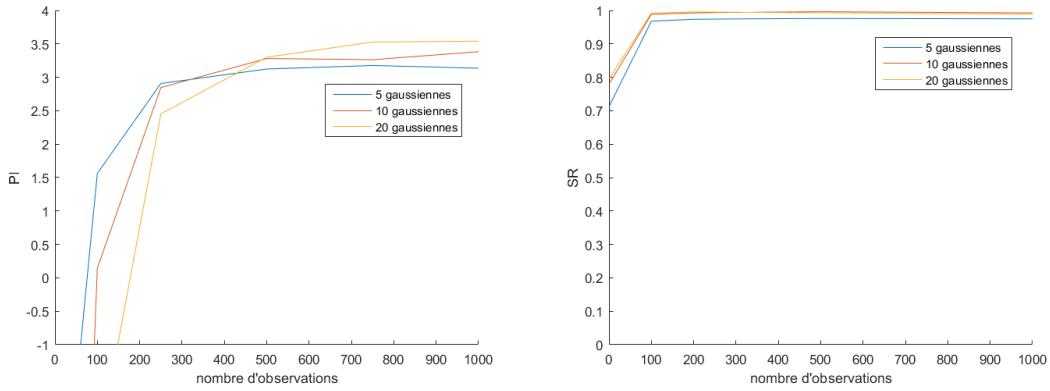
(b) Evolution du SR en fonction du nombre d'observations

FIGURE B.4 – Analyse des résultats sur les données réelles pour la méthode du KNN

On remarque que la méthode des K plus proches voisins rapporte un taux de succès très vite élevé même si celle-ci n'utilise que 100 observations. Bien que celui-ci ne soit qu'à 0.82 et non plus 0.86 si nous prenons chaque observation une à une, cette valeur reste tout de même impressionnante. Le taux de succès atteint de plus la valeur de 1 sans trop de difficulté puisque 30 observations groupées suffisent à obtenir ce résultat. Par contre, le taux de probabilité nulle monte dans ce cas à 0.15.

Modèle par maximisation d'espérance

Nous effectuerons à nouveau la même analyse en s'aidant maintenant la méthode d'attaque utilisant la maximisation d'espérance. Comme nous n'avons aucune idée du nombre de gaussiennes q a priori liées aux utilisateurs, et que nous ne savons d'ailleurs pas si le modèle réel des utilisateurs est proche d'une mixture de gaussiennes, nous essayerons avec des valeurs pour q de 5, 10 et 20.



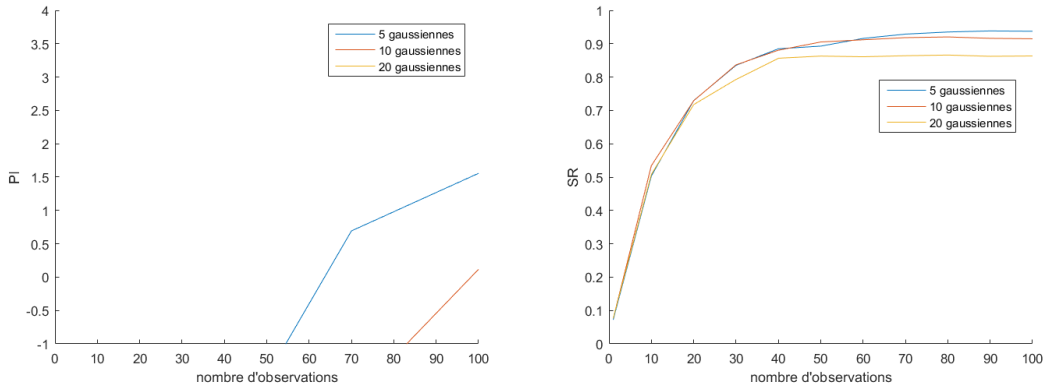
(a) Evolution de la PI en fonction du nombre d'observations

(b) Evolution du SR en fonction du nombre d'observations

FIGURE B.5 – Analyse des résultats sur les données réelles pour la méthode de maximisation d'espérance

Ici comme pour les autres modèles, l'information perçue atteint une valeur comprise entre 3 et 3.5 suivant celle de q. Au plus q est grand, au plus l'information perçue atteint une valeur élevée mais avec une vitesse de convergence plus lente. On observe sur les courbes de taux de succès qu'avec un modèle basé sur les 1000 observations, nous atteignons une valeur de 1 ,ou légèrement en dessous pour $q = 5$, pour seulement 100 observations. Le taux de probabilité nulles est dans ce cas égale à 0 ce qui nous permet de nous fier totalement au graphique du taux de succès obtenu.

Nous allons dès-à-présent analyser le taux de succès pour un modèle créé sur non plus 1000 mais 100 observations. Les résultats sont obtenus sur la figure B.6.



(a) Evolution de la PI en fonction du nombre d'observations

(b) Evolution du SR en fonction du nombre d'observations

FIGURE B.6 – Analyse des résultats sur les données réelles pour la méthode de maximisation d'espérance

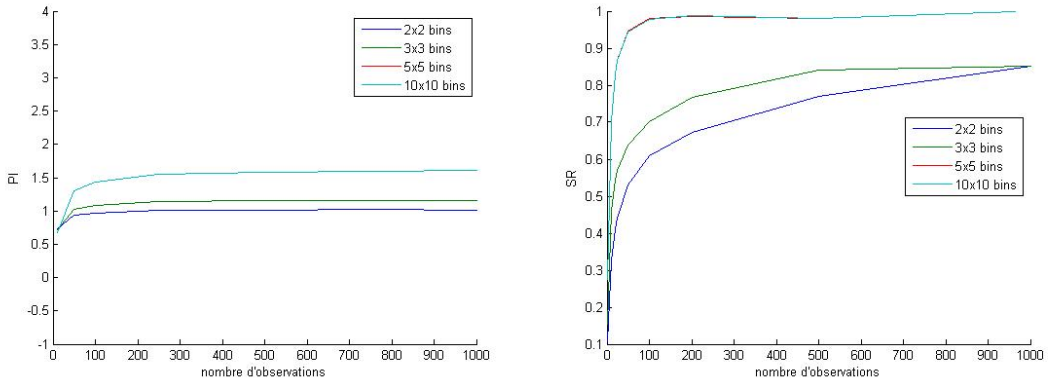
Etant donné que la convergence de l'information perçue est, dans le cas de la maximisation

d'espérance, relativement lente, l'information perçue ne dépasse pas -1 pour $q = 20$, arrive à peine au dessus de 0 pour $q = 10$ et monte tout de même à 1.5 pour $q = 5$. La courbe de taux de succès liée à un nombre de gaussiennes égal à 5 apporte donc dans ce cas de meilleurs résultats et atteint une valeur de 0.94 contre "seulement" 0.86 pour $q = 20$. Les résultats obtenus ici sont remarquables puisque même avec un modèle basé sur peu d'observations, on obtient un taux de succès au dessus de 0.8 pour seulement 50 observations. Pour moins d'observations, il est logique que le taux de succès reste par contre considérablement inférieur et n'est que de 0.076 dans le cas d'une seule observation. Concernant le taux de probabilités nulles, il est à nouveau très bas dans ce modèle et est de 0.00074, 0.001 et 0.003 pour $q = 5$, 10 et 20 respectivement.

B.2 Données agrégées

Modèle discret

Le modèle discret est ici utilisé comme technique d'attaque utilisant une découpe en 5×5 bins. Comme nous l'avons vu dans la section précédente, les résultats liés à ce nombre de bins montrent un taux de succès de 1 et ont surtout un taux très bas de probabilités nulles (0.00074). Regardons maintenant sur la figure B.7 les performances du modèle discret sur les données agrégées.



(a) Evolution de la PI en fonction du nombre d'observations

(b) Evolution du SR en fonction du nombre d'observations

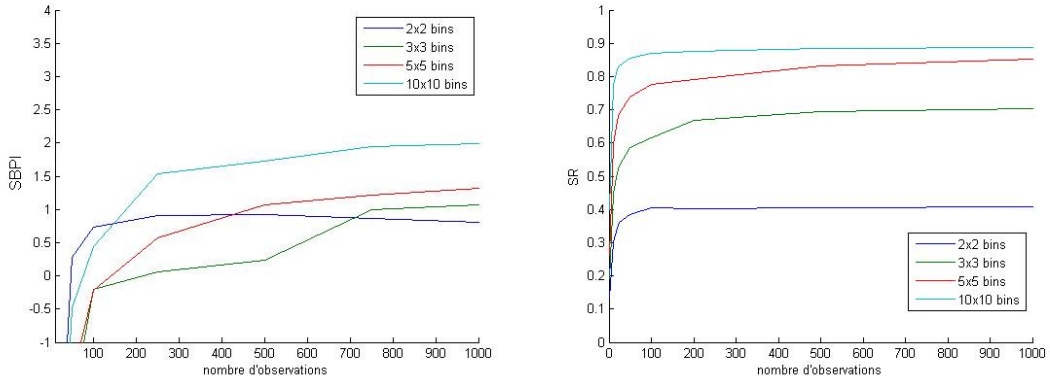
FIGURE B.7 – Analyse des résultats du modèle discret en fonction du nombre de bins utilisés comme moyen de défense

Comme le modèle discret utilise lui même une agrégation de l'espace en 5×5 bins pour estimer les distributions on remarque, en toute logique, qu'une agrégation défensive utilisant un nombre de bins supérieur ou égal à 5×5 donnera toujours les mêmes résultats. La courbe de l'information perçue est alors la même pour 5×5 ou 10×10 bins et est la même que celle obtenue à la section précédente, sans méthode de défense. Le taux de succès atteint donc toujours une valeur de 1 dans ce cas est est très proche de cette valeur pour déjà 200 observations. La technique de défense montre cependant un taux de succès de moins en moins bon pour des découpes plus grandes, c'est à dire utilisant un nombre de bins plus petit. Cela semble logique puisqu'on retire de l'information. On voit bien que pour une découpe en 3×3 bins et en 2×2 bins, la valeur de l'information perçue converge vers une valeur de moins en moins élevée et le taux de succès n'est plus aussi bon qu'avant. Celui-ci ne monte finalement "qu'à" une valeur de plus ou moins 80% pour une découpe en 4 ou 9 bins. Ce taux reste tout de même élevé quand on voit la précision des données qu'il nous reste. Dès lors, on remarque que même si nous acceptons de réduire fortement la précision de nos données, l'information perçue reste positive et le taux de

succès toujours élevé. Les taux de probabilités nulles sont respectivement 0.0001, 0.0005, 0.00074 et 0.00074 pour une agrégation en 2×2, 3×3, 5×5 et 10×10 bins.

Méthode des k plus proches voisins

Nous testerons ensuite cette technique de défense lorsque l'attaquant utilise la méthode des K plus proches voisins. Ainsi nous pourrons voir si l'agrégation marche mieux ou pas pour cette méthode ci.



(a) Evolution de la SBPI en fonction du nombre d'observations

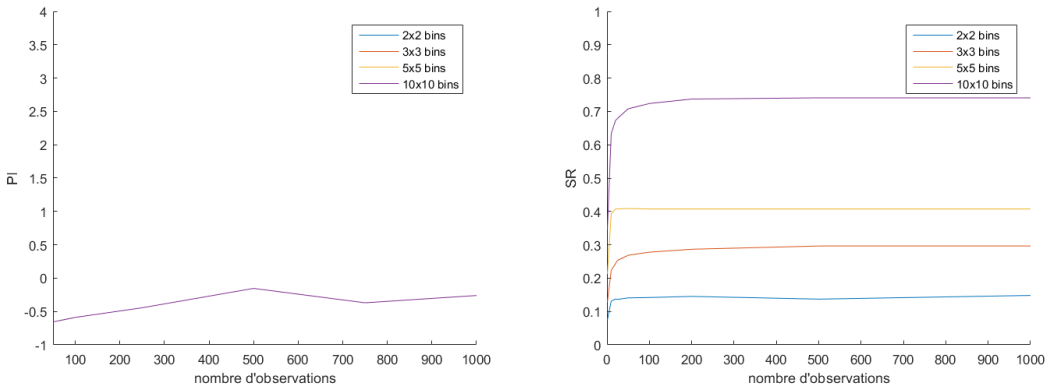
(b) Evolution du SR en fonction du nombre d'observations

FIGURE B.8 – Analyse des résultats du KNN en fonction du nombre de bins utilisés comme moyen de défense

Concernant les courbes de SBPI, celles-ci convergent plus lentement que pour le modèle discret mais atteignent des valeurs finales plus ou moins identiques c'est à dire d'à peu près 0.5 pour une découpe en 2×2 bins et à peu près 1.7 pour une découpe en 10×10 bins. Elles atteignent donc bien une valeur de moins en moins élevée si l'espace est de plus en plus agrégé. Nous regardons alors le taux de succès et on observe que celui-ci varie en fonction de l'agrégation de l'espace : au plus celle-ci est élevée, au moins le taux de succès ne l'est. Cette technique de défense marche donc mieux dans le cas du KNN puisque le taux de succès n'atteint jamais 1, même avec une agrégation de 10×10 bins. Il converge vers une valeur de "seulement" 0.88. Si nous réduisons le nombre de bins jusqu'à 2×2, nous pouvons même arriver à un taux de succès s'arrêtant à 0.4. Les résultats montrent dès lors un intérêt non négligeable de la technique de défense d'agrégation contre une attaque utilisant le KNN. Cela peut s'expliquer par le fait que la méthode des K plus proches voisins se base sur un nombre K d'observations les plus proches d'un point. Dans le cas d'une agrégation de l'espace, il y aura autant d'observations à la même distance de ce point que d'observations se trouvant dans le même bins que lui, et qui plus est aux même coordonnées. Dès lors la méthode devra tous les prendre en compte. Pour ce qui est des différents taux de probabilités nulles, ils sont ici plus alarmants que pour le modèle discret puisqu'il atteint une valeur de 0.092 pour une agrégation de 5×5 bins. Ils sont de 0.039, 0.028 et 0.0057 pour 10×10, 3×3 et 2×2 bins respectivement.

Modèle par maximisation d'espérance

Regardons maintenant les performances du modèle par maximisation d'espérance face à une méthode de défense d'agrégation. Pour ce faire, nous utiliserons une attaque avec un nombre de gaussiennes a priori par utilisateurs fixé à 10. Ce nombre donne de meilleurs résultats pour un modèle basé sur 1000 observations et a un temps de calcul significativement inférieur à celui pour $q = 20$.



(a) Evolution de la PI en fonction du nombre d'observations

(b) Evolution du SR en fonction du nombre d'observations

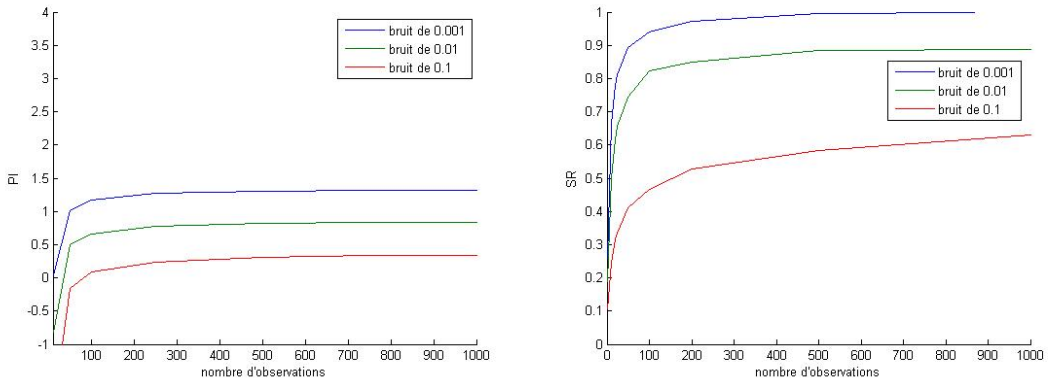
FIGURE B.9 – Analyse des résultats sur les données réelles pour la méthode de maximisation d'espérance

On observe que l'information perçue est exécrable puisque celle-ci n'est même pas positive pour une agrégation en 10×10 bins. Les valeurs pour les autres nombre de bins sont toutes en dessous de -1. On peut donc s'attendre à des résultats de taux de succès peu élevés. Quand on les analyse, on remarque que la méthode de maximisation d'espérance apporte tout de même un taux de succès de 0.7 pour une agrégation de seulement 100 bins. Et ce déjà pour 100 observations. Pour les autres valeurs du nombre de bins, la convergence du taux de succès est presque instantanée mais est de moins en moins élevée lorsque le nombre de bins utilisés diminue. On obtient donc une valeur de seulement 0.4 pour 25 bins, de 0.29 pour 9 bins et de 0.14 pour 4 bins. La méthode de défense joue donc son rôle dans le cas de la maximisation d'espérance et permet de diminuer les valeurs d'information perçue et de taux de succès. Les taux de probabilités nulles sont ici de 0 dans tous les cas.

B.3 Données bruitées

Modèle discret

Le modèle discret est de nouveau ici utilisé avec une découpe en 5×5 bins. La raison est la même que pour l'analyse de l'agrégation. Analysons sur la figure B.10 les performances de ce modèle face au bruitage des données.



(a) Evolution de la PI en fonction du nombre d'observations

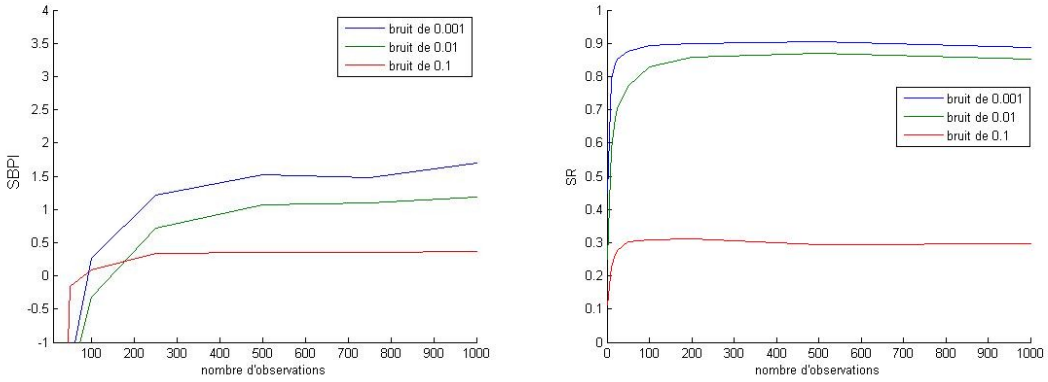
(b) Evolution du SR en fonction du nombre d'observations

FIGURE B.10 – Analyse des résultats pour le modèle discret en fonction du bruitage des données

On remarque qu'au plus on ajoute de bruit aux données, au moins l'information perçue et le taux de succès ne sont élevés. Ce résultat confirme donc bien notre hypothèse. On observe qu'avec un bruit de $v = 0.1$, l'information perçue est à peine positive (0.34 pour 1000 observations) et le taux de succès est loin d'atteindre 1 puisqu'il n'obtient une valeur maximale que de 0.62. Avec un bruit de seulement 0.001 par contre, l'information perçue et le taux de succès ne sont pas énormément affectés et restent similaire aux résultats obtenus sans bruit. L'utilisation d'un niveau de bruit de 0.01 donne quant à elle des résultats intermédiaires avec une valeur de taux de succès allant jusque 0.8. Ces résultats ne sont que peu faussés puisque le taux de probabilités nulles est de 0.001 pour $v = 0.001$ et de 0.02 pour $v = 0.01$ ou 0.1.

Méthode des k plus proches voisins

De nouveau, nous allons également analyser les résultats obtenus avec la méthode des K plus proches voisins.



(a) Evolution de la SBPI en fonction du nombre d'observations

(b) Evolution du SR en fonction du nombre d'observations

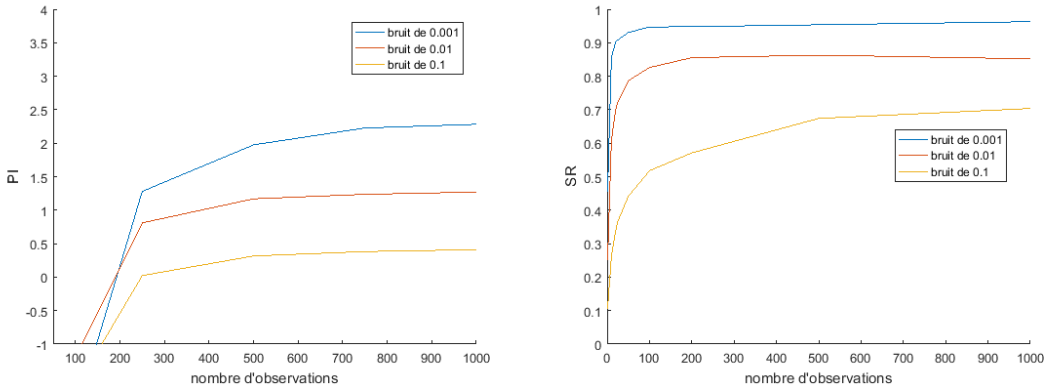
FIGURE B.11 – Analyse des résultats pour le KNN en fonction du bruitage des données

Concernant la pseudo information perçue, elle ne dépasse que très légèrement 0 pour un niveau de bruit de 0.1. Pour les deux autres valeurs de v, la SBPI converge vers une valeur un peu plus haute que dans le cas du modèle discret (1 et 1.5 plutôt que 0.7 et 1.2) mais converge moins vite. De nouveau on observe qu'au plus ce niveau de bruit est faible au plus l'information perçue basée sur des scores sera élevée, ce qui reste logique. Ce résultat est aussi visible sur la courbe du

taux de succès qui converge vers une valeur plus élevée pour un bruitage moins conséquent des données. Dans le cas des K plus proches voisins, on remarque qu'un bruit de $v = 0.1$ va fortement diminuer le taux de succès jusqu'à une valeur de 0.3 seulement ce qui donne alors de l'intérêt à cette technique de défense. Utiliser une telle valeur de bruit est en effet primordial puisque pour des niveaux de seulement $v = 0.01$ ou 0.001, le taux de succès restera plus élevé que 0.8 (pouvant aller jusque 0.9). Mis à part peut être pour le cas $v = 0.001$ où le taux de probabilités nulles est de 0.047, les deux autres courbes sont suffisamment dignes de confiance car elles sont basées sur un taux de seulement 0.009 pour $v = 0.01$ et de 0 pour $v = 0.1$.

Modèle par maximisation d'espérance

Regardons maintenant les performances du modèle par maximisation d'espérance face à une méthode de défense d'ajout d'un bruit gaussien. Pour ce faire, nous utiliserons à nouveau une attaque avec un nombre de gaussiennes a priori par utilisateurs fixé à 10. Les raisons de ce choix sont les mêmes que pour la section sur l'agrégation.



(a) Evolution de la PI en fonction du nombre d'observations

(b) Evolution du SR en fonction du nombre d'observations

FIGURE B.12 – Analyse des résultats sur les données réelles pour la méthode dde maximisation d'espérance

On peut voir sur les graphes que, logiquement, au plus le niveau de bruit est élevé, au moins l'information perçue n'est élevée. De nouveau la valeur de celle-ci n'est que très légèrement positive pour un bruit de $v = 0.1$ et prend des valeurs de 1 et 2.25 pour $v = 0.01$ et 0.001. On remarque que la convergence est de nouveau plus lente pour la maximisation d'espérance que pour les autres méthodes utilisées mais les résultats restent globalement identiques aux autres. Pour le taux de succès, on retrouve à nouveau la corrélation négative logique entre le niveau de bruit et le niveau de taux de succès. Cette technique permet, avec un bruit de $v = 0.1$ de diminuer le taux de succès à 0.65 et lui donne une vitesse de convergence plus lente. Cette défense par bruitage ne permet cependant pas de diminuer considérablement le taux de succès pour des valeurs de $v = 0.01$ et 0.001. En effet, le taux de succès reste respectivement dans ce cas à 0.82 et 0.92. De plus, la convergence est rapide et ces taux sont déjà d'actualité pour seulement 100 observations. De nouveau le taux de probabilités nulles est ici égale à 0 et ne pose donc aucun problème.

Bibliographie

[1] Dictionnaire de français larousse. http://www.larousse.fr/dictionnaires/francais/anonyme/3733. consulté le 04/06 à 15 :22.

[2] Histoire de la géolocalisation. http://www.traceur-gps-pro.fr/quest-ce-que-la-geolocalisation/histoire-de-la-geolocalisation/. consulté le 04/06 à 15 :20.

[3] Multivariate kernel density estimation. https://en.wikipedia.org/wiki/Multivariate_kernel_density_estimation. consulté le 04/06 à 15 :30.

[4] Internet et nouvelles technologies : que reste-t-il de votre vie privée ? http://www.lexpress.fr/actualite/societe/internet-et-nouvelles-technologies-que-reste-t-il-de-votre-vie-privee_911658.html, 2010.

[5] Géolocalisé ? quelle histoire... de gps ! http://www.histoire-cigref.org/blog/geolocalise-quelle-histoire-de-gps/, 2015. consulté le 04/06 à 15 :19.

[6] Usurpation d'identité, prudence ! https://www.nibcdirect.be/fr/blog/usurpation-d-identite-prudence/, 2015. consulté le 04/06 à 15 :27.

[7] Christophe Auffray. Chiffres clés : les ventes de mobiles et de smartphones. http://www.zdnet.fr/actualites/chiffres-cles-les-ventes-de-mobiles-et-de-smartphones-39789928.htm, 2016. consulté le 04/06 à 15 :26.

[8] Marjolaine Baratte. Qu'est-ce que le machine learning ? http://www.jeveuxetredatascientist.fr/quest-ce-que-le-machine-learning/, 2015. consulté le 04/06 à 15 :28.

[9] Louise Barkhuus and Anind K Dey. Location-based services for mobile telephony : a study of users' privacy concerns. In *Interact*, volume 3, pages 702–712. Citeseer, 2003.

[10] B.Bathelot. Définition : Machine learning. http://www.definitions-marketing.com/definition/machine-learning/, 2017. consulté le 04/06 à 15 :29.

[11] Alastair R Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive computing*, 2(1) :46–55, 2003.

[12] Andrew J Blumberg and Peter Eckersley. On locational privacy, and how to avoid losing it forever. *Electronic frontier foundation*, 10(11), 2009.

[13] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. Classification and regression trees. wadsworth & brooks. *Monterey, CA*, 1984.

[14] Guillaume Champeau. Sarthe : l'historique de géolocalisation de google sauve la vie d'un motard. http://www.numerama.com/tech/185157-sarthe-lhistorique-de-geolocalisation-de-google-sauve-la-vie-dun-motard.html, 2016. consulté le 04/06 à 15 :16.

[15] Martin Colbert. A diary study of rendezvousing : implications for position-aware computing and communications for the general public. In *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, pages 15–23. ACM, 2001.

[16] Alexandre Colleau. Namur : deux voleurs d'iphone arrêtés grâce à la géolocalisation. http://belgium-iphone.lesoir.be/2011/06/08/namur-deux-voleurs-diphone-arretes-grace-a-la-geolocalisation/, 2011. consulté le 04/06 à 15 :19.

[17] Frank A Cowell and Emmanuel Flachaire. Statistical methods for distributional analysis. *Handbook of Income Distribution*, 2 :359–465, 2015.

[18] Thomas Coëffé. Chiffres réseaux sociaux - 2017. http://www.blogdumoderateur.com/chiffres-reseaux-sociaux/, 2016. consulté le 04/06 à 15 :25.

[19] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd : The privacy bounds of human mobility. *Scientific reports*, 3 :1376, 2013.

[20] Matt Duckham and Lars Kulik. Location privacy and location-aware computing. *Dynamic & mobile GIS : investigating change in space and time*, 3 :35–51, 2006.

[21] Tarn Duong and Martin Hazelton. Plug-in bandwidth matrices for bivariate kernel density estimation. *Journal of Nonparametric Statistics*, 15(1) :17–30, 2003.

[22] Fabrice Epelboin et Sarah Perez. Les dangers de la géolocalisation sur le web. http://tempsreel.nouvelobs.com/rue89/rue89-tech/20100727.RUE7746/les-dangers-de-la-geolocalisation-sur-le-web.html, 2010. consulté le 04/06 à 15 :18.

[23] Lars Fischer, Stefan Katzenbeisser, and Claudia Eckert. Measuring unlinkability revisited. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, pages 105–110. ACM, 2008.

[24] Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity : Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1) :1–18, 2008.

[25] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.

[26] Wolfgang Härdle, Marlene Müller, Stefan Sperlich, and Axel Werwatz. *Nonparametric and semiparametric models*. Springer Science & Business Media, 2012.

[27] Ibrar Hussain. Clustering in machine learning.

[28] Loïc Knuchel. Le machine learning, c'est quoi exactement ? http://loic.knuchel.org/blog/2013/11/22/le-machine-learning-cest-quoi-exactement/, 2013. consulté le 04/06 à 15 :24.

[29] John Krumm. Inference attacks on location tracks. *Pervasive computing*, pages 127–143, 2007.

[30] John Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6) :391–399, 2009.

[31] Alex Lane. Machine learning and privacy : A problem ? can we be sure of privacy when machine learning is so powerful ? https://channels.theinnovationenterprise.com/articles/machine-learning-and-privacy-a-problem, 2016. consulté le 04/06 à 15 :21.

[32] Jure Leskovec and Andrej Krevl. SNAP Datasets : Stanford large network dataset collection. http://snap.stanford.edu/data, june 2014. consulté le 04/06 à 15 :15.

[33] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[34] John Markoff. Microsoft finds cancer clues in search queries. https://www.nytimes.com/2016/06/08/technology/online-searches-can-identify-cancer-victims-study-finds.html?_r=0, 2016. consulté le 04/06 à 15 :23.

[35] MATLAB. Matlab : predict. https://nl.mathworks.com/help/stats/classificationknn.predict.html, 2017. consulté le 08/06 à 11 :48.

[36] Benjamin NGUYEN. Techniques d'anonymisation. *Statistique et Société*, 2(4) :43–50, 2015.

[37] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information : k-anonymity and its enforcement through generalization and suppression. Technical report, Technical report, SRI International, 1998.

[38] Rob Schapire. Machine learning algorithms for classification. *Princeton University*, 10, 2015.

[39] David W Scott. *Multivariate density estimation : theory, practice, and visualization*. John Wiley & Sons, 2015.

[40] Reza Shokri, Julien Freudiger, Murtuza Jadliwala, and Jean-Pierre Hubaux. A distortion-based metric for location privacy. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 21–30. ACM, 2009.

[41] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *Security and privacy (sp), 2011 ieee symposium on*, pages 247–262. IEEE, 2011.

[42] François-Xavier Standaert. From minimal distortion to good characterization : Perceptual utility in privacy-preserving data publishing. *De Gruyter Open*, 2016.

[43] François-Xavier Standaert. The perceived information : Definition, intuition and applications, December 2017.

[44] Latanya Sweeney. k-anonymity : A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05) :557–570, 2002.

[45] Dan Tynan. Why location privacy is important. http://www.itworld.com/article/2752981/mobile/why-location-privacy-is-important.html, 2010. consulté le 04/06 à 15 :17.

[46] Matt P Wand and M Chris Jones. Comparison of smoothing parameterizations in bivariate kernel density estimation. *Journal of the American Statistical Association*, 88(422) :520–528, 1993.

[47] A. Westin and D.J. Solove. *Privacy and Freedom*. Ig Publishing, 2015.

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve **www.uclouvain.be/epl**